

UNIVERSIDAD AUTONOMA DE NUEVO LEON  
FACULTAD DE CIENCIAS FISICO-MATEMATICAS



LENGUAJE DE PROGRAMACION  
NATURAL Y MANEJO DE BASE  
DE DATOS ADABAS

TESIS

QUE PARA OBTENER EL TITULO DE  
LICENCIADO EN CIENCIAS COMPUTACIONALES  
PRESENTAN

NELLY PATRICIA DUARTE CRUZ  
ADRIANA GUADALUPE ARIZPE IRACHETA

SAN NICOLAS DE LOS GARZA, N. L., MARZO DE 1990



TL

QA76

.73

.N3

D83

1990

c.1





1080171515



**UNIVERSIDAD AUTONOMA DE NUEVO LEON**  
**FACULTAD DE CIENCIAS FISICO-MATEMATICAS**

**TEMA**

**LENGUAJE DE PROGRAMACION NATURAL Y**  
**MANEJO DE BASE DE DATOS ADABAS**

**PARA OBTENER EL TITULO DE**  
**LICENCIADO EN CIENCIAS COMPUTACIONALES**

**PRESENTAN**

**NELLY PATRICIA DUARTE CRUZ**  
**ADRIANA GUADALUPE ARIZPE IRACHETA**

**SAN NICOLAS DE LOS GARZA, N.L.**



**I N D I C E**



# INDICE

	Pag.
OBJETIVO.....	01
INTRODUCCION.....	02
NATURAL.....	04
MENSAJES DEL PROMPT.....	13
SISTEMA DE CONTROL DE COMANDOS Y SINTAXIS .....	14
SUBCOMANDOS DE EDICION .....	16
FACILIDAD DEL HELP.....	18
UTILIDAD DEL MAP .....	19
COMANDOS PARA EL CONTROL DE LA TERMINAL.....	21
SINTAXIS DE LOS ESTATUTOS DE NATURAL.....	24
SINTAXIS DEL CRITERIO DE CONDICION LOGICA.....	25
ESTATUTOS DE NATURAL.....	27
PARAMETROS GLOBALES (SET GLOBALS).....	43
FORMATO Y LONGITUD PARA VARIABLES (USUARIO).....	49
REGLAS DE TRANSFERENCIA DE DATOS.....	50
MANEJO DE OVERFLOW ARITMETICO.....	50
VARIABLES DEL SISTEMA.....	51
SISTEMA DE FUNCIONES ESTANDARD.....	53
PALABRAS RESERVADAS DE NATURAL.....	54
MENSAJES DE ERROR.....	57
EJEMPLOS DE PROGRAMAS.....	58



# INDICE

	Pag.
<b>PREDICT</b> .....	66
DEFINICION DE DATOS DEL ARCHIVO.....	69
EJEMPLOS DE ESTRUCTURAS DE ARCHIVO.....	70
TIPOS DE ARCHIVOS.....	74
ATRIBUTOS DE LOS ARCHIVOS.....	75
SISTEMA DE MANTENIMIENTO.....	76
<b>ADABAS</b> .....	81
ARCHIVOS.....	85
REGISTROS.....	86
COMPRESION DE CAMPOS.....	89
ASOCIADOR.....	92
TIPOS DE ACCESO.....	102
ACCESO SECUENCIAL.....	102
ACCESO ALEATORIO.....	107
CONSIDERACIONES SOBRE ACCESOS.....	114
REINICIO Y RECUPERACION DE INFORMACION.....	115
REINICIALIZACION DE DATOS.....	116
<b>ANEXO</b> .....	118
<b>GLOSARIO</b> .....	125
<b>AGRADECIMIENTOS</b> .....	130
<b>BIBLIOGRAFIA</b> .....	131

**O B J E T I V O**



## **O B J E T I V O**

EL PROPOSITO PRIMORDIAL DE ESTA TESIS ES DAR A CONOCER A CUALQUIER PERSONA RELACIONADA CON EL AREA DE SISTEMAS, EL MANEJO DEL LENGUAJE DE PROGRAMACION **NATURAL** Y SU RELACION, ASI COMO EL USO, DE LA BASE DE DATOS **ADABAS**.

# I N T R O D U C C I O N

## INTRODUCCION

EN LOS ULTIMOS AÑOS LAS COMPUTADORAS HAN EVOLUCIONADO ACELERADAMENTE, CADA DIA SE LOGRAN MAS DESCUBRIMIENTOS QUE PERMITEN CREAR NUEVOS HARDWARE Y SOFTWARE A LA ALTURA DE LAS NECESIDADES ACTUALES DE LA HUMANIDAD.

COMO CONSECUENCIA DE ESTO DECIDIMOS DESARROLLAR NUESTRO TEMA SOBRE **NATURAL** QUE ES UN LENGUAJE RELATIVAMENTE "NUEVO" Y ESCASAMENTE CONOCIDO EN NUESTRO PAIS, PERO QUE, POCO A POCO SE VA DANDO A CONOCER Y ALGUNAS EMPRESAS EMPIEZAN A INTERESARSE POR ADQUIRIRLO, YA QUE, **NATURAL** ES UN LENGUAJE DE FACIL PROGRAMACION QUE PUEDE REALIZAR PROCESOS COMPLEJOS, ADEMAS DE SER MUY VERSATIL POR QUE PERMITE DESARROLLAR APLICACIONES EN UNA GRAN DIVERSIDAD DE AREAS, COMO LO SON EL AREA ADMINISTRATIVA, EN LA INDUSTRIA, EN LA EDUCACION, EN LA CIENCIA, EN LA INVESTIGACION Y MUCHAS MAS.

QUIZAS EN UN FUTURO CERCANO SE PROLIFERE EL USO DE ESTE LENGUAJE O UNO MUY SIMILAR, POR TAL MOTIVO, APORTAMOS ESTE MATERIAL DE UNA MANERA COMPLETA Y SENCILLA PARA DAR AL LECTOR UNA GUIA CLARA Y PRECISA DE LO QUE ES **NATURAL**, LA CUAL LE PUEDE SER MUY UTIL AL MOMENTO DE APRENDER A MANEJARLO.

UN SEGUNDO OBJETIVO A INCLUIR DENTRO DE ESTA TESIS, ES LA BASE DE DATOS **ADABAS**, YA QUE **NATURAL**, SE ENCUENTRA TOTALMENTE INTEGRADO A ELLA.

**ADABAS** FUE CREADO ANTES QUE **NATURAL** Y POR LO TANTO ESTE LENGUAJE FUE DISEÑADO PARA TRABAJAR CONJUNTAMENTE CON **ADABAS**, LA CUAL ES UNA BASE DE DATOS RELACIONAL PERO CON MUCHAS OTRAS VENTAJAS QUE CUALQUIER OTRA BASE DE DATOS RELACIONAL COMUN, ADEMAS DE QUE MANTIENE LA INTEGRIDAD DE LOS DATOS DE UNA MANERA CONFIABLE Y SEGURA, DE ESTO SE DARA CUENTA EL LECTOR AL CONOCER **ADABAS**.

DENTRO DE ESTE TEMA , ENTRE OTRAS COSAS , HABLAREMOS DE COMO LA ESTRUCTURA DE **ADABAS** DENTRO DE LA COMPUTADORA, LOS TIPOS DE ARCHIVOS QUE MANEJA, SUS TIPOS DE ACCESO, LA SEGURIDAD CON LA QUE ESTA CUENTA Y ALGUNAS DE LAS CARACTERISTICAS MAS IMPORTANTES DENTRO DE ELLA, CON EL FIN DE DAR UNA VISION MAS A FONDO DE **ADABAS**.

PARA OBTENER UNA ORGANIZADA DOCUMENTACION Y DEFINICION DE ARCHIVOS DECIDIMOS INCLUIR DENTRO DE NUESTRA TESIS UNA INTRODUCCION A LO QUE ES EL DICCIONARIO DE DATOS **PREDICT**, EL CUAL, NOS ES DE GRAN AYUDA PARA LA ORGANIZACION DE LOS PROGRAMAS DESARROLLADOS DENTRO DE **NATURAL**, NOS PROPORCIONA LA RELACION QUE TIENE CADA PROGRAMA CON CADA ARCHIVO, DEFINICION DE ARCHIVOS Y CAMPOS ASI COMO MUCHAS OTRAS OPCIONES UTILES PARA LA CREACION DE ESTRUCTURAS.



## INTRODUCCION

**NATURAL, ADABAS Y PREDICT SON UNA HERRAMIENTA MUY PODEROSA DENTRO DE EL AREA DE SISTEMAS, POR LO QUE AL LECTOR CON CONOCIMIENTOS EN COMPUTACION, BASICAMENTE EN PROGRAMACION, LE SERA MUY UTIL APRENDER, NO COMPLETAMENTE, PERO SI ADQUIRIRA LAS HERRAMIENTAS NECESARIAS PARA EL FUNCIONAMIENTO DE CADA UNO DE LOS TEMAS CON LA AYUDA DE ESTE TRABAJO. CABE MENCIONAR QUE EN CADA TEMA SE INCLUYEN EJEMPLOS PARA ILUSTRAR AL LECTOR, ADEMAS DE CONTENER UN GLOSARIO DE TERMINOS UTILIZADOS EN EL DESARROLLO DE ESTA TESIS.**

N A T U R A L

## **NATURAL**

**NATURAL** ES UN LENGUAJE DE TECNOLOGIA DE CUARTA GENERACION.

ES UN SISTEMA INTEGRADO DE DESARROLLO DE APLICACIONES EFICIENTE Y RAPIDO. CUENTA CON GRAN AVANCE EN EL SISTEMA DE SOFTWARE Y HARDWARE CON RESPECTO AL PROCESO DE DATOS DANDO FACILIDAD PARA PROCESAR GRANDES VOLUMENES DE INFORMACION.

**NATURAL** PROVEE :

- UN MECANISMO SENCILLO PARA DESARROLLAR APLICACIONES COMPLEJAS CON UN CONJUNTO COMUN DE HERRAMIENTAS.
- EFECTIVIDAD, UNIENDO LA EXPERIENCIA DE LAS APLICACIONES CON LA DE LOS NEGOCIOS ATRAVES DE UN PROTOTIPO AVANZADO.
- ORGANIZA Y SOPORTA UNA AMPLIA VARIEDAD DE ESTRUCTURAS DE ARCHIVOS, SISTEMAS OPERATIVOS, MONITORES DE TELEPROCESO Y CONFIGURACIONES DE HARDWARE.
- UNA BASE INTEGRADA PARA LAS NECESIDADES INDIVIDUALES DE LOS USUARIOS.
- CUBRE AMPLIAMENTE LOS REQUERIMIENTOS DE SEGURIDAD E INTEGRIDAD DE LA INFORMACION.
- OFRECE LOS ULTIMOS IMPLEMENTOS Y AVANCES EN SOFTWARE Y HARDWARE SIN PROBLEMAS DE COMPATIBILIDAD.

**NATURAL** FACILITA EL MANTENIMIENTO PORQUE GENERA MENOS CODIGOS PARA CADA APLICACION QUE OTROS LENGUAJES. AL GENERAR MENOS LINEAS DE CODIGOS SIGNIFICA UN DESARROLLO MAYOR Y MAS RAPIDO A LA VEZ DE QUE E MENOR EL MANTENIMIENTO.

## NATURAL

**NATURAL** OPERA INDEPENDIEMENTE DEL SISTEMA OPERATIVO Y DEL MONITOR DE TELEPROCESO POR LO TANTO NO HAY CAMBIOS EN LA GENERACION DE CODIGOS.

LAS APLICACIONES DE **NATURAL** GENERAN UN CENTRO DE DATOS INCORPORADO NECESARIO PARA NO SER CAMBIADO SI SE VA A CORRER EN LOCALIDADES REMOTAS O MONITORES Y SISTEMAS OPERATIVOS DIFERENTES.

**NATURAL** SOPORTA UN AMPLIO RANGO DE DESARROLLO DE APLICACIONES PARA DIFERENTES ESTRUCTURAS DE ARCHIVOS VSAM Y OTRAS ESTRUCTURAS DE ARCHIVOS SECUENCIALES, BASES DE DATOS COMO LO SON : ADABAS , DL1 IMS, Y CUALQUIER APLICACION QUE USE MULTIPLES ESTRUCTURAS DE DATOS.

TODAS LAS OPERACIONES DE ESTRUCTURAS DE ARCHIVOS QUE SON MANEJADAS POR **NATURAL** SON COMPLETAMENTE TRANSPARENTES PARA EL USUARIO, LO CUAL ES UNA GRAN VENTAJA POR QUE ASI LOS PROGRAMADORES SE DEDICAN Y CENTRAN EXCLUSIVAMENTE A LA APLICACION QUE ESTEN LLEVANDO ACABO Y NO EN PROBLEMAS DE ACCESO DE DATOS Y COMUNICACION DE TERMINALES.

**NATURAL** TIENE UN EXTENSO SOPORTE DE HARDWARE, ESTA DISPONIBLE PARA EQUIPOS IBM Y COMPATIBLES DE IBM, DEC, FUJITSU Y SISTEMAS SIEMENS. LA CAPACIDAD DE SISTEMAS DE HARDWARE ES MUY GRANDE Y TRASPARENTE PARA **NATURAL**.

**NATURAL** SOPORTA UNA GRAN GAMA DE SISTEMAS OPERATIVOS INCLUYENDO

- \* IBM'S MVS
- \* MVS/XA
- \* VM
- \* DOS/VSE
- \* VSE/SP2
- \* OS/VS1
- \* DEC (VMS)
- \* SIEMEN'S BS2000 Y BS3000.

LA VARIEDAD DE MONITORES SOPORTADOS PARA LA IBM INCLUYEN

- \* COM-LETE DE Software AG
- \* CICS
- \* IMS/DC
- \* TSO
- \* CMS
- \* Y OTROS

Y LOS MONITORES PARA SIEMENS RTIO Y UTM.



## NATURAL

**NATURAL** OFRECE TODAS LAS FUNCIONES Y FACILIDADES NECESARIAS PARA EL DESARROLLO DE APLICACIONES, GENERACION DE REPORTES Y PROCEDIMIENTOS INTERACTIVOS QUE SON DADOS EN UN AMBIENTE INTEGRADO SENCILLO, DE ESTA MANERA TIENE UNA FUNCIONALIDAD Y FLEXIBILIDAD MAXIMA EN MANTENIMIENTO.

UN COMPLEMENTO EN LA ESTACION DE TRABAJO SON LAS FACILIDADES INTEGRADAS TALES COMO:

- MANEJADORES DE MENUS DE ACCESO PARA TODAS LAS FUNCIONES.
- COMANDOS DIRECTOS COMPETENTES PARA LA EXPERIENCIA DEL USUARIO
- EDITORES INTELIGENTES PARA DEFINICION DE AREA DE DATOS, DISEÑO DE PANTALLAS (MAPAS), DESARROLLO DE CODIGOS FUENTE, ESPECIFICACIONES Y REPORTES.
- INTEGRACION DE COMPUTADORAS PERSONALES.
- MANEJO DE IMPRESION.
- DEFINICION DE TAREAS

CON **NATURAL** LOS REPORTES SE PUEDEN HACER FÁCILMENTE PARA LAS NECESIDADES DEL USUARIO ATRAVES DEL USO DE FORMAS LIBRES DE SINTAXIS, POR MEDIO DEL EDITOR DE REPORTES, EL CUAL CREA SALIDAS PRE-DEFINIDAS DE REPORTES, O FUNCIONES EN PANTALLA QUE FACILITAN LOS FORMATOS DE LOS REPORTES QUE REQUIEREN UNA BASTA CANTIDAD DE DETALLES Y POSICIONAMIENTOS ESPECIALES.

LOS ESTATUTOS DE **NATURAL** ( CODIGO FUENTE ) COMO SON ALIMENTADOS SON ACTIVAMENTE COMPILADOS Y CORREGIDOS LINEA POR LINEA. UNA VEZ QUE EL PROGRAMA ESTA SINTACTICAMENTE CORRECTO EL RESULTADO ES DESPLEGADO. ESTE MEDIO AMBIENTE REDUCE DRASTICAMENTE EL TIEMPO Y COSTO DE DESARROLLO, ADEMAS DEL COSTO Y ANALISIS DE PROGRAMAS PROTOTIPO.

## NATURAL

**NATURAL** PUEDE SALVAR PROGRAMAS COMO FUENTE Y/O OBJETO AUTOMATICAMENTE INVOCANDO LAS FACILIDADES DE LAS LIBRERIAS. CUANDO UN PROGRAMA ES CATALOGADO, TODA LA INFORMACION CONCERNIENTE A LOS RECURSOS USADOS EN EL PROGRAMA SON INCORPORADOS AUTOMATICAMENTE EN EL DICCIONARIO DE DATOS ATRAVES DE LA FACILIDAD DE **PREDICT CROSS REFERENCE** ( MENCIONADO MAS ADELENTE).

**NATURAL** PROVEE PRUEBAS INTERACTIVAS DE PROGRAMAS EN LINEA Y BATCH. EL CONTROL DE JOBS EN BATCH PUEDE SER CAMBIADO DURANTE LA EJECUCION DEL JOB SI LAS CONDICIONES DE LA APLICACION LO PERMITEN.

**PREDICT** PROVEE VERIFICACION DE DEFINICION DE DATOS AUTOMATICA, DOCUMENTACION AUTOMATICA DE PROGRAMAS, PRESENTACION DE VISTAS DE DATOS Y FACILIDADES DE CONTROL.

**NATURAL** TIENE UN FUNCIONAMIENTO SUPERIOR CON FACILIDADES UNICAS QUE CUALQUIER OTRA TECNOLOGIA DE CUARTA GENERACION. ESTAS FACILIDADES INCLUYEN TRANSMISION DE SOLO LA PORCION MODIFICADA DE MAPAS, TECNICAS DE MANEJO DE GRANDES BUFFERS, UN COMPILADOR OPTIMIZADO QUE MAXIMIZA EL USO DEL TELEPROCESO ATRAVES DE ALTOS AVANCES EN MANEJADORES.

**NATURAL** PUEDE ENCADENAR UN MAINFRAME Y COMPUTADORAS PERSONALES SIENDO TODO ESTO TRANSPARENTE PARA EL USUARIO, ESTO LO HACE POR MEDIO DEL **NATURAL CONNECTION**.

**NATURAL** SOPORTA UN DISEÑO ESTRUCTURADO COMPLETO, USANDO TECNICAS DE PROGRAMACION TOP-DOWN. EL LENGUAJE DE PROGRAMACION **NATURAL** NO INCLUYE LA INSTRUCCION DE "GOTO".

EN NATURAL EXISTEN ESTATUTOS QUE PERMITEN:

- SELECCION DIRECTA DE DATOS CON CAPACIDAD DE LOGICA BOOLEANA Y SORTEO.
- SOPORTE DE ESTRUCTURAS DE DATOS NUMERICOS, ENTEROS, DE PUNTO FLOTANTE, BINARIOS, EMPACADOS, FECHA, TIEMPO, LOGICOS Y ALFANUMERICOS.
- MULTINIVEL DE PUNTOS DE CORTE PARA HACER CALCULOS AUTOMATICAMENTE ATRAVES DE SERIES DE DATOS SORTEADOS.
- UN CONJUNTO COMPLETO DEL SISTEMA DE FUNCIONES ESTANDAR PARA NEGOCIOS Y PROCESOS CIENTIFICOS.
- CORTES AUTOMATICOS DE PAGINA, DE INICIO DE DATOS Y FIN DE DATOS.

## NATURAL

- UN CONJUNTO COMPLETO DE PROCESO DE OPERACIONES DE SUBROUTINAS, INCLUYENDO SUBROUTINAS INTERNAS DE NATURAL Y EXTERNAS DE LENGUAJES DE 3era. GENERACION.
- TRANSFERIR COMPLETAMENTE EL CONTROL A OTRO PROGRAMA ATRAVES DEL ESTATUTO "FETCH" Y PERMITE A UN PROGRAMA CONVENCIONAL LLAMAR A UN PROGRAMA EN NATURAL.
- AVANZADOS PROCESOS LOGICOS USANDO LOS ESTATUTOS IF/THEN/ELSE CON EXPRESIONES ARITMETICAS.
- MANEJO DE FACILIDADES DE DATOS INCLUYENDO ADICION, MODIFICACION Y ELIMINACION DE DATOS.
- SOPORTE DE TRANSACCIONES LOGICAS PARA LA COMPLETA INTEGRIDAD DE LOS DATOS ATRAVES DE PROGRAMAS, PANTALLAS, SERIES DE PANTALLAS Y SERIES DE PROGRAMAS RELACIONADOS.
- NUMERACION AUTOMATICA DE PAGINAS, TITULOS, ENCABEZADOS, MASCARAS DE EDICION Y FORMATEO DE COLUMAS.
- IMPLICITAMENTE LOS LOOPS SON GENERADOS POR LOS ESTATUTOS DE ACCESO DIRECTO DE DATOS, PERO TAMBIEN SE PUEDEN GENERAR ATRAVES DEL "REPEAT" Y "FOR".
- CADA LOOP PUEDA TRANSFERIR EL CONTROL A RUTINAS ESPECIFICAS, AL INICIO O FIN DEL PROCESO LOOP.
- UN SIMPLE COMANDO "FIND" SELECCIONA UN GRUPO DE REGISTROS DE UNA VISTA LOGICA. VARIOS "FIND" ENCADENADOS PUEDEN TENER UN NUMERO ILIMITADO DE REGISTROS DE MULTIPLES VISTAS LOGICAS.

LOS EDITORES DE **NATURAL** SIMPLIFICAN LA ESTRUCTURA DEL PROGRAMA Y REDUCEN LAS REDUNDANCIAS, SON FLEXIBLES Y SIMPLIFICAN EL PROCESO DEL DESARROLLO. CONSTA DE CINCO EDITORES:

- A) EDITOR DE AREA DE DATOS
- B) EDITOR DE CODIGO FUENTE
- C) EDITOR DE DISEÑO DE PANTALLAS
- D) EDITOR DE GENERACION DE REPORTE
- E) EDITOR DE ESPECIFICACION DE PROGRAMAS

## NATURAL

A CONTINUACION SE EXPLICAN CADA UNO DE ELLOS.

A) **EDITOR DE AREA DE DATOS** - ES USADO PARA DEFINIR LA ESTRUCTURA GLOBAL DE DATOS PARA CADA APLICACION EN BASE A LAS VISTAS LOGICAS CREADAS POR EL ADMINISTRADOR. LOS PROGRAMADORES PUEDEN SELECCIONAR LOS ELEMENTOS DE UNA O MAS VISTAS LOGICAS Y SI SE NECESITA SOLO SE PUEDEN INCLUIR LOS ELEMENTOS REQUERIDOS PARA LA APLICACION INDIVIDUAL Y SELECCIONAR LOS CAMPOS EN CUALQUIER ORDEN Y HACER MULTIPLES REFERENCIAS.

PARA CADA CAMPO EL PROGRAMADOR PUEDE DEFINIR EL NOMBRE DEL CAMPO, FORMATO, LONGITUD, INDICE DEL VECTOR, VALOR INICIAL, MASCARA DE EDICION Y COMENTARIOS.

EL EDITOR SOPORTA TODOS LOS TIPOS ESTANDAR INCLUYENDO EL LOGICO, FECHA, TIEMPO Y PUNTO FLOTANTE.

B) **EDITOR DE CODIGO FUENTE** - ESTE EDITOR INCLUYE UN GRAN RANGO DE ASPECTOS PARA LA CREACION Y MODIFICACION DE ESTATUTOS PARA LA PROGRAMACION ESTRUCTURADA. ALGUNOS DE LOS ASPECTOS INCLUYEN:

- COMANDOS DE LINEA PARA MODIFICACIONES FUENTE.
- MULTIPLES COMANDOS DE LINEA EN UNA SOLA ENTRADA.
- COMANDOS DE POSESIONAMIENTO QUE SON MANEJADOS Y MODIFICADOS POR LAS LLAVES PF'S.
- SALVA PERIODICA Y AUTOMATICA DE PROGRAMAS.
- COMANDOS QUE PERMITEN EL DESPLIEGE DE PANTALLAS EN CODIGO FUENTE Y/O VISTAS LOGICAS DURANTE LA EDICION DEL PROGRAMA.
- GENERACION DINAMICA DEL **CROSS REFERENCE**
- MENU DE AYUDA



## NATURAL

C) **EDITOR DE DISEÑO DE PANTALLAS** - ES UN EDITOR EFICIENTE Y FLEXIBLE, EL CUAL CONTIENE UN CONJUNTO COMPLETO DE FUNCIONES CREADAS ESPECIALMENTE PARA EL DISEÑO, PRUEBA, ALMACENAMIENTO, E IMPLANTACION DE PANTALLAS (MAPAS) INCORPORANDO VISTAS LOGICAS Y REGLAS DE VALIDACION DE CAMPOS.

### INCLUYE:

- DESPLIEGE MULTIPLE DE VISTAS LOGICAS Y MAPAS ACTUALMENTE DEFINIDOS.
- SOPORTE DE MAPAS HASTA DE 250 POR 250 CARACTERES.
- INCLUSION DE TEXTO DE AYUDA.
- VALIDACION DE REGLAS QUE PUEDEN SER ESPECIFICAS PARA LA APLICACION O PUEDEN SER REGLAS INCORPORADAS DEL DICCIONARIO.
- ESTRUCTURAS VERTICALES U HORIZONTALES.
- COMANDOS FLEXIBLES PARA EDICION DE CAMPOS Y OPERACIONES DE LINEA.
- GENERACION DE MAPAS DE ENTRADA O SALIDA.
- UNA GAMA DE ATRIBUTOS INCLUYENDO COLOR, VIDEO REVERSA, SUBRAYADO Y MAS.
- OBTENCION DE PANTALLAS PRE-DEFINIDAS PARA ASEGURAR QUE SEAN CONSISTENTES CON LA APLICACION.
- MENSAJES DE ERROR, DESPLEGANDOS AL TOPE O FINAL DEL MAPA.
- CHECADOR QUE VERIFICA QUE LA PANTALLA (MAPA) SEA COMPATIBLE CON LOS PATRONES USADOS EN LA APLICACION.
- SOPORTA LENGUAJES COMO INGLES, ALEMAN, ESPAÑOL Y OTROS.
- COMANDOS DE SALVACION AUTOMATICA.

D) **EDITOR DE GENERACION DE REPORTES** - ES UNA HERRAMIENTA PODEROSA Y EFICIENTE PARA DISEÑAR FORMATOS DE REPORTES PARA VARIAS APLICACIONES DE PROGRAMAS. UNA VENTAJA ES QUE EL PROGRAMADOR Y EL USUARIO PUEDEN VER EL REPORTE ANTES DE QUE SEA PARTE DE LA APLICACION.

## NATURAL

**NATURAL** OFRECE MAXIMA FLEXIBILIDAD PARA LA CREACION Y GENERACION CONTINUA DE UNA VARIEDAD DE REPORTE USANDO COMANDOS. INCLUYENDO:

- REPORTE DE PROGRAMAS ESPECIFICOS GENERADOS POR ESTATUTOS COMO " WRITE " Y " DISPLAY ", LOS CUALES PUEDEN SER USADOS EN UNA ESTRUCTURA O MODO DE OPERACION DE REPORTE.
- UN EDITOR QUE GENERA SALIDAS ATRAVES DEL GENERADOR DE REPORTE, EL CUAL PERMITE DESARROLLAR REPORTE EN FORMA DE COLUMNA.

CADA METODO DE GENERACION DE REPORTE CONSTA DE UNA GAMA COMPLETA DE CONTROLES AUTOMATICOS QUE INCLUYEN:

- TITULO, TEXTO Y DATOS.
- TEXTO FINAL (PIE DE PAGINA) COMBINANDO TEXTO Y DATOS.
- CORTES DE PAGINA CON EVALUACION DE FUNCIONES ESTANDAR TALES COMO VALOR MAXIMO, VALOR MINIMO, MEDIA, SUMAS, CONTEOS Y VARIEDAD DE FUNCIONES CIENTIFICAS.
- CREACION DE ENCABEZAMIENTOS COLUMNARES USANDO LA INFORMACION DISPONIBLE DEL DICCIONARIO DE DATOS **PREDICT** O LA INFORMACION ALMACENADA POR EL USUARIO.
- MASCARAS DE EDICION DISPONIBLES EN **PREDICT** O EN LAS ESPECIFICACIONES DADAS POR EL USUARIO.
- FUNCIONES DEL SISTEMA COMO FECHA, HORA Y NUMERO DE PAGINA.

**E) EDITOR DE ESPECIFICACION DE PROGRAMAS** - ES USADO PARA DOCUMENTAR LOS REQUERIMIENTOS DEL USUARIO Y ESPECIFICACIONES DEL PROGRAMA EN EL DICCIONARIO DE DATOS **PREDICT**. ESTAS DEFINICIONES SERAN USADAS COMO BASE PARA TODA LA DOCUMENTACION DEL SISTEMA PARA EL PROCESO DE COMUNICACION ENTRE LA EVOLUCION DE LA APLICACION Y EL USUARIO FINAL.

NATURAL

AL INICIAR UNA SESION EN NATURAL, APARECE LA SIGUIENTE PANTALLA:

```
-----  
NEXT                                     LIB=SYSTEM  
HH:MM:SS                               SYSTEM MAINTENANCE LEVEL 0001      DD/MM/AA  
  
NNN  NN  AAAAAAAAAA  TTTTTTTT  UU    UU  RRRRRRRR  AAAAAAAAAA  LL  
NNNN  NN  AA    AA    TT    UU    UU  RR    RR  AA    AA  LL  
NN NN NN  AA    AA    TT    UU    UU  RR    RR  AA    AA  LL  
NN  NNNN  AAAAAAAAAA  TT    UU    UU  RRRRRRRR  AAAAAAAAAA  LL  
NN  NNN  AA    AA    TT    UU    UU  RR    RR  AA    AA  LL  
NN  NN  AA    AA    TT    UUUUUUUU  RR    RR  AA    AA  LLLLLLL  
  
111 222222  
11111 22 22  
111 22  
111 22  
111 22  
1111111 ( ) 222222
```

EL PROMPT "NEXT" INDICA QUE NATURAL HA SIDO INVOCADO Y ESTA LISTO PARA RECIBIR CUALQUIER COMANDO.

# NATURAL

## MENSAJES DEL PROMPT

ADEMAS DEL "NEXT" PUEDE HABER LOS SIGUIENTES PROMPTS DE ACUERDO A LAS CIRCUNSTANCIAS:

PROMPT	FUNCION
MORE	ESTE PROMPT APARECE DESPUES DE QUE UN PANTALLA DE SALIDA HA SIDO DESPLEGADA. EL MORE INFORMA QUE UNA SALIDA ADICIONAL ES DISPONIBLE DEL PROGRAMA QUE HA SIDO EJECUTADO . EL ENTER ES OPRIMIDO PARA DESPLEGAR LA SIGUIENTE PANTALLA.
*****	ESTE PROMPT APARECE SI UNA ENTRADA FISICA DE UNA PANTALLA DE SALIDA HA SIDO PRODUCIDA DESPUES DE LA COMPLETACION DE UN COMANDO O DE UN ESTATUTO DE ENTRADA/SALIDA. EL ENTER PUEDE SER USADO PARA OBTENER LA SIGUIENTE PANTALLA DE SALIDA.
ENTER	ES SIMILAR AL **** . SE PRESIONA ENTER PARA OBTENER LA SIGUIENTE PAGINA DE SALIDA. EN ESTE PROMPT SOLO SE PUEDE DAR ENTER. ESTE PROMPT ES DESPLEGADO CUANDO EL DISPOSITIVO FISICO DEL BUFFER TIENE UN OVERFLOW DURANTE LA EJECUCION DE UN ESTATUTO O COMANDO DE NATURAL.
ENTER INPUT DATA	ESTE PROMPT APARECE CUANDO LA EJECUCION DE UN ESTATUTO DE ENTRADA HA SIDO COMENZADO EN EL MODO PALABRA RESERVADA/DELIMITADORA.
ERROR....IN FIELD	UN ERROR HA SIDO DETECTADO DURANTE LA VALIDACION DE LA ENTRADA DE UN DATO DADO POR EL USUARIO EN EL MODO PALABRA RESERVADA/DELIMITADORA.
NAT....error text	UN ERROR HA SIDO DETECTADO DURANTE LA VALIDACION DE LA ENTRADA DE UN DATO DADO POR EL USUARIO EN EL MODO PANTALLA.
'text'	UN TEXTO DEFINIDO POR EL USUARIO PUEDE SER DESPLEGADO COMO RESULTADO DE LA EJECUCION DE UN ESTATUTO REINPUT.

# NATURAL

## SISTEMA DE CONTROL DE COMANDOS Y SINTAXIS

EL SIGUIENTE SISTEMA DE CONTROL DE COMANDOS ES UTILIZADO PARA EJECUTAR FUNCIONES RELACIONADAS CON LA CREACION, MANIPULACION, ALMACENAMIENTO Y EJECUCION DEL LOS PROGRAMAS DE NATURAL. AUNQUE LOS COMANDOS SON USADOS PRIMORDIALMENTE PARA MODO INTERACTIVO, HAY ALGUNOS COMANDOS QUE SON USADOS EN MODO BATCH.

### SINTAXIS DE LOS COMANDOS

LOS SIGUIENTES SIMBOLOS SON USADOS CON LOS COMANDOS DE NATURAL, PARA LA DESCRIPCION DE LA SINTAXIS.

SIMBOLO	SIGNIFICADO
[ ]	LOS ELEMENTOS CONTENIDOS EN LOS PARENTESIS SON OPCIONALES.
{ }	UNO DE LOS ELEMENTOS CONTENIDOS DEBE SER ESPECIFICADO.
...	EL ELEMENTO PUEDE SER REPETIDO.
abc	LETRAS MINUSCULAS REPRESENTAN INFORMACION QUE PUEDE SER REEMPLAZADA POR EL USUARIO.
ABC	LETRAS MAYUSCULAS INDICAN PALABRA RESERVADA DE NATURAL.

SISTEMA DE COMANDOS	FUNCION
FIN, GLOBALS, KEY, LOGOFF, LOGON, UPDATE	INICIALIZACION Y TERMINACION DE UNA SESION DE NATURAL.
ADHOC, CHECK, CREATE, EDIT, MAP, EDT	CREACION Y MODIFICACION DE PROGRAMAS.
CLEAR, PURGE, READ, RUN, SAVE	MANIPULACION DEL PROGRAMA FUENTE.
CATALOG, CATMAP, EXECUTE UNCATALOG	MANIPULACION DEL PROGRAMA OBJETO.

## NATURAL

SISTEMA DE COMANDOS	FUNCION
SCRATCH, STOW	MANIPULACION DEL PROGRAMA FUENTE Y DEL OBJETO.
LIST, LIST FILES, LIST FILE..., LIST MODULES, LIST PROGRAMS	LISTAN INFORMACION DE PROGRAMAS, MODULOS Y ARCHIVOS.
HELP	REFERENCIA EN MODO INTERACTIVO (AYUDA).

### COMANDOS

#### ADHOC

EL COMANDO ADHOC PERMITE CREAR PROGRAMAS LOS CUALES INMEDIATAMENTE DESPUES DE SER COMPILADOS Y EJECUTADOS SON BORRADOS Y NO PUEDEN SER SALVADOS PARA SU POSTERIOR MODIFICACION O EJECUCION.

#### CATALOG [nom-programa [id-libreria] ]

ESTE COMANDO ES USADO PARA COMPILAR UN PROGRAMA FUENTE EN EL AREA DE TRABAJO (Work Area) Y ALMACENAR EL RESULTADO EN UN MODULO OBJETO EN EL SISTEMA DE ARCHIVOS DE NATURAL. SI ES UN PROGRAMA NUEVO ENTONCES EL NOMBRE DEL PROGRAMA DEBE SER ESPECIFICADO.

#### CATMAP [nom-map [id-libreria] ]

ESTE COMANDO ES USADO PARA COMPILAR UN PROGRAMA EN EL AREA DE TRABAJO Y ALMACENAR EL RESULTADO EN UN MODULO OBJETO EN EL SISTEMA DE ARCHIVOS DE NATURAL COMO UN MAPA. PARA CATALOGAR UN PROGRAMA COMO UN MAPA, ESTE DEBE CONSISTIR DE SOLO UNA ENTRADA, UN FIN Y COMENTARIOS OPCIONALES.

#### CHECK

ESTE COMANDO ES UTILIZADO PARA REVISAR LA SINTAXIS DE UN PROGRAMA FUENTE QUE ACTUALMENTE SE ENCUENTRE EN EL AREA DE TRABAJO.

#### CLEAR

ESTE COMANDO LIMPIA EL AREA DE TRABAJO. EL CLEAR DEBE SER USADO SI UN NUEVO PROGRAMA VA A SER CREADO USANDO EL MODO EDIT Y HAY OTRO PROGRAMA USANDO EL AREA DE TRABAJO.



## NATURAL

### CREATE

EL CREATE PERMITE AL USUARIO CREAR UN PROGRAMA BAJO EL CONTROL DEL CHECADOR DE SINTAXIS.

EDIT { (nom-programa [id-libreria])  
VARIABLES }

ESTE COMANDO ES USADO PARA INVOCAR AL EDITOR DE PANTALLA COMPLETA Y ENTRAR AL MODO EDIT. EN ESTE MODO SE PUEDE MODIFICAR, BORRAR O INSERTAR LINEAS EN UN PROGRAMA EXISTENTE EN NATURAL. NO CHECA LA SINTAXIS. EL EDIT VARIABLES SE USA PARA EDITAR LAS VARIABLES GLOBALES ACTUALES.

### SUBCOMANDOS DE EDICION

EDITOR DE PANTALLA COMPLETA (Full Screen)

LOS SIGUIENTES COMANDOS DEBEN SER TECLEADOS EN LA LINEA DEL EDIT:

COMANDO	FUNCION
ENTER, P, +P	AVANZAN UNA PAGINA (20 LINEAS).
-, -P	RETROCEDE UNA PAGINA (20 LINEAS).
+n, -n	AVANZA O REGRESA n LINEAS.
n	SE VA A LA LINEA n .
B, ++	SE VA AL FINAL DEL PROGRAMA.
T, --	SE VA AL INICIO DEL PROGRAMA.
D H, D P	BORRA UNA PAGINA COMPLETA O MEDIA PAGINA (D P).
A, ADD	ADHIERE LINEAS AL FINAL DEL PROGRAMA.
N, RENUMBER	RENUMERA LA LINEAS DE 10 EN 10.
+H, -H	AVANZA O RETROCEDE 10 LINEAS.
S, SC, SCAN	EXPLORA UN STRING EN EL PROGRAMA, DONDE LO LOCALICE PONDRÁ UNA 'S' AL PRINCIPIO DE LA LINEA. EL STRING PUEDE SER REEMPLAZADO Y LO MARCARÁ CON UNA 'R'.

LOS SIGUIENTES COMANDOS PUEDEN SER TECLEADOS EN LA LINEA DEL EDIT O EN LA COLUMNA 1 DEL PROGRAMA FUENTE:

## NATURAL

COMANDO	FUNCION
.X, .Y	MARCA LINEAS CON ETIQUETAS 'X' Y 'Y'. SON USADAS PARA DELIMITAR LINEAS.
.D, ?, ERASE	BORRA LINEAS.
.DX, .DX-Y	
.I, >	INSERTA LINEAS.
.C, .CX, .CX-Y	COPIA UNA LINEA O LINEAS.
.M, .MX, .MX-Y	MUEVE UNA LINEA O LINEAS.
.J	UNE LA LINEA CON LA LINEA ANTERIOR.
.S	DIVIDE LA LINEA DE DONDE ESTE POSESIONADO EL CURSOR.

### EDT [ nom-programa [id-libreria] ]

EL EDT ES UN EDITOR DE LINEAS, SE CAPTURA EL PROGRAMA LINEA POR LINEA. SI EL EDT ES INVOCADO CON UN NOMBRE DE PROGRAMA Y EL AREA DE TRABAJO ESTA LIMPIA, EL USUARIO VERA DESPLEGADA EN SU PANTALLA LA LINEA NUMERO '0010'. TIENE UN CONJUNTO DE SUBCOMANDOS SIMILAR AL EDIT.

{EXECUTE [REPEAT] nom-programa [id-libreria]  
nom-programa [parametro...]}

ESTE COMANDO ES UTILIZADO PARA EJECUTAR UN MODULO OBJETO EN NATURAL (Transaction). LA EJECUCION DE UN MODULO OBJETO NO AFECTA EL PROGRAMA QUE SE ENCUENTRE EN EL AREA DE TRABAJO. EL REPEAT INDICA QUE SI EL PROGRAMA PRODUCE MULTIPLES PANTALLAS DE SALIDA, ENTONCES SE PODRAN DESPLEGAR UNA TRAS OTRA SIN INTERVENCION DE LOS PROMPTS.

### FIN

ESTE COMANDO ES PARA TERMINAR UN SESION DE NATURAL YA SEA EN MODO INTERACTIVO O EN MODO BATCH.

### GLOBALS [parametro...]

EL COMANDO DE GLOBALS ES UN SISTEMA DE COMANDOS EQUIVALENTE AL CONJUNTO DE ESTATUTOS GLOBALES.

LOS PARAMETROS PUEDEN SER PROPORCIONADOS EN CUALQUIER ORDEN Y DEBEN ESTAR SEPARADOS POR COMAS O POR BLANCOS. SI EL COMANDO GLOBALS ES TECLEADO SIN PARAMETROS ENTONCES UN MENU ES DESPLEGADO EL CUAL CONTIENE TODOS LOS VALORES DE LOS PARAMETROS ACTUALES. EL MENU PUEDE SER USADO PARA CAMBIAR LOS VALORES DE LOS PARAMETROS.

# NATURAL

## FACILIDAD DEL HELP

{ HELP } [ 'estatuto' ]  
 { ? } [ 'comando' ]  
 [ numero ]

EL COMANDO HELP HABILITA AL USUARIO A INVOCAR LA AYUDA EN MODO INTERACTIVO.

EL HELP DESPLEGARA EL MENU DE AYUDA.

EL HELP CON ESTATUTO O COMANDO DESPLEGARA LA INFORMACION ESPECIFICA DEL COMANDO O ESTATUTO.

EL HELP NUMERO (MAXIMO 4 DIGITOS) DESPLEGARA EN DETALLE EL MENSAJE QUE FUE MARCADO POR LA CONDICION DE ERROR DE NATURAL.

Ej.

\*\*\*\*\* NATURAL HELP FUNCTION \*\*\*\*\*

THE NATURAL HELP FUNCTION GIVES THE FOLLOWING INFORMATIONS:

NATURAL COMMANNDS.....: C  
 NATURAL STATMENTS.....: S  
 ARITHMETIC RULES.....: A  
 GLOBAL AND FORMAT PARAMETERS.....: P  
 STANDARD SYSTEM FUNCTIONS.....: F  
 DDM REPORT INTERPRETATION.....: D  
 ERROR MESSAGE INTERPRETATION.....: M

DESCRIPTION OF SYNTAXIS USED IN HELP FUNCTION...: X

END OF HELP FUNCTION.....: E

ESTA ES LA PANTALLA DEL MENU PRINCIPAL DEL HELP, PARA TERMINAR DE USAR LA FUNCION DE AYUDA TECLEAR UN 'E'.

KEY { ON  
 OFF } { { PAn } = { ON  
 OFF } } ... } n = 1-3 PARA PA KEYS  
 n = 1-24 PARA PF KEYS

ESTE COMANDO ES USADO PARA DEFINIR O CAMBIAR LOS VALORES DE LAS LLAVES ASOCIADAS CON LA ATENCION DEL PROGRAMA ( PA ) O FUNCIONES DEL PROGRAMA (PF) EN EL TABLERO DE LA TERMINAL, SE PUEDEN ASIGNAR VALORES EN LAS LLAVES PA QUE SEA VALIDAS EN EL SISTEMA NATURAL, CONTROL DE LA TERMINAL Y COMANDOS DEFINIDOS POR EL USUARIO.

## NATURAL

```
LIST { FILES  
      FILE nom-archivo  
      SHORT nom-archivo  
      PROGRAMS  
      [PROGRAM] nom-programa  
      MODULES  
      n[-m]  
      ? }
```

EL COMANDO LIST ES UNA HERRAMIENTA PARA LISTAR PROGRAMAS, ARCHIVOS Y MODULOS.

### LOGOFF

ESTE COMANDO CAUSA QUE EL ID-LIBRERIA (IDENTIFICADOR DE LIBRERIA) DEL SISTEMA Y EL PASSWORD DE ADABAS SEAN BLANCOS. LOGOFF NO TERMINA LA SESION. HAY QUE RECORDAR QUE LA SESION SE TERMINA CON EL COMANDO FIN O CON EL ESTATUTO TERMINATE. LOGOFF NO AFECTA LOS CONJUNTOS DE PARAMETROS GLOBALES.

### LOGON [id-libreria]

EL LOGON ES USADO PARA ESTABLECER EL IDENTIFICADOR DE LIBRERIA DEL USUARIO. LOGON CAUSA QUE TODAS LAS VARIABLES GLOBALES SEAN LIBERADAS.

### UTILIDAD DEL MAP

### MAP

ESTE COMANDO MAP INVOCA LA PANTALLA COMPLETA DE MAPEO. NO NECESITA PARAMETROS. DENTRO DEL MAP SE DESPLEGARA UN MENU DE OPCIONES PARA CREAR MAPAS. EN ESTE MENU DE OPCIONES SE PUEDE EDITAR, LISTAR, CATALOGAR, PURGAR (BORRAR EL MAPA), LEER UN MAPA CREADO, SALVAR, LIMPIAR EL AREA DE TRABAJO ( LLAMADA AREA FUENTE), etc. DENTRO DE ESTE MENU TAMBIEN SE PUEDE PEDIR AYUDA (HELP), EN DONDE SE EXPLICARAN CADA UNA DE LAS OPCIONES MAS EXTENSAMENTE.

# NATURAL

Ej.

\*\*\*\*\* ONLINE MAPPING UTILITY \*\*\*\*\*

LIB: SYSTEM

CODE	FUNCTION	MAP NAME
C	CATALOG MAP FROM THE SOURCE AREA	O
E	EDIT MAP IN THE SOURCE AREA	
L	LIST MAP IN THE SOURCE AREA	
P	PURGE SOURCE MAP	R
R	READ MAP INTO THE SOURCE AREA	R
S	SAVE SOURCE MAP	O
T	TEST MAP IN THE SOURCE AREA	
U	UNCATALOG MAP	R
X	CLEAR SOURCE AREA	
?	HELP	
.	TERMINATE MAPPING UTILITY	

ENTER FUNCTION CODE: ?  
MAP NAME:

MAP NAME R = REQUIRED  
O = OPTIONAL

## PURGE [nom-programa] ...

ESTE COMANDO ES USADO PARA BORRAR UNO O MAS PROGRAMAS FUENTE DEL SISTEMA DE ARCHIVOS DE NATURAL. SI NO SE DA NOMBRE DEL PROGRAMA ENTONCES SE DESPLEGARA UN MENU QUE CONTIENE TODOS LOS PROGRAMAS FUENTES EXISTENTES EN LA LIBRERIA, EL USUARIO PUEDE SELECCIONAR LOS QUE DESEA SEAN PURGADOS POR MEDIO DE UNA 'X' AL FINAL DEL NOMBRE DEL PROGRAMA.

## READ nom-programa [id-libreria]

ESTE COMANDO READ ES USADO PARA TRANSFERIR UN PROGRAMA FUENTE DEL SISTEMA DE ARCHIVOS DE NATURAL AL AREA DE TRABAJO.

## RUN [REPEAT] [nom-programa [id-libreria] ]

ESTE COMANDO COMPILA Y EJECUTA UN PROGRAMA FUENTE,  
EL REPEAT INDICA QUE SI EL PROGRAMA PRODUCE MULTIPLES PANTALLAS DE SALIDA, ENTONCES SE PODRAN DESPLEGAR UNA TRAS OTRA SIN INTERVENCION DE LOS PROMPTS.

## NATURAL

### SAVE [nom-programa [id-libreria] ]

ESTE COMANDO ES USADO PARA ALMACENAR UN PROGRAMA FUENTE EN EL SISTEMA DE ARCHIVOS DE NATURAL.

### SCRATCH [nom-programa...]

ESTE COMANDO ES USADO PARA BORRAR UNO O MAS PROGRAMAS (EL FUENTE Y EL OBJETO ) DEL SISTEMA DE ARCHIVOS DE NATURAL. SI ESTE COMANDO SE DA SIN NOMBRE DEL PROGRAMA ENTONCES SE DESPLEGARA UN MENU QUE CONTIENE TODOS LOS PROGRAMAS QUE TENGAN FUENTE Y OBJETO. EL USUARIO PODRA SELECCIONAR AQUELLOS PROGRAMAS QUE QUIERA BORRAR MARCANDOLOS CON UNA 'X' AL FINAL DEL NOMBRE DEL PROGRAMA.

### STOW [nom-programa [id-libreria] ]

ESTE COMANDO ES USADO PARA ALMACENAR UN PROGRAMA ( EL FUENTE Y EL OBJETO) EN EL SISTEMA DE ARCHIVOS DE NATURAL.

### UNCATALOG [nom-programa...]

ESTE COMANDO ES USADO PARA BORRAR UNO O MAS PROGRAMAS OBJETO DEL SISTEMA DE ARCHIVOS DE NATURAL. SI NO SE DA NOMBRE DEL PROGRAMA ENTONCES SE DESPLEGARA UN MENU QUE CONTIENE TODOS LOS PROGRAMAS OBJETO EXISTENTES EN LA LIBRERIA, EL USUARIO PUEDE SELECCIONAR LOS QUE DESEA SEAN BORRADOS POR MEDIO DE UNA 'X' AL FINAL DEL NOMBRE DEL PROGRAMA.

### UPDATE { ON OFF }

EL UPDATE ES USADO PARA HABILITAR O DESHABILITAR LA ACTUALIZACION DE LA BASE DE DATOS. ESTE COMANDO NO TIENE EFECTO EN LA EJECUCION DE LOS COMANDOS: CATALOG, CATMAP, PURGE, SAVE, STOW Y UNCATALOG. EL UPDATE ON SERA IGNORADO SI EL DBA HA SIDO DESHABILITADO DURANTE LA INSTALACION DE NATURAL, DE OTRO MODO FUNCIONARA. EL UPDATE OFF DESHABILITARA LA ACTUALIZACION NORMAL DE LOS ESTATUTOS UPDATE, STORE Y DELETE. LOS PROGRAMAS QUE CONTENGAN ESTOS ESTATUTOS SE EJECUTARAN NORMALMENTE, PERO NO MODIFICARAN LA BASE DE DATOS.

## COMANDOS PARA EL CONTROL DE LA TERMINAL

LOS SIGUIENTES COMANDOS SON USADOS PARA TENER UNA COMUNICACION DIRECTA ENTRE EL USUARIO DE NATURAL Y EL MANEJADOR DEL MODULO TP. SON VALIDAS UNICAMENTE EN MODO INTERACTIVO.

## NATURAL

COMANDO

FUNCION

-----  
COMANDOS DE INTERRUPCION:

%% CUANDO SON TECLEADOS EN RESPUESTA A UN NUMERO DE LINEA, UN 'MORE' O \*\*\*\* PROMPT, O 'ENTER INPUT DATA' O EL PRIMER CAMPO DE UN MAPA, CAUSA QUE LA OPERACION EN PROGRESO SEA INTERRUMPIDA. NATURAL ENTRARA AL MODO COMANDO Y DESPLEGARA EL 'NEXT'.

CLEAR TIENE EL MISMO EFECTO QUE EL %%; PERO ESTA PUEDE SER TECLEADA EN CUALQUIER CAMPO EN EL MAPA. USANDOLO EN RESPUESTA DEL NEXT SE DA POR TERMINADA LA SESION.

CTRL/D TIENE EL MISMO EFECTO QUE EL CLEAR USANDO UNA TERMINAL 3270.

RESET UNICAMENTE EN AMBIENTE COM-PLETE. USANDO ESTA LLAVE CAUSA LA TERMINACION DE UN LOOP. EL LOOP DEBE CONTENER UN MINIMO DE UN ACCESO A LA BASE DE DATOS Y NO PRODUCIR NINGUNA SALIDA.

COMANDOS DE ENTRADA/SALIDA:

%\* CUANDO SON TECLEADOS EN RESPUESTA A UN NUMERO DE LINEA, UN 'MORE' O \*\*\*\* PROMPT, O 'ENTER INPUT DATA' O EL PRIMER CAMPO DE UN MAPA, CAUSA QUE LOS CARACTERES QUE VAN A SER ESCRITOS EN EL SIGUIENTE CAMPO, LINEA, COMANDO EN LINEA, O PANTALLA, NO APAREZCAN EN LA PANTALLA. PARA REGRESAR AL ESTADO NORMAL ( VER LOS CARACTERES ), TECLEAR ENTER.

%H EN AMBIENTE COM-PLETE UNICAMENTE. PRODUCE UNA SALIDA (EN IMPRESION), DESPUES DE NECESITA DAR EL CODIGO DE DESTINO. SU DEFAULT SON PAGINAS DE 80 LINEAS, PERO PUEDEN SER CAMBIADAS POR MEDIO DE LAS VARIABLES GLOBALES (SET GLOBALS) O DEL FORMAT

%L CAUSA QUE LAS LETRAS MINUSCULAS NO SEAN TRASLADADAS A LETRAS MAYUSCULAS POR NATURAL.

%U TRASLADA LAS LETRAS MINUSCULAS A LETRAS MAYUSCULAS. ESTE ES EL CASO DE DEFAULT.



## NATURAL

### COMANDOS DE CONTROL DE ESTATUTOS DE ENTRADA

- %D CAUSA QUE EL MODO PALABRA RESERVADA/DELIMITADORA SEA ACTIVADO. EN MODO INTERACTIVO SOLO PUEDE SER PERMITIDO EN RESPUESTA A UN MENSAJE DEL PROMPT.
- %F CAUSA QUE EL MODO FORMA/PANTALLA SEA ACTIVADO. EN MODO INTERACTIVO SOLO PUEDE SER PERMITIDO EN RESPUESTA A UN MENSAJE DEL PROMPT.
- %R CAUSA QUE EL ESTATUTO DE ENTRADA QUE ESTA EN EL BUFFER SEA LIMPIADO, CUANDO UN MENSAJE DE ERROR ES GENERADO DURANTE LA EJECUCION DE UN ESTATUTO DE ENTRADA. ESTO PERMITE EL REPROCESO DE UNA OPERACION DE ENTRADA DESDE EL PRINCIPIO.

### COMANDOS DE PAGINACION DE PANTALLAS:

- %E PROCESO DE ENTRADA DE PAGINAS. LOS SIGUIENTES SUBCOMANDOS PUEDEN SER TECLEADOS:
- TOP (T) SE POSESIONA AL PRINCIPIO DEL CONJUNTO DE DATOS.
- BOT (B) SE POSESIONA AL FINAL DEL CONJUNTO DE DATOS.
- n REENUMERA LOS NUMEROS DE LA PANTALLA.
- i MUEVE i PAGINAS ATRAS O ADELANTE.
- E TERMINA LA PAGINA PROCESADA Y DESPLIEGA LA ULTIMA PAGINA.
- %I COPIA LA PANTALLA INMEDIATAMENTE.
- %O TERMINA LA CREACION DE LA PAGINA.
- %P CREA UN PANTALLA NUEVA PARA EL CONJUNTO DE DATOS.
- %S SE MANDA TODOS LOS DATOS DE LA PANTALLA AL CONJUNTO DE DATOS, SI NO EXISTE LO CREA.

# NATURAL

## SINTAXIS DE LOS ESTATUTOS DE NATURAL

LOS SIGUIENTES SIMBOLOS SON USADOS CON LAS INSTRUCCIONES DE NATURAL, PARA LA DESCRIPCION DE LA SINTAXIS.

SIMBOLO	SIGNIFICADO
[ ]	LOS ELEMENTOS CONTENIDOS EN LOS PARENTESIS SON OPCIONALES.
{ }	UNO DE LOS ELEMENTOS CONTENIDOS DEBE SER ESPECIFICADO.
...	EL ELEMENTO PUEDE SER REPETIDO.
=	EQUIVALENCIA.
abc	LETRAS MINUSCULAS REPRESENTAN INFORMACION QUE PUEDE SER REEMPLAZADA POR EL USUARIO.
ABC	LETRAS MAYUSCULAS INDICAN PALABRA RESERVADA DE NATURAL.
(x)	INDICA EL NUMERO DE REPORTE.
'texto'	EL TEXTO PUEDE SER USADO EN INSTRUCCIONES DE INPUT, DISPLAY, WRITE, WRITE TITLE, WRITE TRAILER, QUE CONTIENE UN MAXIMO DE 72 CARACTERES Y NO PUEDE CONTINUAR EN LA LINEA SIGUIENTE. SI SE DESEAN COMILLAS UTILIZAR EL CARACTER ".
'a' (n)	'a' CHARACTER (n) NUMERO DE VECES QUE EL CHARACTER VA A SER REPETIDO. MAXIMO 132 CARACTERES.
[*] std-sf.	FUNCION STANDARD.
[(r,f1/i)]	r 4 DIGITOS DEL NUMERO DEL LOOP AL CUAL ES REFERIDO.  f1 REPRESENTA EL FORMATO Y LONGITUD DEL CAMBIO DE LA BASE DE DATOS O VARIABLE DEFINIDA POR EL USUARIO.  i ES USADA PARA INDICAR EL NUMERO DE OCURRIENCIAS DE UN CAMPO PERIODICO O MULTIPLE  , LA COMA ES REQUERIDA EN LA SINTAXIS.

# NATURAL

## SINTAXIS DEL CRITERIO DE CONDICION LOGICA

EL CRITERIO DE CONDICION LOGICA DEBE SER USADO EN LAS SIGUIENTES INSTRUCCIONES: FIND, READ, HISTOGRAM, ACCEPT/REJECT, IF, Y REPEAT.

LA CLAUSULA WHERE CONTIENE EL CRITERIO DE CONDICION LOGICA Y ES USADO CON EL FIND, READ E HISTOGRAM.

LA CLAUSULA IF ES USADA CON EL CRITERIO DE CONDICION LOGICA, QUE ES USADO CON EL IF Y CON EL ACCEPT/REJECT.

LA CLAUSULA UNTIL TAMBIEN UTILIZA CRITERIO DE CONDICION LOGICA Y ES UTILIZADO POR EL REPEAT.

## CONECTORES DEL CRITERIO DE CONDICION LOGICA

PUEDE SER CONECTADO POR LOS SIGUIENTES OPERADORES: AND, OR Y NOT.

criterio - [NOT] [ ( ) criterio [ ( AND O OR ) [NOT] criterio ] ( ) ] ...  
condición - [ ( ) condición [ ( AND O OR ) [NOT] condición ] ( ) ] ...  
lógica - [ ( ) lógica [ ( AND O OR ) [NOT] lógica ] ( ) ] ...

ORDEN DE PRIORIDAD DE LOS OPERADORES: ( )  
NOT  
AND  
OR

## CRITERIO DE CONDICION LOGICA

(1) campo [(f1/i)] { =  
EQ  
EQUAL TO } { constante  
campo[(r, f1/i)] } [ { OR = { constante  
campo[(r, f1/i)] } }  
THRU { constante  
campo[(r, f1/i)] } } ]

...

[ BUT NOT { constante  
campo[(r, f1/i)] } ] THRU { constante  
campo[(r, f1/i)] } ]

# NATURAL

## CRITERIO DE CONDICION LOGICA (continuación)

(2) campo [(f1/i)]

}	¬ =	}	{ constante campo[(r,f1/i)] }
	NE		
	NOTEQUAL		
	<		
	LT		
	LESS THAN		
	< =		
	LE		
	LESS EQUAL		
	>		
GT			
GREATER THAN			
> =			
GE			
GREATER EQUAL			

### OPCIONES ESPECIALES:

campo [(f1/i)] { =  
EQ  
EQUAL TO  
NE  
NOT EQUAL } { MASK (mask-def) [ { constante  
campo[(r,f1/i)] } ] }  
SCAN constante  
campo[(r,f1/i)] }

BREAK [OF] campo[(r,f1/i)] [/n/]

LA OPCION MASK HABILITA SELECCIONAR POSICIONES DE UN VALOR A SER CHECADO.

Mask-def ES LA DEFINICION DE LA MASCARA QUE SE DESEA.

/n/ INDICA QUE SOLO VERIFICARA LAS PRIMERAS n POSICIONES DE UN CAMPO.



## NATURAL

[AT] TOP [OF] [PAGE] [(x)] { estatuto  
DO estatuto ... DOEND }

EL AT TOP OF PAGE ESPECIFICA UN PROCESO ESPECIAL QUE SE EJECUTARA CUANDO SE INICIE UNA PAGINA.

### BACKOUT TRANSACTION

ESTE ESTATUTO ES USADO PARA REGRESAR LOS DATOS A SU FORMA ORIGINAL DURANTE LA ACTUALIZACION DE UNA TRANSACCION LOGICA. EL ESTATUTO ES LIBERADO DEL ESTADO DE ESPERA (HOLD).

BEFORE [BREAK] [PROCESSING] estatuto  
DO estatuto ... DOEND

ESTE ESTATUTO HABILITA A EJECUTAR EL AT BREAK.

CALL 'nom-programa' { constante  
campo[(r,f1/i)] } ... 40

EL CALL ES USADO PARA LLAMAR PROGRAMAS REALIZADOS POR EL USUARIO DE UN PROGRAMA DE NATURAL.

CALL FILE 'nom-programa' campo-cntrl 1er { FILLER nX  
campo campo[(f1)] } ...  
en el constante }  
registro.

ESTE ESTATUTO ES USADO PARA LLAMAR PROGRAMAS REALIZADOS POR EL USUARIO LOS CUALES LEEN UN REGISTRO DE UN ARCHIVO NO-ADABAS Y TRAE EL REGISTRO AL PROGRAMA DE NATURAL PARA EFECTUAR ALGUN PROCESO.

CALL LOOP 'nom-programa' { campo[(r,f1/i)] } ... 40  
constante

ESTE ESTATUTO ES USADO PARA GENERAR UN PROCESO DE LOOP CONTENIENDO PROGRAMAS HECHOS POR EL USUARIO.

COMPRESS { constante  
campo[(r,f1/i)] } ... { TO } campo[(r,f1/i)] { LEAVING NO [SPACE] }  
INTO

EL COMPRESS ES USADO PARA COMBINAR LOS CONTENIDOS DE DOS O MAS OPERANDOS EN UN CAMPO.

[COMPUTE [ROUNDED] ] campo[(r,f1/i)] [=campo[(r,f1/i)]]... = E.A.

EL COMPUTE ES UTILIZADO PARA EJECUTAR OPERACIONES ARITMETICAS.  
 LAS E.A. ( EXPRESIONES ARITMETICAS ) CONSISTEN DE NUMEROS, CAMPOS NUMERICOS, VARIABLES QUE PUEDEN ESTAR CONECTADAS POR LOS SIGUIENTES OPERADORES NUMERICOS : ( ) HASTA OCHO NIVELES, \*\* EXPONENCIACION, \* MULTIPLICACION, / DIVISION, + ADICION Y - SUBSTRACION.

DEFINE SUBROUTINE nom-subrutina estatuto ... RETURN

ESTE ESTATUTO DEFINE UNA SUBROUTINA LA CUAL ES INVOCADA ATRAVES DE UN ESTATUTO PERFORM. REGRESA EL CONTROL A LA SIGUIENTE INSTRUCCION DE DONDE FUE LLAMADA CUANDO SE ENCUENTRA EL RETURN.

DELETE [RECORD] [IN] [STATEMENT] [(r)]

EL ESTATUTO DELETE ES USADO PARA BORRAR UN REGISTRO DEL ARCHIVO ADABAS.

DISPLAY [(x)] { NOTITLE  
NOHDR  
NOTITLE NOHDR } [[AND] [GIVE] [SYSTEM] FUNCTIONS]

[(FORMAT)]

{ [elem-salida ...][/...]  
 { [nX] { 'text' [(I)] }  
 { [nT] { 'a' (n)[(I)] } } { { VERT  
VERTICALLY AS { 'text' [(I)][CAP] }  
CAP  
CAPTIONED } [/...] elem [/...] } }  
 { [HORIZ  
HORIZONTALLY] } elem-salida... [/...] } }

elem-salida = { [nX  
nT  
{ 'text' [(I)] }  
{ 'a' (n)[(I)] } } [ '=' ] { campo[(r,f1/i)] [(formato)]  
[\*]std-sf[(campo)][(r,f1/i)][(formato)]  
grupo [(r,f1/i)] [(formato)] }

EL DISPLAY ESPECIFICA LOS CAMPOS DE SALIDA EN UN REPORTE EN FORMA DE COLUMNAS. CAMPOS DE LA BASE DE DATOS, VARIABLES Y FUNCIONES STANDARD PUEDEN SER ESPECIFICADAS.

DENTRO DE LA INSTRUCCION DISPLAY SE PUEDEN UTILIZAR PARAMETROS, ELEMENTOS DE PARAMETROS, FORMATOS PARA IMPRESION, POSICIONAMIENTOS EN LINEA, MASCARAS DE EDICION TANTO PARA CAMPOS NUMERICOS COMO ALFANUMERICOS Y TAMBIEN PARA NUMEROS HEXADECIMALES.

NATURAL



ESCUELA DE CIENCIAS  
FISICO-MATEMATICAS  
BIBLIOTECA

PARAMETROS DEL ESTATUTO DISPLAY:

AD DEFINICION DE ATRIBUTO. (I, N, B, D Y L, R) .  
DEFAULTL DR.  
AL LONGITUD ALFANUMERICA 1-99.  
EM MASCARA DE EDICION. OFF.  
ES SUPRIMIR LINEA EN BLANCO. ON O OFF. DEFAULT OFF.  
FC CARACTER DE RELLENO. DEFAULT ES UN ESPACIO.  
GC GRUPO DE CARACTERES DE RELLENO. DEFAULT ESPACIO.  
HC CENTRADO DE ENCABEZADO C=CENTRO,L=IZQ. DEFAULT C.  
HW ANCHO DEL ENCABEZADO. ON O OFF. DEFAULT ON.  
IC CARACTERES DE INSERCION. HASTA 10 CARACTERES.  
IS SUPRIME CARACTER IDENTICO. ON O OFF. DEFAULT OFF.  
LC CARACTER PRINCIPAL. HASTA 10 CARACTERES.  
LS LONGITUD DE LA LINEA (1-250). DEFAULT 132.  
MC CONTADOR DE UN CAMPO MULTIPLE (1-191).DEFAULT 1.  
MP MAXIMO NUMERO DE PAGINAS (1-99999).  
NL LONGITUD NUMERICA (nn.m). nn+m <= 27,m<=7.  
PC CONTADOR DE GRUPO PERIODICO (1-99). DEFAULT 1  
PS LONGITUD DE LA PAGINA (1-250). DEFAULT 60  
SF FACTOR DE ESPACIAMIENTO (1-30). DEFAULT 1  
SG POSICION DEL SIGNO . ON O OFF. DEFAULT ON.  
TC CARACTER ULTIMO. HASTA 10 CARACTERES.  
UC CARACTER DEBAJO DE LA LINEA. DEFAULT -.  
ZP IMPRESION DEL CERO. ON O OFF. DEFAULT OFF.

PARAMETROS DE LOS ELEMENTOS DEL ESTATUTO DISPLAY:

AD DEFINICION DE ATRIBUTO. (I, N, B, D Y L, R) .  
DEFAULTL DR.  
AL LONGITUD ALFANUMERICA 1-99.  
EM MASCARA DE EDICION. OFF.  
FC CARACTER DE RELLENO. DEFAULT ES UN ESPACIO.  
GC GRUPO DE CARACTERES DE RELLENO. DEFAULT ESPACIO.  
HC CENTRADO DE ENCABEZADO C=CENTRO,L=IZQ. DEFAULT C.  
HW ANCHO DEL ENCABEZADO. ON O OFF. DEFAULT ON.  
IC CARACTERES DE INSERCION. HASTA 10 CARACTERES.  
IS SUPRIME CARACTER IDENTICO. ON O OFF. DEFAULT OFF.  
LC CARACTER PRINCIPAL. HASTA 10 CARACTERES.  
MC CONTADOR DE UN CAMPO MULTIPLE (1-191).DEFAULT 1.  
NL LONGITUD NUMERICA (nn.m). nn+m <= 27,m<=7.  
PC CONTADOR DE GRUPO PERIODICO (1-99). DEFAULT 1  
SG POSICION DEL SIGNO . ON O OFF. DEFAULT ON.  
TC CARACTER ULTIMO. HASTA 10 CARACTERES.  
UC CARACTER DEBAJO DE LA LINEA. DEFAULT -.  
ZP IMPRESION DEL CERO. ON O OFF. DEFAULT OFF.

LA OPCION GIVE SYSTEM FUNCTIONS ES USADA CUANDO EL SISTEMA DE FUNCIONES ESTANDAR SEA EVALUADO CADA VEZ QUE EL DISPLAY VA A SER EJECUTADO.



## NATURAL

### NOTACION DEL FORMATO DE LOS REPORTES:

LA SIGUIENTE TABLA LISTA NOTACIONES ADICIONALES QUE PUEDEN SER ESPECIFICADAS EN EL ESTATUTO DISPLAY:

nX	CAUSA n ESPACIOS A SER INSERTADOS ENTRE LOS ELEMENTOS DE SALIDA O COLUMNAS.
nT	CAUSA TABULACION PARA IMPRESION EN LA POSICION n.
'text' (I)	CAUSA QUE EL 'text' SE IMPRIMA EN DETALLE EN LA LINEA Y (I) INTENSIFICA EL TEXTO A SER DESPLEGADO.
'text' '='	CON 'text' SE IMPRIMIRA EL TEXTO ANTES DEL CAMPO Y CON '=' SE IMPRIMIRA EL ENCABEZADO DE DEFAULT.
'a' (n)	IDENTICO QUE 'text' EXCEPTO QUE EL CARACTER 'a' ES DESPLEGADO n VECES (n=1-132).
/	CAUSA EL AVANCE DE UNA LINEA.
-	CAUSA QUE EL PROXIMO ELEMENTO SEA PUESTO EN LA MISMA LINEA DE LA MISMA COLUMNA SOLO SI EL VERT ES USADO.

### MASCARAS DE EDICION:

9	CARACTER IMPRIMIDO
Z	SUPRIME LOS CEROS DE UN CAMPO NUMERICO. LA Z NO PUEDE SER ESPECIFICADA A LA DERECHA DE UN PUNTO DECIMAL.
+	IMPRIME EL SIGNO DE UN NUMERO FLOTANTE .
-	IMPRIME EL SIGNO DE MENOS DE UN NUMERO FLOTANTE SI EL VALOR ES NEGATIVO.
S	IMPRIME EL SIGNO COMO CARACTER PRINCIPAL.
N	IMPRIME UN SIGNO DE MENOS COMO CARACTER PRINCIPAL.
.	REPRESENTA UNA POSICION DECIMAL.
H	ES USADA PARA LAS MASCARAS DE LOS HEXADECIMALES.

PARA CAMPOS NUMERICOS SE PERMITEN '9' Y 'N'.

PARA CAMPOS ALFANUMERICOS SE PERMITEN 'X' Y '-' (PARA REPRESENTAR UN BLANCO).

PARA CAMPOS HEXADECIMALES SOLO 'H', SIN IMPORTAR COMO ESTE DEFINIDO EL CAMPO (ALFANUMERICO O NUMERICO).

## NATURAL

(1) DIVIDE [ROUNDED]  $\left\{ \begin{array}{l} \text{constante numerica} \\ \text{campo}[(r, f1/i)] \end{array} \right\}$  INTO campo[(r, f1/i)]  
(2) DIVIDE [ROUNDED]  $\left\{ \begin{array}{l} \text{constante numerica} \\ \text{campo}[(r, f1/i)] \end{array} \right\}$  INTO  $\left\{ \begin{array}{l} \text{constante numerica} \\ \text{campo}[(r, f1/i)] \end{array} \right\}$   
[[GIVING campo[(r, f1/i)]]]

ESTE ESTATUTO DIVIDE ES PARA DIVIDIR DOS OPERANDOS. YA SEA DIVIDIENDO Y DEJANDO EL RESULTADO EN EL SEGUNDO CAMPO O BIEN DEJANDO EL VALOR EN OTRO CAMPO.

### DO estatuto ... DOEND

EL DO Y DOEND SON USADOS PARA ESPECIFICAR UN GRUPO DE ESTATUTOS QUE VAN A SER EJECUTADOS BASADOS EN UNA CONDICION LOGICA.

EJECT  $\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\} [(x)]$

ES USADO EN MODO BATCH O MODO INTERACTIVO PARA CONTROLAR EL CAMBIO DE PAGINA.

EJECT [(x)]  $\left[ \left[ \begin{array}{l} \text{IF} \\ \text{WHEN} \end{array} \right] [\text{LESS}] [\text{THAN}] n [\text{LINES}] [\text{LEFT}] \right]$

ESTE ESTA FORMA DE EJECT ES USADA EN MODO BATCH Y TP PARA CAUSAR EL AVANCE DE PAGINA SIN TITULO O LINEA DE ENCABEZADO AL GENERAR LA SIGUIENTE PAGINA.

$\left\{ \begin{array}{l} \text{END} \\ . \end{array} \right\}$

EL END O . (Punto) SON USADOS PARA INDICAR QUE EL PROGRAMA ESCRITO EN NATURAL HA SIDO TERMINADO.

### ENDHOC

ESTE ESTATUTO ES USADO PARA INDICAR QUE EL PROGRAMA FUENTE HA SIDO CREADO EN ADHOC.

END [OF] TRANSACTION  $\left\{ \begin{array}{l} \text{campo}[(r, f1/i)] \\ \text{constante} \end{array} \right\} \dots 40$

EL END OF TRANSACTION ES USADO PARA INDICAR EL FIN DE UNA TRANSACION LOGICA. ES UTILIZADA PARA LA ACTUALIZACION DE LA BASE DE DATOS.

# NATURAL

## ESCAPE

EL ESCAPE TERMINA LA EJECUCION DE UN PROCESO LOOP CUANDO UNA CONDICION LOGICA ES SATISFECHA.

EXAMINE campo[(r,f1/i)][FOR] { literal  
campo[(r,f!/i)] } { ABSOLUTE  
WITH[DELIMITERS][ 'del-c' ] }  
[ AND ] REPLACE [ FIRST ] [ WITH ] { literal  
campo[(r,f!/i)] } [ { GIVING NUMBER } [ IN ]  
NUMBER ]  
campo[(r,f1/i)]

ESTE ESTATUTO ES USADO PARA EXPLORAR UN STRING EN EL CONTENIDO DE UN CAMPO ALFANUMERICO Y REMPLAZARLO O CONTAR EL NUMERO DE OCURRENCIAS DE ESE STRING.

FETCH [ REPEAT ] { 'nom-programa'  
campo[(r,f1/i)] } [ campo[(r,f1/i)]  
constante ] ...

ESTE ESTATUTO SE UTILIZA PARA CARGAR UN PROGRAMA OBJETO DE NATURAL DEL SISTEMA DE ARCHIVOS DE NATURAL. EN EL FETCH SE PUEDEN PASAR PARAMETROS AL PROGRAMA QUE VA SER CARGADO. EL PROGRAMA RECEPTOR DEBERA, EN CASO DE TRANSFERENCIA DE PARAMETROS CONTAR CON UN ESTATUTO INPUT AL INICIO DEL PROGRAMA.

## FIN

EL ESTATUTO FIN ES PARA TERMINAR LA SESION DE NATURAL.

FIND [ { ALL  
(n)  
(campo[(r,f1/i)] ) } ] [ RECORDS ] [ IN ] [ FILE ] { nom-archivo  
num-archivo } [ PASSWORD =  
'password'  
campo[(r,f1/i)] ] [ CIPHER = código cipher ] [ WITH ] [ [ LIMIT ] { campo[(r,f!/i)]  
(n) } ]  
criterio básico de [ { AND } COUPLED [ TO ] [ FILE ] { nom-archivo  
num-archivo } [ WITH ] criterio básico ] ... 4  
búsqueda [ { OR } ] búsqueda ]  
[ SORTED ] [ BY ] descriptor ... 3 [ DESCENDING ] [ RETAIN AS { set-nom  
'set-nom' } ] [ WHERE  
criterio condición ] [ IF NO [ RECORDS ] [ FOUND ] ENTER  
lógica { estatuto  
DO estatuto ... DOEND } ]

## NATURAL

### OPCIONES ESPECIALES:

FIND  $\left\{ \begin{array}{l} \text{FIRST} \\ \text{NUMBER} \\ \text{UNIQUE} \end{array} \right\}$  No Generan Loop

ESTE ESTATUTO ES USADO PARA SELECCIONAR UN CONJUNTO DE REGISTROS DE UN ARCHIVO ADABAS.

### CRITERIO BASICO DE BUSQUEDA

EL CRITERIO ES USADO PARA IDENTIFICAR UN CONJUNTO DE REGISTROS A SER SELECCIONADOS. SOLAMENTE LOS CAMPOS DE LA BASE DE DATOS QUE ESTAN DEFINIDOS COMO DESCRIPTORES, SUBDESCRIPTORES, SUPERDESCRIPTORES O DESCRIPTORES FONETICOS PUEDEN SER USADOS EN LA CONSTRUCCION DEL CRITERIO BASICO DE BUSQUEDA.

criterio - [NOT]  $\left[ \left( \begin{array}{l} \text{criterio} \\ \text{básico de} \\ \text{búsqueda} \end{array} \right) \left\{ \begin{array}{l} \text{AND} \\ \text{OR} \end{array} \right\} \left[ \text{NOT} \right] \begin{array}{l} \text{criterio} \\ \text{básico de} \\ \text{búsqueda} \end{array} \left[ \right] \right] \dots$

### Criterio básico de búsqueda:

(1) descriptor [(i)]  $\left\{ \begin{array}{l} = \\ \text{EQ} \\ \text{EQUAL TO} \end{array} \right\} \left\{ \begin{array}{l} \text{constante} \\ \text{campo}[(r, f1/i)] \end{array} \right\} \left[ \left\{ \begin{array}{l} \text{OR} = \left\{ \begin{array}{l} \text{constante} \\ \text{campo}[(r, f1/i)] \end{array} \right\} \\ \text{THRU} \left\{ \begin{array}{l} \text{constante} \\ \text{campo}[(r, f1/i)] \end{array} \right\} \end{array} \right\} \dots \right]$

$\left[ \text{BUT NOT} \left\{ \begin{array}{l} \text{constante} \\ \text{campo}[(r, f1/i)] \end{array} \right\} \right]$

(2) descriptor [(i)]  $\left\{ \begin{array}{l} \neg = \\ \text{NE} \\ \text{NOTEQUAL} \\ < \\ \text{LT} \\ \text{LESS THAN} \\ < = \\ \text{LE} \\ \text{LESS EQUAL} \\ > \\ \text{GT} \\ \text{GREATER THAN} \\ > = \\ \text{GE} \\ \text{GREATER EQUAL} \end{array} \right\} \left\{ \begin{array}{l} \text{constante} \\ \text{campo}[(r, f1/i)] \end{array} \right\}$

(3) set-nom

## NATURAL

LA CLAUSULA WHERE ES USADA PARA ESPECIFICAR UN CRITERIO ADICIONAL EL CUAL ES EVALUADO DESPUES DE QUE EL REGISTRO ES SELECCIONADO POR MEDIO DE WITH.

LA CLAUSULA COUPLED ES USADA PARA ESPECIFICAR UNA BUSQUEDA QUE ENVUELVE EL ACOPLAMIENTO DE ADABAS. ESTA FACILIDAD PERMITE EL USO DE DESCRIPTORES DE LA BASE DE DATOS DE DIFERENTES ARCHIVOS ESPECIFICADOS EN UN CRITERIO DE BUSQUEDA DE UN COMANDO FIND.

LA CLAUSULA SORTED CAUSA QUE ADABAS ORDENE LOS REGISTROS SELECCIONADOS BASADO EN LA SECUENCIA DE UNO A TRES DESCRIPTORES.

UN CONJUNTO DE REGISTROS SELECCIONADOS CON EL WITH PUEDEN SER RETENIDOS PARA UN PROCESO FUTURO CON UNA LISTA DE ISN, POR MEDIO DE LA CLAUSULA RETAIN.

LA CLAUSULA IF NO RECORDS FOUND ES USADA PARA CAUSAR LA EJECUCION DE UN PROCESO CUANDO NO SE ENCUENTRAN REGISTROS POR EL CRITERIO ESPECIFICADO.

EL ESTATUTO FIND NUMBER DETERMINA EL NUMERO DE REGISTROS QUE SATISFACEN EL CRITERIO ESPECIFICADO.

EL ESTATUTO FIND UNIQUE ES USADO PARA ASEGURAR QUE SOLO UN REGISTRO ES SELECCIONADO POR EL PROCESO.

EL ESTATUTO FIND FIRST ES EJECUTADO UNA VEZ PARA LOCALIZAR EL PRIMER REGISTRO QUE CUMPLA CON EL CRITERIO WITH Y WHERE.

## NATURAL

SISTEMA DE VARIABLES DISPONIBLES CON EL ESTATUTO FIND:

- \*ISN - CONTIENE EL NUMERO DE SECUENCIA INTERNO DEL REGISTRO QUE ESTA SIENDO ACTUALIZADO. ESTA VARIABLE NO ESTA DISPONIBLE EN EL ESTATUTO FIND NUMBER.
- \*NUMBER - CONTIENE LA CANTIDAD DE REGISTROS QUE SATISFACEN EL CRITERIO BASICO ESPECIFICADO EN LA CLAUSULA WITH.
- \*COUNTER - CONTIENE EL NUMERO DE VECES QUE EL PROCESO HA SIDO LLAMADO.

FOR campo [(r,f1/i)] [ { =  
EQ  
FROM } ] { constante  
campo[(r,f1/i)] } [ { TO  
THRU } ] { constante entera  
campo[(r,f1/i)] }  
[ STEP { constante entera  
campo[(r,f1/i)] } ] estatuto ...

EL FOR ES USADO PARA INICIALIZAR UN PROCESO LOOP Y PARA CONTROLAR EL NUMERO DE VECES QUE UN LOOP VA SER PROCESADO. EL FOR DEBE SER CERRADO CON UN ESTATUTO LOOP O END.

FORMAT [(x)] parametro ...

EL FORMAT ES USADO PARA ESPECIFICAR PARAMETROS DE ENTRADA Y SALIDA. UTILIZA LOS MISMOS PARAMETROS QUE EL DISPLAY Y ADEMAS EL 'IP' QUE ES EL PROMPT DE ENTRADA DE UN TEXTO.

GET [IN] [FILE] { nom-archivo  
num-archivo } [ PASSWORD = { 'password'  
campo[(r/i)] } ] [ CIPHER =  
{ código cipher  
campo[(f1/i)] } ] [RECORD] { constante entera  
campo[(r,f1/i)]  
\*ISN [(r)] } [ campo[(f1/i)] ]

EL ESTATUTO GET ES USADO PARA LEER UN REGISTRO DANDO EL ISN DEL REGISTRO DESEADO.

GET SAME [(r)] [RECORD] [campo[(f1/i)]]...

ESTE ESTATUTO ES USADO PARA RELEER UN REGISTRO QUE ACTUALMENTE ESTA SIENDO PROCESADO CON EL PROPOSITO DE OBTENER UN MIEMBRO DE UN GRUPO PERIODICO O DE UN CAMPO MULTIPLE USANDO UN NOTACION PARA LA VARIABLE INDEXADA.

## NATURAL

GET TRANSACTION DATA campo (f1/i) ...40

ESTE ESTATUTO HABILITA SALVAR EL DATO CON ESTATUTO END TRANSACTION A SER LEIDO. NO GENERA NINGUN PROCESO LOOP. EL CONTENIDO DE LA VARIABLE DEL SISTEMA \*INIT-USER IDENTIFICA EL DATO DEL USUARIO A SER RECUPERADO

HISTOGRAM  $\left\{ \left( \begin{array}{l} (n) \\ \text{campo}[(r/i)] \end{array} \right) \right\}$  [VALUES][IN][FILE]  $\left\{ \begin{array}{l} \text{nom-archivo} \\ \text{num-archivo} \end{array} \right\}$  [PASSWORD =  
 $\left\{ \begin{array}{l} \text{'password'} \\ \text{campo}[(r/i)] \end{array} \right\}$ ] [VALUES][FOR][FIELD] descriptor [(i)] [STARTING]  $\left\{ \begin{array}{l} \text{WITH} \\ \text{FROM} \end{array} \right\}$   
 $\left\{ \begin{array}{l} \text{constante} \\ \text{campo}[(r, f1/i)] \end{array} \right\}$   $\left\{ \begin{array}{l} \text{THRU} \\ \text{ENDING AT} \end{array} \right\}$   $\left\{ \begin{array}{l} \text{constante} \\ \text{campo}[(r/i)] \end{array} \right\}$  WHERE descriptor [(i)] =  
criterio de condición lógica

ESTE ESTATUTO ES USADO PARA LEER LOS VALORES DEL CAMPO QUE ES DEFINIDO COMO DESCRIPTOR, SUPERDESCRIPTOR O SUBDESCRIPTOR EN LA BASE DE DATOS. EL PASSWORD ES ESPECIFICADO DE LA MISMA FORMA QUE EN ESTATUTO FIND. LOS VALORES SON LEIDOS DIRECTAMENTE DE LAS LISTAS INVERTIDAS DE ADABAS Y LOS VALORES SON REGRESADOS EN SECUENCIA ASCENDENTE. EL ESTATUTO HISTOGRAM NO PERMITE ACCESO PARA NINGUN CAMPO DEL ARCHIVO QUE NO SEA EL DESCRIPTOR.

IF condición Lógica [THEN]  $\left\{ \begin{array}{l} \text{IGNORE} \\ \text{estatuto} \\ \text{DO estatuto... [DOEND]} \end{array} \right\}$   $\left[ \begin{array}{l} \text{ELSE} \\ \text{estatuto} \\ \text{DO estatuto... DOEND} \end{array} \right]$

EL ESTATUTO IF ES USADO PARA CONTROLAR LA EJECUCION DE UN GRUPO DE ESTATUTOS O PARA UN CONTROL DIRECTO DE UNO O DOS GRUPOS DE ESTATUTOS. LA OPCION ELSE SE UTILIZA PARA EJECUTAR OTRO GRUPO DE ESTATUTOS QUE NO CUMPLAN CON LA CONDICION LOGICA. EL IGNORE SE USA CUANDO REQUERIMOS UN EVENTO DE FORMA INVERTIDA EN LA CONDICION LOGICA.

IF SELECTION  $\left[ \begin{array}{l} \text{NOT UNIQUE [IN FIELDS]} \\ \text{campo}[(r, f1/i)] \end{array} \right]$  ... [THEN]  
 $\left\{ \begin{array}{l} \text{estatuto} \\ \text{DO estatuto... DOEND} \end{array} \right\}$   $\left[ \begin{array}{l} \text{ELSE} \\ \text{estatuto} \\ \text{DO estatuto... DOEND} \end{array} \right]$

LA OPCION SELECTION CONTROLA LA EJECUCION DE UN ESTATUTO O GRUPO DE ESTATUTOS DO/DOEND BASADO EN LA PRESENCIA O ABSTINENCIA DE UN VALOR NULO EN UNO O MAS CAMPOS.

## NATURAL

```

INPUT { USING MAP { 'nom-map' } [NO ERASE] [{campo[(r,f1/i)}...]}
MAP { [NO ERASE] [(formato)] { { nT } { 'text' [(I)] } { *IN } campo[(r,f1/i)] }
  { { nX } { 'a' (n) [(I)] .. } { *OUT } }
  { { x/y } { '- ', '= ', / } { *OUTIN } } } }
[(formato)]
    
```

EL ESTATUTO INPUT ES USADO EN MODO INTERACTIVO PARA CREAR UN PANTALLA FORMATEADA O MAPA PARA LOS DATOS DE ENTRADA. ESTE ESTATUTO PUEDE SER USADO EN MODO BATCH, PERO LOS DATOS DEBEN SER PROPORCIONADOS POR EL USUARIO.

LA OPCION USING MAP PERMITE AL PROGRAMADOR CAMBIAR LA DEFINICION DEL MAPA SEPARADA DEL PROGRAMA EN NATURAL.

EL CONTENIDO DE LOS CAMPOS PUEDE SER ESPECIFICADO PARA ENTRADA, SALIDA Y PARA MODIFICACION DE LA SALIDA USANDO LAS PALABRAS RESERVADAS \*IN, \*OUT, \*OUTIN RESPECTIVAMENTE. EL DEFAULT ES \*IN.

ALGUNOS PARAMETROS ESTAN DISPONIBLES PARA CONTROLAR EL POSICIONAMIENTO Y CREACION DEL TEXTO:

nX	CAUSA n ESPACIOS A SER INSERTADOS ENTRE LOS ELEMENTOS.
nT	CAUSA TABULACION PARA IMPRESION EN LA POSICION n.
x/y	CAUSA QUE EL SIGUIENTE ELEMENTO ESTE POSESIONADO EN LA LINEA x Y EN LA COLUMNA y. LA COLUMNA 0 NO PUEDE SER UTILIZADA PARA POSICIONAMIENTO. EL POSESIONAMIENTO HACIA ATRAS NO ES PERMITIDO EN LA MISMA LINEA.
'text' (I)	CAUSA QUE EL 'text' ESTE PROTEJIDO CONTRA ESCRITURA Y (I) INTENSIFICA EL TEXTO EN LAS PANTALLAS 3270.
'a' (n) (I)	IDENTICO QUE 'text' EXCEPTO QUE EL CARACTER 'a' ES DESPLEGADO n VECES (n=1-132).
'-'	CUANDO ES PUESTO ADELANTE DE UN CAMPO, SUPRIME LA GENERACION DEL NOMBRE DEL CAMPO.
'='	CUANDO ES PUESTO ADELANTE DE UN CAMPO, DESPLIEGA EL ENCABEZADO DE DEFAULT DEL CAMPO.
/	CAUSA EL AVANCE DE UNA LINEA.



## NATURAL

### PARAMETROS QUE PUEDEN SER ESPECIFICADOS AL NIVEL DEL ESTATUTO:

AD	DEFINICION DE ATRIBUTO.
AL	LONGITUD ALFANUMERICA ( SOLO PARA ELEMENTOS DE SALIDA) 1-99.
EM	MASCARAS DE EDICION ( SOLO PARA ELEMENTOS DE SALIDA).
IP	PROMPT DE ENTRADA DE UN TEXTO.
LS	LONGITUD DE LA LINEA (1-250). DEFAULTL 132.
MC	CONTADOR DE UN CAMPO MULTIPLE (1-191). DEFAULT 1.
NL	LONGITUD NUMERICA (nn.m) (SOLO PARA ELEMENTOS DE SALIDA). nn+m <= 27, m < 8.
PC	CONTADOR DE GRUPO PERIODICO (1-99). DEFAULT 1.
PS	LONGITUD DE LA PAGINA (1-250). DEFAULT 60.
SG	POSICION DEL SIGNO. ON O OFF.
ZP	IMPRESION DEL CERO. ON O OFF.

### PARAMETROS DEL LOS ELEMENTOS DE ENTRADA:

AD	DEFINICION DE ATRIBUTO.
AL	LONGITUD ALFANUMERICA ( SOLO PARA ELEMENTOS DE SALIDA) 1-99.
EM	MASCARAS DE EDICION ( SOLO PARA ELEMENTOS DE SALIDA).
NL	LONGITUD NUMERICA (nn.m) (SOLO PARA ELEMENTOS DE SALIDA). nn+m <= 27, m < 8.
SG	POSICION DEL SIGNO. ON O OFF .
ZP	IMPRESION DEL CERO. ON O OFF .

### LIMIT n

ESTE ESTATUTO ES USADO PARA LIMITAR EL NUMERO DE EJECUCIONES DE UN PROCESO LOOP INICIALIZADO CON ESTATUTOS FIND, READ O HISTOGRAM.

### [CLOSE] LOOP [(r)]

EL ESTATUTO LOOP ES UTILIZADO PARA CERRAR UN PROCESO DE CICLO (LOOP), PUEDE SER ESPECIFICADO CON UN NUMERO DE REFERENCIA (r). TODOS LOS PROCESOS LOOPS DEBEN SER CERRADOS CON ESTE ESTATUTO.

MOVE [ROUNDED]  $\left\{ \begin{array}{l} [*]std-sf (campo[(r,f1/i)]) \\ constante \\ campo[(r,f1/i)] \end{array} \right\}$  TO {campo[(r,f1/i)]}...

EL MOVE ES USADO PARA MOVER EL VALOR O CONTENIDO DE UN OPERANDO A UNO O MAS CAMPOS. EL MOVE FUNCIONA EQUIVALENTE AL ESTATUTO ASSIGN.





## NATURAL

- LA PALABRA RESERVADA **LOGICAL** INDICA QUE LOS REGISTROS SERAN LEIDOS EN EL ORDEN QUE INDIQUEN LOS VALORES DEL DESCRIPTOR (EN LA SECUENCIA DEL DESCRIPTOR).

LA OPCION **STARTING FROM** Y **ENDING AT** PUEDEN SER UTILIZADOS PARA ESPECIFICAR EL LIMITE DE LOS REGISTROS QUE DESEAMOS BASADO EN UN RANGO DE VALORES ESPECIFICADOS POR EL USUARIO.

LA CLAUSULA **WHERE** ES USADA PARA ESPECIFICAR UN CRITERIO ADICIONAL EL CUAL ES EVALUADO DESPUES DE QUE EL REGISTRO ES LEIDO Y ANTES DE QUE EJECUTE CUALQUIER PROCESO ( INCLUYENDO EL PROCESO **BREAK** ).

LOS SISTEMA DE VARIABLES DIDSPONIBLES EN EL ESTATUTO **READ** SON EL **\*ISN** Y **\*COUNTER** LAS CUALES SON AUTOMATICAMENTE CREADAS PARA CADA **READ**. EL **\*ISN** CONTIENE EL **ISN** DEL REGISTRO QUE ESTA SIENDO PROCESADO Y EL **\*COUNTER** CONTIENE EL NUMERO DE VECES QUE EL **READ** ES EJECUTADO.

**READ WORK [FILE] n [ONCE] { RECORD campo [(f1)][FILLER nX] ...  
AND SELECT [OFFSET n] campo [(f1)] ...  
[FILLER nX] }**

EL **READ WORK FILE** ES USADO PARA LEER DATOS DE UN ARCHIVO DE TRABAJO SECUENCIAL FISICO NO-ADABAS. **n** INDICA EL NUMERO DEL ARCHIVO DE TRABAJO. ESTE ESTATUTO SOLO PUEDE SER EJECUTADO EN PROGRAMAS EN MODO **BATCH**.

**REDEFINE { campo[(r,f1/i)] ((nX  
campo[(f1)]))... } ...**

ESTE ESTATUTO ES UTILIZADO PARA REDEFINIR UN CAMPO NUMERICO, ALFANUMERICO O BINARIO. EL CAMPO A SER REDEFINIDO DEBE SER UN CAMPO DE LA BASE DE DATOS O VARIABLE DEFINIDA POR EL USUARIO. EL RESULTADO DE LA REDEFINICION PUEDE SER UN CAMPO SIMPLE O CONJUNTO DE CAMPOS LOS CUALES DEBEN SER VARIABLES DEFINIDAS POR EL USUARIO Y NO DEBEN ESTAR PREVIAMENTE DEFINIDAS.

**REINPUT [WITH][TEXT] 'text' MARK[FIELD] { n  
campo[(r,f1/i)] }**  
**[[AND][SOUND] ALARM]**

EL ESTATUTO **REINPUT** ES UTILIZADO PARA REGRESAR Y RE-EJECUTAR UN ESTATUTO **INPUT**.

## NATURAL

RELEASE {  
STACK  
SETS  
SET { 'set-nom' }  
VARIABLES  
}

ESTE ESTATUTO ES USADO PARA RESETEAR UNA PILA DE NATURAL, LIBERAR UN CONJUNTO DE REGISTROS PREVIAMENTE RETENIDOS POR UN COMANDO FIND O PARA RESETEAR CUALQUIER VARIABLE GLOBAL.

REPEAT [[UNTIL] condición lógica]

ESTE REPEAT ES USADO PARA INICIALIZAR Y EJECUTAR UN PROCESO DE CICLO (LOOP). EL LOOP CONTINUA INDEFINIDAMENTE HASTA QUE LA CONDICION LOGICA ESPECIFICADA ES SATISFECHA.

RESET { campo[(r, f1/i)] } ...  
nom-grupo

EL ESTATUTO RESET DEFINE EL VALOR DE UN CAMPO (O CAMPOS) CON VALORES NULOS (BLANCOS PARA LOS CAMPOS DE FORMATO A, CEROS PARA LOS DE FORMATO N, P O B).

RETRY

EL ESTATUTO RETRY SOLO PUEDE SER ESPECIFICADO CON UNA CONDICION DE ON ERROR.

RUN [REPEAT] { 'nom-programa' } { constante } { campo[(r/i)] } ...

EL ESTATUTO RUN ES USADO PARA LEER DINAMICAMENTE UN PROGRAMA FUENTE EN NATURAL DEL SISTEMA DE ARCHIVOS DE NATURAL Y EJECUTARLO.

SET GLOBALS parametro ...

EL SET GLOBALS ES USADO PARA DEFINIR LOS VALORES DE DEFAULT EN UNA SESION. LOS VALORES DE DEFAULT TIENE EFECTO PARA TODOS LOS PROGRAMAS EN UN TRABAJO BATCH O UNA SESION TP, A MENOS QUE SEA ATRAVES DE UN COMANDO GLOBALS O DE UN NUEVO ESTATUTO SET GLOBALS,

PARAMETROS QUE PUEDEN SER ESPECIFICADOS POR EL SET GLOBALS:

CC CODIGO DE CONDICION ( SOLO BATCH ). ON O OFF  
DEFAULT OFF.  
CF CONTROL DE FUNCION (TERMINAL) | CARACTER ESPECIAL.

## NATURAL

DC	SUBSTITUTO DE CARACTER DECIMAL. CUALQUIER CARACTER. DEFAULT . (PUNTO).
DU	OPCION DUMP. ON O OFF. DEFAULT OFF.
EJ	EJECT (CONTROL DE CARRO DE IMPRESORA). ON O OFF. DEFAULT ON.
IA	CARACTER DE ENTRADA. CARACTER ESPECIAL. DEFAULT -
ID	CARACTER DELIMITADOR DE ENTRADA. CARACTER ESPECIAL. DEFAULT ' .
IM	MODO DE SELECCION DE ENTRADA A LA TERMINAL. (F= FORMA /PANTALLA, D=DELIMITADOR/PALABRA RESERVADA) DEFAULT F.
LE	LIMITE ERROR. ON O OFF. DEFAULT OFF.
LS	LONGITUD DE LA LINEA (1-250). DEFAULT 132.
LT	LIMITE (1-99999999). DEFAULT 99999999.
MT	MAXIMO TIEMPO DE CPU (SEGS DE 1-9999). DEFAULT 60
NC	COMANDOS DE NATURAL. ON O OFF. DEFAULT ON.
PD	DATOS PAGINA (1-60). DEFAULT 50.
PS	LONGITUD DE LA PAGINA (1-250). DEFAULT 60.
SA	SONIDO DE ALARMA (FUNCION DE HARDWARE). ON O OFF DEFAULT OFF.
SF	FACTOR DE ESPACIAMIENTO (1-99). DEFAULT 1.
SL	LONGITUD DEL ESTATUTO FUENTE (1-74). DEFAULT 72.
TR	MODO TRANSACCION. ON O OFF. DEFAULT ON.
WH	ESPERA EN HOLD. ON O OFF. DEFAULT ON.
ZD	CHECADOR DE DIVISION DE CERO. ON O OFF. DEFAULT ON.
ZP	IMPRESION DEL CERO. ON O OFF. DEFAULT ON.

TODOS LOS PARAMETROS SON EVALUADOS AL TIEMPO DE EJECUCION SALVO : DC Y PS SON EVALUADOS AL TIEMPO DE COMPILACION Y EJECUCION, Y LS, SF, SL, TR Y ZP SON EVALUADOS AL TIEMPO DE COMPILACION.

SET KEY  $\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \\ \left\{ \begin{array}{l} \text{PAn} \\ \text{PFn} \end{array} \right\} \end{array} \right\} = \left\{ \begin{array}{l} \text{PGM} \\ \text{ON} \\ \text{OFF} \\ \text{valor} \end{array} \right\} \dots$

PA/PF SON LLAVES PA 1-3 Y LAS LLAVES PF 1-24

EL ESTATUTO SET KEY ES USADO PARA ASIGNAR FUNCIONES PARA LLAVES PA Y PF DURANTE LA EJECUCION DEL PROGRAMA. OFF DESACTIVA TODAS LAS LLAVES Y REGRESA EL CONTROL DE LAS LLAVES AL SISTEMA DE COMUNICACION DE TERMINALES. ON REACTIVA EL COMANDO O NOMBRE DE LA TRANSACCION ASIGNADO A LAS LLAVES. PGM HACE LA FUNCION SENSITIVA DE INTERROGACION EN EL PROGRAMA ACTUAL Y EL OFF DESACTIVA LA FUNCION Y REGRESA EL CONTROL AL SISTEMA DE COMUNICACION DE LA TERMINAL, ON REACTIVA EL COMANDO O NOMBRE DE LA TRANSACCION ASIGNADO A LAS LLAVES Y VALOR ASIGNA EL NOMBRE DE UN COMANDO O TRANSACCION A LA FUNCION DE LA LLAVE, EL VALOR PUEDE SER UNA CONSTANTE O EL CONTENIDO DE UNA VARIABLE ALFANUMERICA.

## NATURAL

{ SET TIME  
SETTIME }

ESTE ESTATUTO ES UTILIZADO PARA DEFINIR EL TIEMPO BASE DURANTE LA EJECUCION DE UN PROGRAMA.

SKIP [(x)] n [LINES]

ESTE ESTATUTO ES USADO PARA GENERAR UNA O MAS LINEAS EN BLANCO EN LA SALIDA DE UN REPORTE. EL SKIP SOLO PUEDE SER USADO EN CONJUNCION CON LOS ESTATUTOS WRITE, WRITE TITLE O DISPLAY.

SORT { RECORDS } [BY] { campo[(r,f1/i)] { ASCENDING  
ASC  
DESCENDING  
DESC } } ... 10 [ GIVE  
THEM ]

{ TOTAL  
MAX  
MIN  
NMIN  
AVER  
NAVER  
COUNT  
NCOUNT } ... [OF] ( campo [(r,f1/i)] ) ... ]

EL SORT ES USADO PARA INVOCAR UN SORT EXTERNO DE UN ARCHIVO NO-ADABAS O UN SORT INTERNO. UNO O MAS CAMPOS DE LA BASE DE DATOS (DESCRIPTORES Y NO-DESCRIPTORES) Y/O VARIABLES DEFINIDAS POR EL USUARIO PUEDEN SER USADOS EN ESTE ESTATUTO. UN CAMPO MULTIPLE O UN GRUPO PERIODICO (INDEXADOS UNICAMENTE) PUEDEN SER USADOS EN EL CRITERIO SORT, SOLO EL NOMBRE DE GRUPO NO ESTA PERMITIDO.

STACK [TOP] { [DATA] { constante  
campo[(r,f1/i)] } ... }  
{ COMMAND { constant  
campo[(r,f1/i)] } } }

EL STACK ES USADO PARA PONER DATOS EN LA PILA DE NATURAL DURANTE LA EJECUCION DE UN PROGRAMA. LOS DATOS PUEDEN SER USADOS COMO UN PROGRAMA O NOMBRES DE COMANDO PARA INVOCAR OTROS PROGRAMAS O COMANDOS O DATOS PARA UN ESTATUTO INPUT.

STOP

EL STOP ES UTILIZADO PARA TERMINAR LA EJECUCION DE UN PROGRAMA USUALMENTE BASADO EN UNA CONDICION LOGICA DE UN ESTATUTO IF.







## NATURAL

```

WRITE [(x)] [ { NOHDR
                NOTITLE
                NOTITLE NOHDR } ] [(formato)] { { [nX]
                                                    nT
                                                    x/y ] [ 'text' [(I)]
                                                         'a'(n) [(I)] ... ] }
{ campo[(r,f1/i)] [(formato)]
  [*]std-sf (campo[(r,f1/i)]) [(formato)]
  grupo [(r/i)] [(formato)] } ...

```

ESTE ESTATUTO WRITE ES USADO PARA PRODUCIR SALIDAS EN LIBRE FORMATO SIN COLUMNAS.

### PARAMETROS DEL ESTATUTO WRITE:

AD	DEFINICION DE ATRIBUTO. (I, N, B, D Y L, R). DEFAULT DR.
AL	LONGITUD ALFANUMERICA 1-99.
EM	MASCARAS DE EDICION. OFF UNICAMENTE.
ES	SUPRIMIR LINEA EN BLANCO. ON O OFF. DEFAULT OFF.
IS	SUPRIMIR IDENTICO. ON O OFF. DEFAULT OFF.
LS	LONGITUD DE LA LINEA (1-250). DEFAULTL 132.
MC	CONTADOR DE UN CAMPO MULTIPLE (1-191). DEFAULT 1.
MP	MAXIMO NUMERO DE PAGINAS (1 -99999). DEFAULT SIN LIMITE.
NL	LONGITUD NUMERICA (nn.m). nn+m <= 27, m < 8.
PC	CONTADOR DE GRUPO PERIODICO (1-99). DEFAULT 1.
PS	LONGITUD DE LA PAGINA (1-250). DEFAULT 60.
SG	POSICION DEL SIGNO. ON O OFF. DEFAULT ON.
ZP	IMPRESION DEL CERO. ON O OFF. DEFAULT OFF.

### NOTACION DEL FORMATO DE LOS REPORTES:

LA SIGUIENTE TABLA LISTA NOTACIONES ADICIONALES QUE PUEDEN SER ESPECIFICADAS EN EL ESTATUTO WRITE:

nX	CAUSA n ESPACIOS A SER INSERTADOS ENTRE LOS ELEMENTOS DE SALIDA O COLUMNAS.
nT	CAUSA TABULACION PARA IMPRESION EN LA POSICION n.
x/y	CAUSA QUE EL SIGUIENTE ELEMENTO ESTE POSESIONADO EN LA LINEA x Y EN LA COLUMNA y. LA COLUMNA O NO PUEDE SER UTILIZADA PARA POSICIONAMIENTO. EL POSESIONAMIENTO HACIA ATRAS NO ES PERMITIDO EN LA MISMA LINEA.
'text' (I)	CAUSA QUE EL 'text' SE IMPRIMA EN DETALLE EN LA LINEA Y (I) INTENSIFICA EL TEXTO A SER DESPLEGADO.

## NATURAL

'a' (n) (I) IDENTICO QUE 'text' EXCEPTO QUE EL CARÁCTER 'a' ES DESPLEGADO n VECES (n=1-132), (I) INTENSIFICA.

'=' CUANDO ES PUESTO ADELANTE DE UN CAMPO, DESPLIEGA EL ENCABEZADO DE DEFAULT DEL CAMPO.

/ CAUSA EL AVANCE DE UNA LINEA.

PARAMETROS DE LOS ELEMENTOS DEL ESTATUTO WRITE:

ESTOS PARAMETROS PUEDEN SER ESPECIFICADOS PARA CAMPOS INDIVIDUALES.

AD	DEFINICION DE ATRIBUTO. (I, N, B, D Y L, R). DEFAULT DR.
AL	LONGITUD ALFANUMERICA 1-99.
EM	MASCARAS DE EDICION ( VER ESTAUTO DISPLAY).
IS	SUPRIMIR IDENTICO. ON O OFF. DEFAULT OFF.
MC	CONTADOR DE UN CAMPO MULTIPLE (1-191). DEFAULT 1.
NL	LONGITUD NUMERICA (nn.m). nn+m <= 27, m < 8.
PC	CONTADOR DE GRUPO PERIODICO (1-99). DEFAULT 1.
SG	POSICION DEL SIGNO. ON O OFF. DEFAULT ON.
ZP	IMPRESION DEL CERO. ON O OFF. DEFAULT OFF.

WRITE [(x)] TITLE [ LEFT] PJUSTIFIED] ] [UNDELINED] [(formato)]

$$\left\{ \begin{array}{l} nX \\ nT \\ x/Y \end{array} \right\} \left\{ \begin{array}{l} \text{'text' [(I)]} \\ \text{'a'(n) [(I)]} \\ \text{'=', /} \end{array} \right\} \left\{ \begin{array}{l} \text{campo}[(r, f1/i)] [(formato)] \\ \text{std-sf (campo}[(r, f1/i))[(formato)] \end{array} \right\} \text{[SKIP n]...}$$

EL WRITE TITLE ES USADO PARA PONER EL TITULO DE DEFAULT EN LA PAGINA.

WRITE [(x)] TRAILER [ LEFT] PJUSTIFIED] ] [UNDELINED] [(formato)]

$$\left\{ \begin{array}{l} nX \\ nT \\ x/Y \end{array} \right\} \left\{ \begin{array}{l} \text{'text' [(I)]} \\ \text{'a'(n) [(I)]} \\ \text{'=', /} \end{array} \right\} \left\{ \begin{array}{l} \text{campo}[(r, f1/i)] [(formato)] \\ \text{std-sf (campo}[(r, f1/i))[(formato)] \end{array} \right\} \dots$$

EL WRITE TRAILER ES USADO PARA IMPRIMIR UN TEXTO AL FINAL DE CADA PAGINA.

WRITE WORK [FILE] n [VARIABLE]  $\left\{ \begin{array}{l} \text{campo} [(r, f1/i)] \\ \text{nombre-grupo} [(r/i)] \\ \text{constante} \end{array} \right\} \dots$

ESTE ESTATUTO ES USADO PARA GRABAR REGISTROS DE UN USUARIO A UN ARCHIVO DE TRABAJO FISICO SECUENCIAL. ESTE ESTATUTO PUEDE SER USADO EN MODO BATCH UNICAMENTE, n INDICA EL NUMERO DEL ARCHIVO DE TRABAJO (COMO LOS DEFINIDOS PARA NATURAL). ESTE ESTATUTO REQUIERE UN JCL ADICIONAL. PARA MAYOR REFERENCIA VER EL MANUAL DE OPERACION E INSTALACION DE NATURAL.

## NATURAL

### FORMATO Y LONGITUD PARA LAS VARIABLES DEFINIDAS POR EL USUARIO

#### FORMATO :

- N NUMERICO. LONGITUD MAXIMA DE 27 POSICIONES MAS SIGNO. LAS POSICIONES DE DEFAULT SON 7 MAS SIGNO.
- P EMPACADO. LONGITUD MAXIMA DE 27 POSICIONES MAS SIGNO. LAS POSICIONES DE DEFAULT SON 7 MAS SIGNO.
- B BINARIO NUMERICO. LONGITUD MAXIMA DE 4 POSICIONES LAS POSICIONES DE DEFAULT SON 10.
- B BINARIO NO - NUMERICO. LONGITUD MAXIMA > 128 POSICIONES. LAS POSICIONES DE DEFAULT SON 10.
- A ALFANUMERICO. LONGITUD MAXIMA DE 253 POSICIONES. LAS POSICIONES DE DEFAULT SON 10.

#### VARIABLES GLOBALES:

- + nombre SI UN + ES USADO COMO EL PRIMER CARACTER DE UN NOMBRE, A LA VARIABLE SE LE PUEDEN ASIGNAR VALORES DURANTE LA EJECUCION DE UN PROGRAMA.
- & nombre SI UN & ES USADO COMO EL PRIMER CARACTER DE UN NOMBRE, EL CONTENIDO DE LA VARIABLE SERA INTERPRETADO COMO UN TEXTO FUENTE CUANDO EL PROGRAMA SEA COMPILADO. EL CONTENIDO DE LA VARIABLE PUEDE SER PASADO DE UN PROGRAMA PREVIAMENTE EJECUTADO. LOS PROGRAMAS QUE UTILIZEN ESTE TIPO DE VARIABLE NO PUEDEN SER CATALOGADOS EN FORMA OBJETO.

LAS VARIABLES GLOBALES SIGUEN LAS MISMAS REGLAS DE FORMATO Y LONGITUD QUE LAS VARIABLES DEFINIDAS POR EL USUARIO.

## NATURAL

### REGLAS DE TRANSFERENCIA DE DATOS

AL TRANSFERIR DATOS DE UN CAMPO A OTRO SE NECESITAN SEGUIR LAS SIGUIENTES REGLAS:

CAMPO EMISOR	CAMPO RECEPTOR				
	ALFANUMERICO	NUMERICO	EMPACADO	B 1-4	B > 4
ALFANUMERICO	SI	NO	NO	SI	SI
NUMERICO	SI	SI	SI	SI	NO
EMPACADO	SI	SI	SI	SI	NO
BINARIO 1-4 Bytes	SI	SI	SI	SI	SI
BINARIO > 4 Bytes	SI	NO	NO	SI	SI

### MANEJO DE OVERFLOW ARITMETICO

EL MANEJO DE OVERFLOW ARITMETICO SE EXPLICA EN LA SIGUIENTE TABLA:

OPERACION	POSICIONES ANTES DEL PUNTO DECIMAL	POSICIONES DESPUES DEL PUNTO DECIMAL	ERROR
ADICION SUBSTRACCION	$A_{i+1}$ OR $B_{i+1}$ SEA MAYOR	$A_d$ OR $B_d$ SEA MAYOR	$C_i + C_d > 31$
MULTIPLICACION	$A_i + B_i + 2$	$A_d + B_d$ A UN MAXIMO DE 7	$C_i + C_d > 31$
DIVISION	$A_i + B_d$	$C_d$ OR $A_d$ SEA MAYOR PARA VALORES NO ENTEROS SUMAR 1 *	$C_i + C_d > 31$
EXPONENCIACION	$31 - A_d$	$A_d$	EXPONENTE $(B_i) > 15$ **
RAIZ CUADRADA SQRT	$A_i$	$A_d$	

## NATURAL

DONDE:

A: PRIMER OPERANDO  
B: SEGUNDO OPERANDO  
C: RESULTADO  
i: POSICIONES ENTERAS  
d: POSICIONES DECIMALES

\* EL RESULTADO UNICAMENTE ES USADO EN ESTATUTOS COMPUTE PARA LA EVALUACION DE LAS POSICIONES DECIMALES.

\*\* EL OVERFLOW ES DETECTADO DURANTE LA EJECUCION DE UN PROGRAMA DE NATURAL.

Ej.

$$N3.5 + N4.2 = N6.3$$

$$N3.5 * N4.2 = N11.5$$

$$N3.5 / N4.2 = N7.3$$

### VARIABLES DEL SISTEMA

- \*APPLIC-ID - LONGITUD A8, IDENTIFICA EL AMBIENTE DE APLICACION DEL USUARIO EN EL QUE ESTA TRABAJANDO ACTUALMENTE O EL IDENTIFICADOR DE LA LIBRERIA SI EL SISTEMA DE SEGURIDAD DE NATURAL NO ESTA INSTALADO.
- \*APPLIC-NAME - LONGITUD A32, IDENTIFICA EL NOMBRE DEL AMBIENTE DE APLICACION DEL USUARIO EN EL QUE ESTA TRABAJANDO ACTUALMENTE ( ESTE ES UNICAMENTE PARA EL SISTEMA DE SEGURIDAD DE NATURAL).
- \*COUNTER [(r)] - LONGITUD P8, CONTIENE EL NUMERO DE VECES QUE UN PROCESO LOOP ES INICIALIZADO POR UN ESTATUTO READ, FIND O HISTOGRAM . (r) ES USADO PARA INDICAR EL NUMERO DE LA LINEA DONDE SE ENCUENTRA EL ESTATUTO QUE LO REFIERE.
- \*CURSOR - LONGITUD N6, CONTIENE LA POSICION DEL CURSOR EN UNA PANTALLA AL TIEMPO QUE UN ENTER O UNA TECLA DE FUNCION ES PRESIONADA.
- \*DATA - LONGITUD N3, CONTIENE EL NUMERO DE ELEMENTOS QUE ESTAN DISPONIBLES EN LA PILA PARA SER PASADOS AL SIGUIENTE ESTATUTO INPUT COMO DATOS DE ENTRADA.
- \*DEVICE - LONGITUD A8, IDENTIFICA EL TIPO DE DISPOSITIVO DESDE EL CUAL NATURAL FUE INVOCADO.

## NATURAL

- \*ERROR-LINE - LONGITUD N4, CONTIENE EL NUMERO DE LA LINEA DEL ESTATUTO QUE CAUSA EL ERROR.
- \*ERROR - LONGITUD N7, CONTIENE EL NUMERO QUE IDENTIFICA EL ERROR DE NATURAL CAUSADO DE UNA CONDICION DE ON ERROR, VER EL MANUAL DE MENSAJES DE ERROR DE NATURAL
- \*INIT-ID - LONGITUD A8, IDENTIFICA LA TERMINAL ( DEFINIDA ACORDANDO LAS CONVENCIONES DEL SISTEMA TP DEL HOST ) DE LA CUAL NATURAL FUE LLAMADO.
- \*INIT-PROGRAM - LONGITUD A8, IDENTIFICA EL NOMBRE DEL PROGRAMA QUE ACTUALMENTE SE ESTA EJECUTANDO BAJO EL CONTROL DE NATURAL DEL SISTEMA HOST.
- \*INIT-USER - LONGITUD A8, IDENTIFICA EL USUARIO DE LA TERMINAL.
- \*ISN [(r)] - LONGITUD P8, CONTIENE EL NUMERO DE SECUENCIA INTERNO ( I S N ) DEL REGISTRO QUE ACTUALMENTE SE ESTA PROCESANDO EN UN PROCESO LOOP INICIALIZADO POR UN ESTATUTO READ O FIND.
- \*LIBRARY-ID - LONGITUD A8, CONTIENE EL IDENTIFICADOR DEL LA LIBRERIA ACTUAL EN NATURAL. EL IDENTIFICADOR ES ESPECIFICADO POR EL USUARIO EN EL COMANDO LOGON.
- \*LINE-COUNT[(x)] LONGITUD P5, CONTIENE EL NUMERO DE LINEA DE LA PAGINA. SI MULTIPLES REPORTES SON PRODUCIDOS POR UN PROGRAMA, (x) ES USADO PARA ESPECIFICAR EL NUMERO DEL REPORTE.
- \*LINESIZE - LONGITUD N7, IDENTIFICA EL TAMANO FISICO DE LA LINEA CUANDO NATURAL ES INVOCADO.
- \*NUMBER [(r)] - LONGITUD P8, CONTIENE EL NUMERO DE REGISTROS QUE FUERON SELECCIONADOS EN UN FIND (COMO RESULTADO DEL CRITERIO DE USAR LA CLAUSULA WITH) O DE UN HISTOGRAM
- \*PAGE-NUMBER[x] LONGITUD P5, CONTIENE EL VALOR ACTUAL DEL NUMERO DE PAGINA QUE SE ESTA IMPRIMIENDO.
- \*PAGESIZE - LONGITUD N8, IDENTIFICA EL TAMANO FISICO DE LA PAGINA CUANDO NATURAL ES INVOCADO.
- \*PF-KEY - LONGITUD A4, IDENTIFICA A LA LLAVE QUE FUE PRESIONADA PARA CAUSAR EL ULTIMO INPUT EN NATURAL.
- \*PROGRAM - LONGITUD A8, CONTIENE EL NOMBRE DEL PROGRAMA DE NATURAL QUE ACTUALMENTE SE ESTA EJECUTANDO.
- \*USER-NAME - LONGITUD A32, IDENTIFICA EL NOMBRE DEL USUARIO QUIEN ESTA ACTUALMENTE EN EL SISTEMA DE NATURAL. (UNICAMENTE DISPONIBLE CON EL SISTEMA DE SEGURIDAD DE NATURAL).

## NATURAL

### SISTEMA DE FUNCIONES ESTANDARD

EL SISTEMA DE FUNCIONES ESTANDARD PUEDE SER ESPECIFICADO EN ESTATUTOS MOVE, ASSIGN, DISPLAY O WRITE LOS CUALES ESTEN DESPUES DE UN AT END OF PAGE, AT BREAK O DE UN AT END.

- AVER (campo) - CONTIENE LA MEDIA DE LOS VALORES ENCONTRADOS POR EL CAMPO ESPECIFICADO.
- COUNT (campo) - LONGITUD FIJA DE N6, EL COUNT ES INCREMENTADO EN 1 CADA VEZ QUE PASA ATRAVES DEL LOOP AL CUAL ES REFERIDO.
- MAX (campo) - CONTIENE EL MAXIMO VALOR ENCONTRADO POR EL CAMPO ESPECIFICADO.
- MIN (campo) - CONTIENE EL MINIMO VALOR ENCONTRADO POR EL CAMPO ESPECIFICADO.
- NAVER (campo) - CONTIENE LA MEDIA DE LOS VALORES ENCONTRADOS POR EL CAMPO ESPECIFICADO, SOLO QUE EXCLUYE LOS VALORES NULOS.
- NCOUNT (campo) - LONGITUD FIJA DE N6, EL COUNT ES INCREMENTADO EN 1 CADA VEZ QUE PASA ATRAVES DEL LOOP AL CUAL ES REFERIDO, A MENOS QUE EL CAMPO ESPECIFICADO SEA NULO.
- NMIN (campo) - CONTIENE EL MINIMO VALOR ENCONTRADO POR EL CAMPO ESPECIFICADO, SOLO QUE EXCLUYE LOS VALORES NULOS.
- OLD (campo) - CONTIENE EL VALOR ANTERIOR DEL CAMPO CUANDO SE USA UNA CONDICION DE AT BREAK.
- SUM (campo) - CONTIENE LA SUMA DE TODOS LOS VALORES ENCONTRADOS POR EL CAMPO ESPECIFICADO.
- TOTAL (campo) - CONTIENE LA SUMA DE TODOS LOS VALORES ENCONTRADOS EN TODOS LOS LOOPS EN LOS CUALES ES LOCALIZADO.

## NATURAL

LAS SIGUIENTES FUNCIONES ESTANDARD PUEDEN SER ESPECIFICADAS EN ASSIGN, COMPUTE, DISPLAY, MOVE, WRITE O IF. ESTAS FUNCIONES PUEDEN SER UTILIZADAS EN CUALQUIER PARTE DEL PROGRAMA.

*DATD	- LONGITUD A8, FORMATO DE FECHA GERMANO DD.MM.YY.
*DATE	- LONGITUD A8, FORMATO DE FECHA EUROPEO DD/MM/AA.
*DATG	- LONGITUD A15, FORMATO DE FECHA GREGORIANA DDMONTHNAMEYYYY.
*DATI	- LONGITUD A8, FORMATO DE FECHA INTERNACIONAL YY-MM-DD
*DATJ	- LONGITUD A5, FORMATO DE FECHA JULIANA YYDDD.
*DATN	- LONGITUD N8, FORMATO DE FECHA EN FORMA YYYYMMDD.
*DATU	- LONGITUD A8, FORMATO DE FECHA U.S.A. MM/DD/YY.
SQRT (y)	- FUNCION DE RAIZ CUADRADA EN ESTATUTO COMPUTE. (y) REPRESENTA EL CAMPO.
*TIMD(r)	- LONGITUD N7, FORMATO HHMSST, TIEMPO TRANSCURRIDO DESDE QUE SE ESTABLECIO EL SETTIME.
*TIME	- LONGITUD A10, TIEMPO DEL DIA FORMATO HH:MM:SS.T.
*TIME-OUT	- LONGITUD N5, ESPECIFICA EL TIEMPO EN SEGUNDOS ANTES DE QUE LA TRANSACCION TERMINE.
*TIMN	- LONGITUD N7, TIEMPO DEL DIA FORMATO HHMSST.

### PALABRAS RESERVADAS DE NATURAL

.	DIVIDE	LESS EQUAL	SAVE
<	DO	LESS THAN	SCAN
<=	DOEND	LET	SCRATCH
+	DU	LIMIT	SELECT
*	EDIT	LINES	SELECTION
**	EDT	LIST	SEQUENCE
*IN	EJ	LOGICAL	SET
*OI	EJECT	LOGOFF	SETS
*OUT	ELSE	LOGON	SET TIME
*OUTIN	EM	LOOP	SETTIME
- =	END	LS	SF
-	ENDHOC	LT	SG
%	ENDING AT	MAP	SHOW
>	ENTER	MARK	SINGLE
>=	EQ	MASK	SKIP
	EQUAL TO	MAX	SL



## NATURAL

=	ERASE	MC	SORT
ABSOLUTE	ERROR	MD	SORTED
ACCEPT	ES	MIN	SOUND
AD	ESCAPE	MODULES	SOURCE
ADD	EVEN	MOVE	SPACE
ADHOC	EX	MP	SQRT
AL	EXAMINE	MT	STACK
ALARM	EXEC	MULTIPLY	START
ALL	EXECUTE	NAVER	STARTING
AND	FC	NC	STARTING FROM
AS	FETCH	NCOUNT	STATEMENT
ASC	FIELD	NE	STEP
ASCENDING	FIELDS	NEWPAGE	STOP
ASSIGN	FILE	NL	STORE
AT	FILES	NMIN	STOW
AT BREAK	FILL	NO	SUBROUTINE
AT END	FILLER	NOHDR	SUBTRACT
AT START	FIN	NOT	SUM
AT TOP	FIND	NOTEQUAL	SUPPRESS
AVER	FIRST	NOTIT	SUSPEND
BACKOUT	FOR	NOTITLE	SYSTEM
BATCH	FORMAT	NUMBER	TC
BEFORE	FOUND	OBTAIN	TERMINATE
BLOCKED	FROM	OF	TEXT
BREAK	FS	OFF	THAN
BROWSE	FUNCTIONS	OFFSET	THEM
BUT NOT	GC	OLD	THEN
BY	GE	ON	THRU
CALL	GET	ONCE	TIMD
CAP	GIVE	OR	TIME
CAPTIONED	GIVING	OR =	TIMN
CAT	GIVING NUMBER	PAn	TITLE
CATALOG	GLOBALS	PAGE	TO
CATLG	GREATER EQUAL	PAGE-NUMBER	TOP
CATMAP	GREATER THAN	PASSW	TOTAL
CC	GT	PASSWORD	TP
CF	HC	PC	TR
CHECK	HELP	PD	TRAILER
CIPH	HISTOGRAM	PERFORM	TRANSACTION
CIPHER	HORIZ	PFn	UC
CLEAR	HORIZONTALLY	PGM	UNCAT
CLOSE LOOP	HW	PHYSICAL	UNCATALOG
COMMAND	IA	PRIMARY	UNCATLG
COMPRESS	IC	PROCESSING	UNDERLINED
COMPUTE	ID	PROGRAMS	UNDLIN
COUNT	IDENTICAL	PS	UNIQUE
COUNTER	IF	PURGE	UNTIL
COUPLED	IGNORE	READ	UPDATE
CREATE	IM	RECORD	USING
DATA	IN	RECORDS	VALUE
DATD	INDEXED	REDEFINE	VALUES
DATE	INPUT	REINPUT	VARIABLE
DATG	INTO	REJECT	VARIABLES
DATI	IP	RELEASE	VERT

# NATURAL

DATJ  
DATN  
DATU  
DC  
DEFINE  
DELETE  
DELIMITERS  
DESC  
DESCENDING  
DISPLAY  
DISPLAY FORMATTED

IS  
ISN  
JUST  
JUSTIFIED  
KEY  
LC  
LE  
LEAVE  
LEAVING  
LEFT  
LESS

REPEAT  
REPLACE  
RESET  
RETAIN  
RETRY  
RETURN  
ROUNDED  
RUN  
RUNMODE  
SA  
SAME

VERTICALLY  
WH  
WHEN  
WHERE  
WHILE  
WITH  
WORK  
WRITE  
ZD  
ZP

## NATURAL

### MENSAJES DE ERROR

LOS MENSAJES DE ERROR DE NATURAL SON EXTENSIVOS EN SU EXPLICACION Y ESTAN DISPONIBLES EN MODO ON-LINE ATRAVES DEL HELP EN LA FUNCION DE 'ERROR MESSAGE INTERPRETATION', O TAMBIEN SE PUEDE OBTENER REFERENCIA DE ELLOS EN EL MANUAL DE CODIGOS Y MENSAJES DE ERROR.

Ej.

\*\*\*\*\* NATURAL HELP FUNCTION \*\*\*\*\*  
- ERROR MESSAGE DISPLAY -

1 NAT0001 MISSING/INVALID SYNTAX; UNDEFINDED VARIABLE-NAME/KEYWORD  
2 NAT0002 NO FILE AVAILABLE WITH ESPECIFIC NAME OR NUMBER  
3 NAT0003 INVALID CHARACTER STRING FOR FILE NAME OR FILE NUMBER  
4 NAT0005 CLOSING PARENTHESIS MISSING IN ARITH/LOGIC EXPRESSION  
.  
.  
.

ENTER POSITION NUMBER FOR DETALIED INFROMATION OR  
ENTER E TO STOP, M TO GO BACK TO MENU, >NNNN TO POSITION TO MSG : -

CADA MENSAJE DE ERROR CUENTA CON OTRA(S) PANTALLA(S), DONDE SE EXPLICA MAS A DETALLE EL SIGNIFICADO DE CADA ERROR.  
NATURAL TIENE APROXIMADAMENTE 8000 TIPOS DE ERROR.

## EJEMPLOS NATURAL

ESTE PROGRAMA SELECCIONA PERSONAL ENTRE 21 Y 25 AÑOS CON SUELDO MAYOR DE 1,500,000.00.

```

0010 WRITE TITLE 'PERSONAS QUE TIENEN ENTRE 21 Y 25 AÑOS'
0020 SKIP 1
0030 FIND PERSONAL WITH EDAD = 21 THRU 25 AND SALARIO GE 1500000
0040 SORTED BY EDAD APELLIDO NOMBRE /* SORTEA
0050 DISPLAY 'APELLIDO' APELLIDO
0060          1X 'NOMBRE' NOMBRE 1X '$' '=' SALARIO
0070          1X VERT AS 'DIRECCION' CALLE CIUDAD
0080          HORIZ / 'EDO.' ESTADO
0090 IF SEXO = 'M' THEN WRITE 40T 'MASCULINO'
0100 ELSE IF SEXO = 'F' THEN WRITE 40T 'FEMENINO'
0110          ELSE WRITE 40T '?'
0120 LOOP (0030)
0130 WRITE / 20T 'FIN DEL REPORTE'
0140 END
    
```

-----  
NEXT

### PERSONAS QUE TIENEN ENTRE 21 Y 25 AÑOS

APELLIDO	NOMBRE	SALARIO	DIRECCION	EDO.
BARRERA	ENRIQUE	\$ 3,000,000.00	RIO PANUCO MONTERREY	NL
		MASCULINO		
CAVAZOS	ALEJANDRO	\$ 2,500,000.00	DIAMANTE SAN NICOLAS	NL
		MASCULINO		
DIAZ	ELDA	\$ 2,500,000.00	J. ZOZAYA MONTERREY	NL
		FEMENINO		
HUERTA	ALBA	\$ 1,800,000.00	PARICUTIN MONTERREY	NL
		FEMENINO		
RODRIGUEZ	SARA	\$ 1,600,000.00	EUROPA APODACA	NL
		FEMENINO		
VAZQUEZ	ARMANDO	\$ 1,500,000.00	HACIENDA RIO BRAVO	TMS
		MASCULINO		
ZAMARRIPA	ENRIQUE	\$ 2,000,000.00	EVEREST MONTERREY	NL
		MASCULINO		
ZURIGA	JUAN	\$ 1,567,000.00	ANDES HIDALGO	NL

?

FIN DEL REPORTE

## EJEMPLOS NATURAL

ESTE PROGRAMA EFECTUA CORTES POR TRABAJO, EMPEZANDO POR UN DETERMINADO TIPO DE TRABAJO (EN EL CASO DEL EJEMPLO POR 'PROGRAMADOR')

```

0010 RESET #CONT (N7) #TOTAL (N12.2) #PROMEDIO (N8.2)
0020 WRITE TITLE 'CORTE DE SALARIO POR TIPO DE TRABAJO' SKIP 1
0030 READ PERSONAL POR TRABAJO STARTING FROM 'PROGRAMADOR'
0040 AT BREAK OF TRABAJO
0050     DISPLAY 'TRABAJO' OLD(TRABAJO)
0060     'NUMERO' COUNT (TRABAJO) (EM=ZZ,ZZ9)
0070     'SUMA/SALARIO' SUM(SALARIO) (EM=ZZ,ZZZ,ZZ9.99)
0080     'PROM/SALARIO' AVER(SALARIO) (EM=ZZ,ZZZ,ZZ9.99)
0090 * AL FINAL DE LOS DATOS EJECUTA ESTE ESTATUTO *
0100 AT END OF DATA DO
0110 NEWPAGE WITH TITLE 'TOTALES FINALES'
0120 MOVE COUNT(TRABAJO) TO #CONT
0130 MOVE TOTAL (SALARIO) TO #TOTAL
0140 COMPUTE ROUNDED #PROMEDIO = #TOTAL/#CONT
0150 WRITE / 10X 'TOTAL DE TRABAJOS' #CONT (EM=Z,ZZZ,ZZ9)
0160     / 10X 'TOTAL DE SALARIOS' #TOTAL (EM=ZZZ,ZZZ,ZZZ,ZZ9.99)
0170     / 10X 'SALARIO PROMEDIO' #PROMEDIO (EM=ZZ,ZZZ,ZZ9.99)
0180 DOEND
0190 LOOP
0200 END
    
```

-----  
 MORE

### CORTE DE SALARIO POR TIPO DE TRABAJO

TRABAJO	NUMERO	SUMA SALARIO	PROM SALARIO
-----	-----	-----	-----
PROGRAMADOR	15	54,000,000.00	3,600,000.00
RECEPCIONISTA	66	46,000,000.00	696,969.70
SECRETARIA	29	15,834,000.00	546,000.00
SUB-DIRECTOR	10	58,967,000.00	5,896,700.00

-----  
 NEXT

### TOTALES FINALES

TRABAJO	NUMERO	SUMA SALARIO	PROM SALARIO
-----	-----	-----	-----
TOTAL DE TRABAJOS	120		
TOTAL DE SALARIOS		174,801,000.00	
SALARIO PROMEDIO		1,454,675.57	

## EJEMPLOS NATURAL

ESTE PROGRAMA DESPLIGA EL MES CON LETRA DE LA FECHA DE REGISTRO DEL AUTOMOBIL. SE UTILIZA UNA TABLA DE MESES QUE SE REDEFINE, Y EL ARCHIVO SE LEE EN SECUENCIA FISICA.

```

0010 RESET #TAB-MES (A36) #FECH-REG (A12)
0020 REDEFINE #TAB-MES ( #MES-ELEM (A3) )
0030           #FECH-REG ( #DIA (N2) 1X #MES (A3) #CONST (A4)
0040           #YEAR (N2) )
0050 MOVE ', 19' TO #CONST
0060 MOVE 'ENEFEBMARABRMAYJUNJULAGOSEPOCTNOVDIC' TO #TAB-MES
0070 * LEE LOS AUTOMOBILES EN ORDEN FISICO *
0070 READ AUTOMOBILES IN PHYSICAL SEQUENCE
0080 * HACE UNA REDEFINICION DEL CAMPO FECHA DEL ARCHIVO *
0080 REDEFINE FECHA (#DD (N2) #MM (N2) AA(N2) )
0090 IF #MM < 1 OR #MM > 12 THEN MOVE ' ' TO #MES
0100 ELSE MOVE INDEXED #MES-ELEM <#MM> TO #MES
0110 MOVE #DD TO #DIA
0120 MOVE #AA TO #YEAR
0130 DISPLAY NUM-PROPIETARIO 5X KILOMETRAJE
0140           5X 'FECHA/REGISTRO' #FECH-REG
0150 LOOP
0160 END

```

-----  
MORE  
PAGE 1

YY-MM-DD HH:MM:SS

NUM-PROPIETARIO	KILOMETRAJE	FECHA REGISTRO
-----	-----	-----
1000006	50,000	18 ABR, 1976
1000009	39,678	01 NOV, 1985
1000034	12,897	15 MZO, 1988
1000056	42,890	07 ENE, 1989
1000067	13,000	12 MAY, 1976
1000123	49,666	20 FEB, 1989
1000240	11,111	05 JUL, 1989
1004567	58,789	23 JUN, 1989

## EJEMPLOS NATURAL

ESTE PROGRAMA DA MANTENIMIENTO AL ARCHIVO DE PERSONAL DE MANERA INTERACTIVA POR MEDIO DE UN MENU.

```
0010 RESET #SELECCION (A1) #NUM-PER (N8)
0020 INPUT 20T 'MENU DE MANTENIMIENTO DE PERSONAL'
0030      // 10T 'SELECCIONE UNA DE LAS OPCIONES:'
0040      /// 30T 'A - AÑADE UN REGISTRO'
0050      // 30T 'B - BORRA UN REGISTRO'
0060      // 30T 'C - CAMBIA UN REGISTRO'
0070      /// 30T 'F - FIN DEL PROCESO'
0080      /// 5T 'OPCION: ' *IN #SELECCION (AD='_' )
0090 IF #SELECCION = 'F' THEN DO
0100   WRITE 'PROCESO TERMINADO'
0110   STOP
0120 DOEND
0130 IF #SELECCION = 'A' OR = 'C' OR = 'B' THEN IGNORE
0140 ELSE REINPUT 'SELECCION INCORRECTA, INTENTE DE NUEVO'
0150       MARK 1 SOUND ALARM
0160 IF #SELECCION = 'C' THEN DO
0170   INPUT (SG=OFF)
0180       // 'DAR NUMERO DE PERSONAL: ' #NUM-PERS
0190   FIND NUMBER PERSONAL WITH NUMERO-PERSONAL = #NUM-PERS
0200   IF *NUMBER (0170) = 0 THEN
0210     REINPUT 'NUMERO DE PERSONAL INVALIDO, INTENTE DE NUEVO'
0220     ELSE PERFORM CAMBIOS
0230 DOEND
0240 IF #SELECCION = 'A' THEN PERFORM ALTAS
0250 IF #SELECCION = 'B' THEN PERFORM BAJAS
0260 DEFINE SUBROUTINE ALTAS
0270 *
0280 *
0290 * SE EFECTUA PROCESO DE ALTAS *
0300 *
0310 RETURN
0320 DEFINE SUBROUTINE BAJAS
0330 INPUT (SG=OFF)
0340       // 'DAR NUMERO DE PERSONAL: ' #NUM-PERS
0350   FIND FIRST PERSONAL WITH NUMERO-PERSONAL = #NUM-PERS
0360   IF *NUMBER (0170) = 0 THEN
0370     REINPUT 'NUMERO DE PERSONAL INVALIDO, INTENTE DE NUEVO'
0380   ELSE DO
0390     DELETE (0330) /* BORRA EL REGISTRO
0400     END TRANSACTION
0410   DOEND
0420 RETURN
0430 DEFINE CAMBIOS
0440 *
0450 *
0460 * EFECTUA EL PROCESO DE CAMBIOS
0470 *
0480 RETURN /* REGRESA A LA LINEA SIGUIENTE DE DONDE FUE LLAMADO
0490 FETCH 'PER001'
0500 END
```

## EJEMPLOS NATURAL

AL CORRER EL PROGRAMA EN LA PANTALLA APARECERA LO SIGUIENTE:

-----  
MENU DE MANTENIMIENTO DE PERSONAL

SELECCIONE UNA DE LAS OPCIONES:

- A - AñADE UN REGISTRO
- B - BORRA UN REGISTRO
- C - CAMBIA UN REGISTRO
- F - FIN DEL PROCESO

OPCION: \_

ESTA PANTALLA SE PUEDE CREAR POR MEDIO DE LA FACILIDAD DE MAPEO CON LA QUE CUENTA NATURAL. LA INSTRUCCIONES SERIAN ENTONCES:

```
0010 RESET #SELECCION (A1) #NUM-PER (N8)
0020 INPUT USING MAP 'MAP001' #SELECCION
0030 IF #SELECCION = 'F' THEN DO
0040   WRITE 'PROCESO TERMINADO'
0050   STOP
0060 DOEND
.
.
.
.
0480 END
```

LO CUAL AHORRA INSTRUCCIONES EN UN PROGRAMA Y CON EL EDITOR NO SERIA NECESARIO QUE EL PROGRAMADOR ESTABLECIERA LAS COORDENADAS DE LOS TEXTOS O CAMPOS A DESPLEGAR.



## EJEMPLOS NATURAL

ESTE PROGRAMA DA DE ALTA CAMPOS AL ARCHIVO DE AUTOMOBILES

```
0010 RESET #MARCA (A14) #MODELO (A20) #TIPO (A15) #COLOR (A10)
0020      #AÑO (N2)
0030 INPUT 20X 'AÑADIR UN REGISTRO AL ARCHIVO DE AUTOMOBILES'
0040 /// 5X 'MARCA: ' #MARCA 5X 'MODELO:' #MODELO 'TIPO:' #TIPO
0050      // 5X 'COLOR:' #COLOR 5X 'AÑO:' #AÑO
0060 STORE AUTOMOBILES WITH MARCA      = #MARCA
0070      MODELO      = #MODELO
0080      CARROCERIA  = #TIPO
0090      COLOR      = #COLOR
0100      AÑO        = #AÑO
0110 END TRANSACTION /* EFECTUA LA TRANSACCION
0120 WRITE NOTITLE / 'REGISTRO AÑADIDO CON ISN = ' *ISN (0060)
0130 END
```

---

<sup>o</sup>  
AÑADIR UN REGISTRO AL ARCHIVO DE AUTOMOBILES

MARCA: VOLKSWAGEN      MODELO: SEDAN      TIPO: 2 PUERTAS  
COLOR: ROJO              AÑO: 1989

---

NEXT

LIB=MCD

REGISTRO AÑADIDO CON ISN = 1003

EJEMPLOS NATURAL

ESTE PROGRAMA OBTIENE TODAS LAS TARJETAS DE CREDITO DE UNA PERSONA, UTILIZANDO EL ESTATUTO GET SAME JUNTO CON LA INSTRUCCION FOR.

```

# de p > 5
# kjs
0010 INPUT 10T 'NUMERO DE EMPLEADO: ' #NUM (N8)
0020 FIND FIRST FINANZAS WITH NUMERO-EMPLEADO = #NUM
0030 IF *NUMBER = 0 THEN 'REINPUT 'EMPLEADO NO EXISTE' MARK SOUND/
0040 DISPLAY NUMERO-EMPLEADO 5X TARJETA-CREDITO (1-5)
0050      5X LIMITE-CREDITO (1-5)
0060 IF C*CREDITOS > 5 THEN DO
0070   FOR #I = 6 TO C*CREDITOS
0080     GET SAME TARJETA-CREDITO (#I) LIMITE-CREDITO (#I)
0090     WRITE 14X TARJETA-CREDITO (0080/#I)
0100     5X LIMITE-CREDITO (0080/#I)
0110   LOOP
0120 DOEND
0130 END

```

-----  
 NUMERO DE EMPLEADO: 1050  
 -----

NUMERO EMPLEADO	TARJETA-CREDITO	LIMITE-CREDITO
1050	HERTZ	860,000.00
	AMERICAN EXPRESS	1,700,000.00
	VISA	540,000.00
	SEARS	780,000.00
	LIVERPOOL	990,000.00
	SORIANA	678,000.00

-----  
 NUMERO DE EMPLEADO: 10740  
 -----

NUMERO EMPLEADO	TARJETA-CREDITO	LIMITE-CREDITO
10740	BUDGET	2,450,000.00
	DINNERS CLUB	1,324,000.00

# EJEMPLOS NATURAL

ESTE PROGRAMA RESPALDA EL ARCHIVO DE PERSONAL EN UN ARCHIVO DE PASO (WORK FILE) PARA USO POSTERIOR.

```
0010 READ PERSONAL IN PHYSICAL SEQUENCE
0020 SORT BY EDAD NUMERO-PERSONAL
0030   AT START OF DATA RESET #CONTADOR (N7)
0040   ADD 1 TO #CONTADOR
0050   DISPLAY EDAD NUMERO-PERSONAL SEXO #CONTADOR
0060 * RESPALDO DE LOS DATOS SELECCIONADOS *
0070   WRITE WORK FILE 1 EDAD NUMERO-PERSONAL SEXO
0080 LOOP
0090 END
```

---

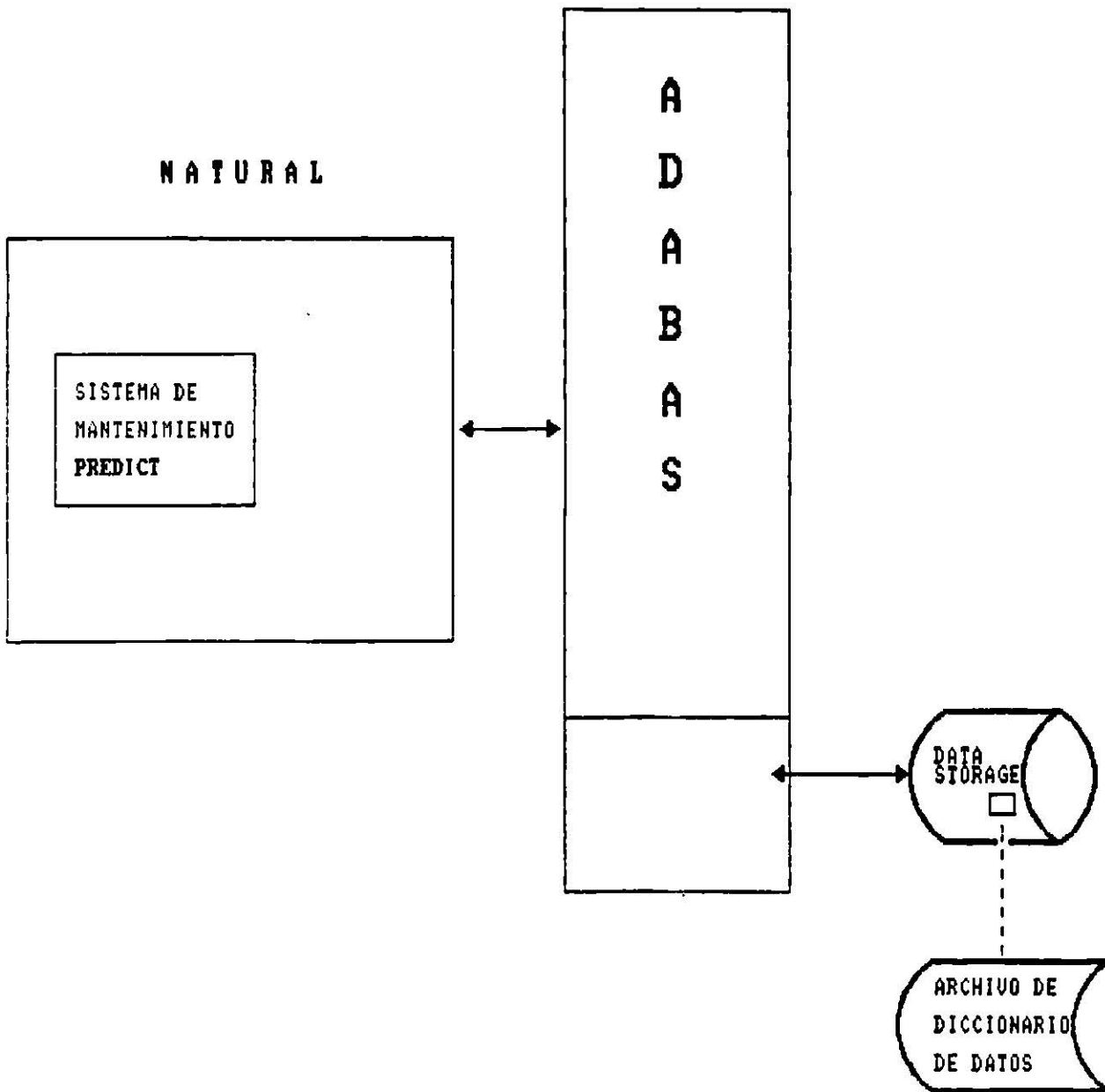
EDAD	NUMERO PERSONAL	SEXO	#CONTADOR
18	1000005	M	1
18	1210000	F	2
19	1000004	F	3
20	1234567	M	4
21	1000007	F	5
22	1000300	F	6
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

P R E D I C T

# PREDICT

## SISTEMA DE MANTENIMIENTO DEL DICCIONARIO DE DATOS

**PREDICT** ES EL SISTEMA DE MANTENIMIENTO DEL DICCIONARIO DE DATOS EN DONDE SE ENCUENTRA TODA LA INFORMACION DE LA ORGANIZACION DE LA BASE DE DATOS, EL CUAL SE CONSIDERA COMO UN ARCHIVO MAS DENTRO DE LA BASE DE DATOS, SE MANEJA POR MEDIO DE MENUS Y RESIDE DENTRO DEL SISTEMA DE **NATURAL**.



## PREDICT

**PREDICT** ES UN DICCIONARIO DE DATOS COMPLETO EN LINEA. ESTE DICCIONARIO FUE DISEÑADO ESPECIALMENTE PARA EL AMBIENTE **NATURAL**.

LA INTEGRACION DE **PREDICT** Y **NATURAL** PERMITE A **PREDICT** SER UN DICCIONARIO DE DATOS ACTIVO - UN DEPOSITO DE INFORMACION QUE USA Y LOCALIZA TODOS LOS DATOS INCORPORADOS Y LLAVES PARA QUE EL DESARROLLO DE LA APLICACION SEA MAS PRODUCTIVO -.

DETALLES DE LOS DATOS, MAPAS Y ARCHIVOS USADOS, SON CAPTURADOS AUTOMATICAMENTE POR **PREDICT**, CUANDO LOS PROGRAMAS DE **NATURAL** SON ALMACENADOS, POR MEDIO DE LA FACILIDAD DEL **CROSS REFERENCE**, ENTONCES, LA INFORMACION PUEDE SER ACCESADA INMEDIATAMENTE ATRAVES DE UN GRAN RANGO DE CONSULTAS ESTANDARD Y REPORTE, DE ESTA FORMA LAS ENTRADAS DE ERROR QUE LLEGAN A HACER LOS PROGRAMADORES QUEDAN ELIMINADAS.

LA INFORMACION ACERCA DE LOS PROGRAMAS DE LAS APLICACIONES ES SIEMPRE PRECISA Y REFLEJA GENUINAMENTE LA VERDAD DEL USO DEL SISTEMA, DICHO DE OTRO MODO, **PREDICT** ES CONFIABLE.

TODO EL MEDIO AMBIENTE DE **NATURAL** ES SEGURO Y CONTROLADO ATRAVES DEL USO DE UN DICCIONARIO DE DATOS **PREDICT**.

LAS VISTAS LOGICAS (USER VIEWS) USADAS EN EL DESARROLLO DE APLICACIONES DE **NATURAL** SON GENERADAS Y MANTENIDAS EN EL DICCIONARIO DE DATOS, DE ESTA FORMA LOS PROGRAMADORES NO NECESITAN CENTRARSE EN ESTRUCTURAS FISICAS DE LOS ARCHIVOS, SINO SOLO EN LAS VISTAS LOGICAS DE DATOS.

UNA VEZ QUE LAS VISTAS LOGICAS DE DATOS HAN SIDO DEFINIDAS EN **PREDICT**, ESTRUCTURAS FISICAS DE ARCHIVOS TALES COMO LAS USADAS EN **ADABAS**, O EN PRODUCTOS DE LA **IBM** COMO SON: **VSAM**, **DL1** E **IMS** PUEDEN SER MANEJADAS IDENTICAMENTE POR CUALQUIERA DE LAS APLICACIONES.

## PREDICT

LA SEGURIDAD DE **NATURAL** SE MANEJA EN UN COMPRENSIVO SISTEMA DE SEGURIDAD LLAMADO **NATURAL SECURITY** QUE TRABAJA EN CONJUNTO CON **PREDICT**.

DEBIDO A QUE EL NUMERO DE USUARIOS DEL SISTEMA DE **NATURAL** SE INCREMENTA, EL CONTROL ES UN PUNTO PRINCIPAL DE IMPORTANCIA, POR LO TANTO EL SISTEMA DA OPCION A CARACTERISTICAS DE SEGURIDAD A SER DEFINIDAS EN TRES NIVELES:

A) SEGURIDAD PARA USUARIOS, LA CUAL PUEDE INCLUIR:

- IDENTIFICACION Y PASSWORD POR USUARIO
- FACILIDADES DE **NATURAL** DISPONIBLES, INCLUYENDO TODOS LOS COMANDOS Y ESTATUTOS.

B) SEGURIDAD PARA APLICACIONES, LAS CUALES PUEDEN INCLUIR:

- UN PERIODO DE TIEMPO DETERMINADO DURANTE EL CUAL UNA APLICACION PUEDE SER EJECUTADA.
- DEFINIR LA APLICACION COMO PUBLICA O PRIVADA.
- CORREO DE MENSAJES PARA CUALQUIERA QUE USE LA APLICACION.
- APLICACIONES RESTRINGIDAS PARA ALGUNOS USUARIOS, TERMINALES O AMBAS.

C) SEGURIDAD PARA VISTAS LOGICAS DE DATOS

**NATURAL SECURITY** EN CONJUNTO CON **PREDICT** PROVEEN SEGURIDAD DE VISTA DE DATOS, PERMITIENDO QUE EN LA DEFINICION DE DATOS, ESTOS SEAN ALMACENADOS EN EL DICCIONARIO DE DATOS COMO PUBLICOS O PRIVADOS.

LOS DATOS PRIVADOS PUEDEN SER RESTRINGIDOS PARA UN SOLO USUARIO, O UN GRUPO DE USUARIOS, O PARA UNA APLICACION.

**PREDICT** TRABAJA JUNTO CON **NATURAL SECURITY** CON LO QUE RESPECTA A LA SEGURIDAD DE LAS VISTAS DE DATOS, ESTAS VISTAS PUEDEN ESTAR RESTRIGIDAS A USUARIOS O A APLICACIONES, Y SE DEFINEN EN **PREDICT**.

## PREDICT

### DEFINICION DE DATOS DEL ARCHIVO

LA SIGUIENTE TABLA LISTA LOS TERMINOS USADOS PARA DESCRIBIR ENTRADAS QUE APARECEN EN LOS ARCHIVOS:

---

FILE	EL NUMERO Y NOMBRE EXTERNO DEL ARCHIVO (FILE). LA MISMA LINEA PUEDE CONTENER EL DEFAULT SEQUENCE EL CUAL INDICA EL DESCRIPTOR A SER USADO COMO UN DEFAULT PARA UN ESTATUTO DE READ LOGICO.
TY	TIPO DE CAMPO:  G = GRUPO  P = GRUPO PERIODICO  M = CAMPO MULTIPLE  BLANCO = CAMPO ELEMENTAL
L	NUMERO DE NIVEL ASIGNADO A UN CAMPO ADABAS.
DB	NOMBRE DEL CAMPO ADABAS.
NAME	NOMBRE EXTERNO DEL CAMPO.
HD=XXX	XXX ES EL ENCABEZAMIENTO COLUMNAR POR DEFAULT QUE APARECE EN LOS CAMPOS CUANDO ESTOS SON IMPRESOS POR EL ESTATUTO DISPLAY.
EM=XXX	XXX ES LA MASCARA DE EDICION DE DEFAULT QUE APARECE EN LOS CAMPOS CUANDO ESTOS SON DESPLEGADOS POR EL ESTATUTO DISPLAY
*	INDICA QUE ES UNA LINEA DE COMENTARIOS.
F	INDICA EL FORMATO DE UN CAMPO ( N=NUMERICO, A=ALFANUMERICO, P=EMPACADO).
LENG	LA LONGITUD ESTANDARD DEL CAMPO.
S	CON UNA N INDICA SI EL CAMPO DEFINIDO ES NULO.
D	D INDICA SI UN CAMPO ES UN DESCRIPTOR ADABAS, CON UNA S INDICA QUE ES UN SUBDESCRIPTOR O SUPERDESCRIPTOR. P INDICA QUE EL CAMPO ES UN DESCRIPTOR FONETICO.
REMARKS	UN COMENTARIO QUE APLICA A CADA CAMPO.

EN LAS SIGUIENTES PAGINAS SE PRESENTAN ALGUNOS EJEMPLOS DE ESTRUCTURAS DE ARCHIVOS CREADAS EN PREDICT.



PREDICT

EJEMPLO DE ESTRUCTURA DE ARCHIVO CREADA EN EL DICCIONARIO DE DATOS:

FILE : PERSONAL

TYL	DB	NAME	F	LENG	S	D	REMARKS
1	AA	NUMERO-PERSONAL HD=NUMERO/PERSONAL	N	8.0		D	
G 1	G1	PERSONA					
2	BA	APELLIDO HD=PRIMER-APELLIDO	A	20	N	D	
2	BB	NOMBRE	A	15	N	D	
2	BC	INICIAL HD=I/N	A	1	F		
G 2	G2	P-DES HD=PERSONA-DESCRIPCIONES					
3	CA	SEXO HD=S/E/X/O	A	1	N	D	
3	CB	EDAD HD=E/D/A/D	N	2.0	N	D	
3	CC	ESTADO-FAMILIAR HD=EDO-FAM	A	10	N	D	
3	CD	NUMERO-DE-DEPENDIENTES HD=NUMERO/DEPEND	N	2.0	N	D	
G 1	G4	DIRECCION HD=DIRECCION					
2	DA	NUMERO	N	5.0	N		
2	DB	CALLE HD=NOMBRE-CALLE	A	20	N	D	
2	DC	CIUDAD HD=CD	A	15	N	D	
2	DD	ESTADO HD=EDO	A	2	N	D	
2	DE	CODIGO-POSTAL HD=C/P	N	5.0	N	D	
2	DF	TELEFONO HD=NUM-TEL	A	8	N	D	
G 1	G4	OCUPACION HD=OCUPACION					
2	FA	TRABAJO HD=TRABAJO ACTUAL	A	20	N	D	
2	FB	SALARIO HD=SALARIO	N	6.0	N	D	SALARIO MENSUAL
2	FC	COMISION HD=PORCENT/COMISION	N	6.0	N		
2	GA	AÑOS-EDUCACION HD=AÑOS/EDUC	N	2.0	N		

PREDICT

FILE : PERSONAL (CONTINUACION)

TYL	DB	NAME	F	LENG	S	D	REMARKS
---	---	-----	-	-----	-	-	-----
2	HA	AÑOS-COMPANIA HD=AÑOS/CIA	N	2.0	N		
2	IA	DIAS-VACACIONES HD=DIAS/VAC	N	2.0	N		
2	KA	DIAS-PERMISO HD=DIAS/PER	N	2.0	N		
1	LA	HOBBY-ACTIVIDAD HD=HOBBY	A	30	N	D	

# PREDICT

EJEMPLO DE ESTRUCTURA DE ARCHIVO CREADA EN EL DICCIONARIO DE DATOS:

FILE : AUTOMOBILES

TYL	DB	NAME	F	LENG	S	D	REMARKS
G 1	G1	CAR-DESCRIPCION					
2	AA	MARCA	A	14	N	D	
2	AB	MODELO	A	20	N	D	
2	AC	CARROCERIA	A	15	N	D	
2	BA	CILINDROS	N	2.0	N	D	
		HD=CIL					
2	BB	POTENCIA	N	3.0	N	D	
2	BC	PISTON	N	5.0	N		
G 1	G2	DETALLE					
2	BD	PESO	N	5.0	N		
2	CA	COLOR	A	10	N	D	
G 1	G3	REGISTRO					
2	DA	AÑO	N	2.0	N	D	
2	DB	NUM-SERIE	A	16	N		
2	FA	FECHA	N	6.0	N	D	
		EM=99/99/99					
2	FB	KILOMETRAJE	N	8.0	N	D	
		EM=99,999,999					
2	GA	NUM-PROPIETARIO	N	8.0		D	

PREDICT

EJEMPLO DE ESTRUCTURA DE ARCHIVO CREADA EN EL DICCIONARIO DE DATOS:

FILE : FINANZAS

TYL	DB	NAME	F	LENG	S	D	REMARKS
---	--	-----	-	-----	-	-	-----
1	AA	NUMERO-EMPLEADO HD=NUMERO/EMPLEADO	N	8.0		D	
P 1	MC	CREDITOS					
2	CC	TARJETA-CREDITO	A	18	N	D	
2	CL	LIMITE-CREDITO EM=ZZZ,ZZZ,ZZ9.99	N	9.2	N	D	
2	CB	BALANCE-ACTUAL HD=BALANCE EM=ZZZ,ZZZ,ZZ9.99	N	9.2	N	D	
M 1	OC	CREDITO-GAS	A	7		D	
1	NW	CIA	N	8.0	N	D	
P 1	IP	TIPOS-SEGURO					
M 2	IC	CIA-SEGURO	A	25	N	D	
M 2	PA	MONTO-POLIZA EM=ZZZ,ZZZ,ZZ9.99	N	9.2	N	D	
1	CG	COLEGIO	A	16	N	D	
P 1	VC	VACACIONES					
2	OB	EN-VAC	A	1	N	D	
1	IV	INVERSIONES	A	15	N	D	
1	SV	AHORROS	N	7.0	N	D	
1	BK	BANCO	A	20	N	D	

## PREDICT

### TIPO DE ARCHIVOS QUE SE MANEJAN EN PREDICT:

- \* ADABAS
- \* SISTEMA OPERATIVO
- \* CONCEPTUALES
- \* USERVIEWS
- \* ESTANDARD
- \* OTROS

ADABAS - SON LOS ARCHIVOS QUE FISICAMENTE ESTAN CONTENIDOS EN LA BASE DE DATOS.

DE SISTEMA OPERATIVO - (NO-ADABAS) SON LOS ARCHIVOS INDEXADOS, ACCESO DIRECTO, SECUENCIALES, ISAM O CONJUNTO DE DATOS PARTICIONADOS.

CONCEPTUALES - SON LA RELACION DE ALGUNOS CAMPOS DE PROPOSITO COMUN. NO EXISTEN REALMENTE EN LA BASE DE DATOS.

PREDICT NO EJECUTA NINGUNA VALIDACION PARA ESTE TIPO DE ARCHIVOS, NI GENERA NINGUN CODIGO. ESTE ARCHIVO PUEDE SER SUBSECUENTEMENTE COPIADO O ASIGNADO A OTROS TIPOS DE ARCHIVOS.

USERVIEWS ( VISTAS LOGICAS ) - UN USERVIEW ES UN SUBCONJUNTO DEL ARCHIVO CORRESPONDIENTE ADABAS, VSAM FISICO O VSAM LOGICO. SON ARCHIVOS CONCEPTUALES QUE SE OFRECEN AL USUARIO (PUEDEN SER TODO EL ARCHIVO FISICO O SOLO PARTE DEL ARCHIVO). NO HAY LIMITE DE CREACION DE USERVIEWS.

ESTANDARD - ES EL ARCHIVO QUE SE UTILIZA PARA EL DICCIONARIO DE DATOS, EN EL SE GUARDAN LOS ESTANDARES DE LOS CAMPOS Y ATRIBUTOS DE USO COMUN, DE ESTA MANERA SE ELIMINAN LAS NECESIDADES DE DEFINIR CADA CAMPO EN CADA ARCHIVO DE CADA APLICACION. SI SE CAMBIAN LOS ATRIBUTOS DE LOS CAMPOS EN EL ARCHIVO ESTANDARD, SERAN AUTOMATICAMENTE CAMBIADOS EN LOS ARCHIVOS FISICOS, LOGICOS Y USERVIEWS. LAS DESCRIPCIONES. ESTANDARD DE LOS CAMPOS PUEDEN SER COPIADAS PARA SER USADAS EN LAS DESCRIPCIONES DE ARCHIVOS ADABAS O NO-ADABAS.

OTROS - CUALESQUIER OTRO QUE NO ESTE MENCIONADO ANTERIORMENTE.

EN PREDICT SE PUEDEN DEFINIR VARIOS DICCIONARIOS DE DATOS EN UNA SOLA BASE DE DATOS, PERO SOLO UNO COMO ESTANDARD.

## PREDICT

### ATRIBUTOS DE LOS ARCHIVOS

NOMBRE DEL ARCHIVO - ES EL NOMBRE CON EL QUE SE IDENTIFICARA EL ARCHIVO.

NUMERO DE ARCHIVO - ES EL NUMERO QUE SE ASIGNA AL ARCHIVO.

ENCRIPCION DE DATOS Y SEGURIDAD DE ARCHIVO - SE DEFINE SI ESTA O NO CODIFICADO Y EL TIPO DE SEGURIDAD.

SECUENCIA DE LLAVE PRIMARIA - DEFINE LA LLAVE QUE SE VA A UTILIZAR PARA LA CARGA DE LA INFORMACION.

DUENO DEL ARCHIVO - SE DEFINE EL IDENTIFICADOR DEL USUARIO QUE PUEDE USAR, CODIFICAR, CAMBIAR EL ARCHIVO DEL CUAL ES DUENO.

DOCUMENTACION Y COMENTARIOS DEL ARCHIVO - CONTIENE LA INFORMACION DEL ARCHIVO.

PREDICT

EL SISTEMA DE MANTENIMIENTO DE PREDICT ESTA BASADO EN MENUS.

EL MENU PRINCIPAL SE SUBDIVIDE EN LOS SIGUIENTES SUBSISTEMAS:

\*\*\*\*\* P R E D I C T \*\*\*\*\*  
- MENU -

CODE	SUBSYSTEM
A	ACTIVE REFERENCES
D	DDA SERVICES
G	GENERATION
I	INCORPORATION
M	MAINTENANCE
R	RETRIEVAL
?	HELP
.	TERMINATE

PLEASE ENTER CODE: ?  
OR DIRECT COMMAND:

CADA OPCION ES UN SUBSISTEMA Y CADA UNA TIENE UNA DETERMINADA TAREA.

CADA SUBSISTEMA SE DIVIDE EN OBJETOS QUE SON ELEMENTOS QUE AYUDAN A DEFINIR UN SISTEMA ( MAPAS, PROGRAMAS, REPORTES, ARCHIVOS, RELACIONES, SISTEMAS, CAMPOS, BASES DE DATOS , etc. ).

SI SE ESCOGERIA LA OPCION 'A', LA PANTALLA SERIA LA SIGUIENTE:

\*\*\*\*\* P R E D I C T \*\*\*\*\*  
- ACTIVE REFERENCE MENU -

CODE	OBJECTS
E	ELEMENTARY FIELD
F	FILE
M	MODULE
P	PROGRAM
R	REPORT/MAP
S	SYSTEM/APPLICATION
?	HELP
.	RETURN TO MENU

PLEASE ENTER CODE: ?  
OR DIRECT COMMAND:

# PREDICT

ESTE SUBSISTEMA ES EL QUE NOS DESPLEGARA LA LISTA YA SEA DE CAMPOS ELEMENTALES O DE MODULOS O DE PROGRAMAS, Etc. EN UNA GRAN CANTIDAD DE RELACIONES YA QUE CADA OBJETO CUENTA A LA VEZ CON UN SUBMENU. POR EJEMPLO SI SE ESCOGERA LA OPCION 'F', LA SIGUIENTE PANTALLA SERIA:

\*\*\*\*\* P R E D I C T \*\*\*\*\*  
- ACTIVE REFERENCE FILE -

CODE	FUNCTION	FILE-ID	TYPE	DB-ID
A	LIST REAL FILES USED BY PROGRAMS	O	O	O
C	COUNT FILES USED IN LIBRARY	O	O	O
D	LIST FILES WHICH ARE USED BUT NOT DEFINED	O		
N	LIST FILES WHICH ARE DEFINIDED BUT NOT USED	*	O	O
R	LIST FILES USED BY PROGRAMS	O	O	O
S	SELECT FILE FROM A LIST	*	O	*
?	HELP			
.	RETURN TO ACTIVE REFERENCE MENU			

PARAMETERS: O = OPTIONAL  
\* = '\*' NOTATION VAL

PLEASE ENTER CODE: ?  
FILE-ID:  
FILES OF TYPE:  
IN DB-ID:  
OWNER-ID:  
WITH KEYWORD:  
OR DIRECT COMMAND:

CADA OBJETO TIENE DIFERENTES FUNCIONES COMO MODIFICAR, ANADIR, DESPLEGAR, COPIAR, BORRAR, EDITAR, LISTAR, etc.

PREDICT NOS DA INFORMACION MUY UTIL CON RESPECTO A LOS SISTEMAS, EN EL SE PUEDE OBTENER DE UNA MANERA FACIL Y RAPIDA CUALQUIER INFORMACION CON RESPECTO A PROGRAMAS, REPORTE, LA RELACION DE PROGRAMAS Y REPORTE, MAPAS Y SU RELACION CON LOS PROGRAMAS, REPORTE VS ARCHIVOS Y VICEVERSA, ARCHIVOS VS PROGRAMAS Y VICEVERSA, etc.

ASI, CADA SUBSISTEMA TIENE PANTALLAS CON OBJETOS QUE A SU VEZ TIENEN DIFERENTES FUNCIONES QUE NOS FACILITAN EL MANEJO DE INFORMACION. A CONTINUACION SE EXPLICAN BREVEMENTE CADA UNO DE LOS SUBSISTEMAS:



## PREDICT

EL 'ACTIVE REFERENCES' ES EL SUBSISTEMA QUE PERMITE REALIZAR CONSULTAS DE NUESTRAS APLICACIONES, COMO:

- LISTAR CAMPOS, ARCHIVOS, MODULOS Y REPORTES UTILIZADOS EN PROGRAMAS.
- CONTAR CAMPOS, ARCHIVOS Y MODULOS UTILIZADOS EN LAS LIBRERIAS.
- LISTAR LOS PROGRAMAS, SISTEMAS Y REPORTES QUE SON USADOS PERO NO DEFINIDOS.
- LISTAR LOS ESTADOS DE DOCUMENTACION DE LOS PROGRAMAS, REPORTES Y SISTEMAS IMPLEMENTADOS.
- LISTAR LOS CAMPOS O ARCHIVOS QUE NO HA SIDO DEFINIDOS.
- LISTAR LOS CAMPOS, ARCHIVOS, MODULOS, PROGRAMAS, REPORTES Y SISTEMAS QUE HAN SIDO DEFINIDOS PERO NO UTILIZADOS.
- SELECCIONAR CAMPOS, ARCHIVOS, MODULOS, PROGRAMAS, REPORTES Y SISTEMAS DE UNA LISTA.

Y OTRAS OPCIONES.

EL 'DDA SERVICES' (DATA DICTIONARY ADMINISTRATOR) ES EL SUBSISTEMA QUE MANTIENE VARIAS CLASES DE GENERACION DE CODIGOS, DEFINE VARIOS DEFAULTS PARA EL SISTEMA **PREDICT**, ESPECIFICA ENCABEZADOS PARA LAS DESCRIPCIONES EN EXTENSION, RECUPERA ARCHIVOS EN SITUACIONES DONDE HAY PROBLEMAS CON EL HARDWARE O SOFTWARE Y BORRA REGISTROS DEL ACTIVE CROSS REFERENCE. EN ESTE SUBSISTEMA SE MODIFICAN LAS TARJETAS DE DEFAULT ADAWAN/ADACMP, LAS BAL (ASSEMBLER), COBOL, DDM, PL/1, Etc.

EL 'GENERATION' PERMITE AL USUARIO TENER EL CONTROL DE LA GENERACION DE SALIDAS, BASADO EN LA INFORMACION DEL DICCIONARIO DE DATOS, LA SIGUIENTES FUNCIONES PUEDEN SER EJECUTADAS USANDO ARCHIVOS Y CAMPOS ACTUALMENTE DEFINIDOS EN EL DICCIONARIO: GENERACION DE DEFINICIONES PARA LA UTILERIA DE COMPRESION DE **ADABAS** ( ADAWAN / ADACMP ), GENERACION DE ENSAMBLADOR, DE COBOL, DE DDM, DE PL/I, DE DEFINICIONES DE UTILERIAS DE SEGURIDAD (ADASCR), Etc.

EL 'INCORPORATION' ES USADO PARA DESCRIBIR LA ACTUALIZACION DEL DICCIONARIO DE DATOS, COPIANDO LA INFORMACION DE OTRO SISTEMAS DE SOFTWARE. ESTE SUBSISTEMA INCLUYE FUNCIONES PARA COMPARAR LA

## PREDICT

INFORMACION QUE RESIDE EN EL OTRO SOFTWARE CON LA INFORMACION ALMACENADA EN EL DICCIONARIO DE DATOS, SI SE ENCUENTRAN DISCREPANCIAS, LOS DATOS DEL DICCIONARIO PUEDEN SER OPCIONALMENTE CAMBIADOS. ESTE SUBSISTEMA ES UTILIZADO PARA:

- CREAR DATOS PARA EL DICCIONARIO USANDO PROGRAMAS FUENTES DE **NATURAL**.
- CREAR DATOS PARA EL DICCIONARIO USANDO DEFINICIONES DE DATOS DE COBOL.
- CREAR DATOS PARA EL DICCIONARIO USANDO DATOS DE LA DEFINICION DE MODULOS (DDM), DE UN ARCHIVO O USERVIEW, O COMPARACION DE ENTRADAS DE DATOS CON EL DDM.
- CREAR DATOS PARA EL DICCIONARIO USANDO FDT (TABLA DE DEFINICION DE CAMPOS - FIELD DEFINITION TABLE-) O UN ARCHIVO **ADABAS** O COMPARACION DE ENTRADAS DE DATOS CON EL FDT.
- INCORPORAR LA DESCRIPCION DE **NATURAL SECURITY** EN EL DICCIONARIO DE DATOS.

Y OTRAS OPCIONES.

EL 'MAINTENANCE' ES EL SUBSISTEMA QUE PERMITE AÑADIR, COPIAR, DESPLEGAR, MODIFICAR, RENOMBRAR, BORRAR, Etc. LOS CAMPOS Y SU INFORMACION DEL DICCIONARIO DE DATOS. ESTE SUBSISTEMA CUENTA CON UN EDITOR PARA CREAR LAS ESTRUCTURAS DE ARCHIVOS, PARA DAR DE ALTA CAMPOS EN EL ARCHIVO ESTANDARD, DAR DE ALTA PROGRAMAS, MODULOS, PALABRAS RESERVADAS, REPORTES, RELACIONES, SISTEMAS, USUARIOS, Etc. EN ESTE SUBSISTEMA SE INDICAN LOS TIPOS DE ARCHIVOS QUE SE VAN A UTILIZAR (ADABAS, CONCEPTUAL, ACCESO DIRECTO, ISAM, USERVIEWS, Etc.).

EL 'RETRIEVAL' TIENE UN GRAN CONJUNTO DE FUNCIONES DISPONIBLES QUE PERMITEN SELECCIONAR LA INFORMACION DEL DICCIONARIO DE DATOS EN DETALLE, PUEDE EXPLORAR ARCHIVOS, CAMPOS, PROGRAMAS, MODULOS, REPORTES, Etc. ESTE SUBSISTEMA DE CONSULTA NOS DA INFORMACION MUCHO MAS COMPLETA QUE EL 'ACTIVE REFERENCES'.

TODOS LOS MENU CUENTAN CON UNA AYUDA (HELP) EN EL CUAL SE EXPLICAN CADA UNA DE LAS OPCIONES A LAS CUALES SE ESTAN HACIENDO REFERENCIA.

## PREDICT

DENTRO DE **NATURAL** SE PUEDE OBTENER INFORMACION DE RELACIONES DE PROGRAMAS Y ARCHIVOS Y DE OTRAS RELACIONES SENCILLAS QUE SON DE FACIL ACCESO POR MEDIO DEL LIST XREF ( **CROSS REFERENCE** ) , SIN NECESIDAD DE SALIR DE **NATURAL** PARA ENTRAR A **PREDICT**.

LA FORMA DE INVOCAR ESTA FACILIDAD, ES TECLEAR 'L X' EN EL PROMPT 'NEXT' DE **NATURAL**.

LA SIGUIENTE PANTALLA APARECERA:

```
***** P R E D I C T *****  
- CROSS REFERENCE -
```

CODE	FUNCTION
E	CALLED EXTERNAL SUBROUTINES
G	GLOBAL VARIABLES
M	MAPS
N	INVOKED NATURAL PROGRAMS
R	XREF DATA REPORT
V	VIEWS AND FIELDS
?	HELP
.	TERMINATE

CODE: ?

OR ENTER COMMAND:

Y SE MANEJARA DE LA MISMA FORMA MENCIONADA ANTERIORMENTE.  
ES IMPORTANTE RECALCAR QUE LA INFORMACION QUE SE DESPLEGARA ESTA CONTENIDA EN EL DICCIONARIO DE DATOS, YA QUE SE ALMACENA AUTOMATICAMENTE EN **PREDICT**.

A D A B A S

## ADABAS

**ADABAS** ( ADAPTABLE DATA BASE SYSTEM ) ES UN MANEJADOR DE BASE DE DATOS RELACIONAL CUYA PRIMER VERSION PRODUCTIVA TUVO LUGAR EN 1971.

**ADABAS** ES UN SISTEMA DE MANEJO DE BASE DE DATOS RELACIONAL, ES MUY FLEXIBLE Y TIENE UNA GRAN ADAPTABILIDAD, FACIL USO Y CUENTA CON UN EFICIENTE Y EFECTIVO CONTROL DE SEGURIDAD.

**ADABAS** CUENTA CON LAS SIGUIENTES CARACTERISTICAS:

- MAXIMA INTEGRIDAD DE DATOS
- SELECCION COMPLEJA DE DATOS
- FACILIDADES DE RECUPERACION Y REINICIO
- SEGURIDAD DE DATOS
- INTERFASE INTEGRADA AL DICCIONARIO DE DATOS
- FLEXIBLE PARA LAS NECESIDADES DE CAMBIO
- ACCESO ON-LINE Y BATCH
- CENTRALIZA, CONTROLA Y CONSOLIDA INFORMACION
- COMPARTE INFORMACION MULTIUSUARIO
- PROTECCION DE DATOS
- CONSERVA LA INTEGRIDAD DE LA INFORMACION
- INDEPENDENCIA DE DATOS
- EFICIENTE Y DE FACIL MANEJO DE BASE DE DATOS

LA ARQUITECTURA DE **ADABAS** HACE POSIBLE SELECCIONAR UN CONJUNTO DE REGISTROS DE LA BASE DE DATOS USANDO UNA VARIEDAD DE CRITERIOS DE BUSQUEDA DE SIMPLES A COMPLEJOS.

LOS REGISTROS DE DATOS PUEDEN SER SELECCIONADOS A TRAVES DE LA ESTRUCTURA INDEXADA DE **ADABAS** SIN EL ACCESO POR EL DATO EN SI, ES UNA TECNICA QUE GARANTIZA UN TIEMPO DE RESPUESTA RAPIDO AUN CUANDO SEA EL CRITERIO DE BUSQUEDA MAS COMPLEJO.

CON **ADABAS** LAS TRANSACCIONES INCOMPLETAS CAUSADAS POR FALLAS DEL SISTEMA SON ABANDONADAS AUTOMATICAMENTE ASEGURANDO LA INTEGRIDAD DE

## ADABAS

LOS DATOS TODO EL TIEMPO. ADEMAS **ADABAS** PROVEE LA SEGURIDAD ESENCIAL DE LOS DATOS EN UN AMBIENTE MULTIUSUARIO.

LAS MEDIDAS DE SEGURIDAD SOPORTADAS SON A NIVEL DE ARCHIVO, DE CAMPO Y DE VALOR DE CAMPO, CON UN FORMATO DE ENCRIPCION OPCIONAL.

CONSTA DE TRES GRUPOS PRINCIPALES DE DATOS:

- ASOCIADOR (ASSO)
- AREA DE DATOS (DATA STORAGE)
- AREA DE TRABAJO (WORK AREA)

EL ASOCIADOR ES EL QUE CREA LAS LISTAS INVERTIDAS DE LOS DESCRIPTORES DE LOS ARCHIVOS Y TAMBIEN LOS CONVERTIDORES DE DIRECCIONES.

EL AREA DE DATOS ES EL LUGAR DONDE SE ENCUENTRA LA INFORMACION DE LOS ARCHIVOS.

EL AREA DE TRABAJO ES LA AREA UTILIZADA POR ADABAS PARA LLEVAR ACABO DIVERSAS OPERACIONES (COMO SORTEAR, BUSCAR, Etc.).

DOS PARTES PRINCIPALES DE **ADABAS** SON:

- EL NUCLEO
- BUFFER INPUT/OUTPUT

EL NUCLEO ES DONDE ESTA REALMENTE LA BASE DE DATOS Y SE ENCARGA DE ANALIZAR LA SINTAXIS Y CONGRUENCIA DE LAS INSTRUCCIONES.

EL BUFFER DE I/O ES EL QUE CONTROLA EL MANEJO DE LAS ENTRADAS Y SALIDAS DE LA INFORMACION.

**ADABAS** PARA CONTROLAR LA INFORMACION UTILIZA DOS TIPOS DE PASSWORDS DE ACCESO:

- ACESSO
- ACTUALIZACION

**ADABAS** SE DIVIDE EN TRES NIVELES DE PASSWORDS DE SEGURIDAD:

## ADABAS

- NIVEL DE ARCHIVO
- NIVEL DE CAMPO
- NIVEL DE VALOR

UN METODO ADICIONAL PARA LA SEGURIDAD DE **ADABAS** ES EL CIFRAMIENTO (CIPHERING - CODIGOS DE CONVERSION -), QUE CONSISTE DE QUE EL USUARIO UTILICE UN PASSWORD EXCLUSIVO QUE **ADABAS** NO REGISTRA, ES DECIR SI EL USUARIO LLEGARA A OLVIDARLO NO HABRIA FORMA DE ACCESAR LA BASE DE DATOS.

MANEJA AREA DE PADDING QUE ABSORBE EL CRECIMIENTO DE LOS CAMPOS. EN ESTA AREA SE PUEDEN AGREGAR O ACTUALIZAR REGISTROS ( DEFINIR CAMPOS DE MAYOR LONGITUD). ESTE PARAMETRO DEPENDERA DE LAS NECESIDADES DE EXPANSION QUE SE REQUIERAN, GENERALMENTE SE UTILIZA UN 5% DE PADDING.

### ADABAS MANEJA UN CONJUNTO DE UTILERIAS:

HAY UTILERIAS PARA COMPRESION Y CARGA, PARA MODIFICACION DE ARCHIVOS, AUDITORIA Y SEGURIDAD, ASI COMO PARA REPORTE DE STATUS DE LA BASE DE DATOS PARA LOS ADMINISTRADORES DE LA MISMA, TAMBIEN CUENTA CON UTILERIAS PARA LA MODIFICACION DE LA BASE DE DATOS, REORDENAMIENTO DEL ASOCIADOR, REINICIO, DESCOMPRESION, RESTAURACION, RECUPERACION Y ACTUALIZACION DE ARCHIVOS. A CONTINUACION SE MENCIONAN ALGUNAS DE ELLAS:

ADAMER - PERMITE CALCULAR EL ESPACIO OPTIMO. (HAY QUE BALANCEAR EL ESPACIO DISPONIBLE EN DISCO Y EL TIEMPO DE I/O).

ADAWAN - SE ENCARGA DE COMPRIMIR REGISTROS Y CONSTRUIR ARCHIVOS FISICOS INTERNAMENTE PARA LA BASE DE DATOS.

ADALD1 - CARGA REGISTROS COMPRIMIDOS DENTRO DEL AREA DE DATOS , ASIGNA ISN'S A CADA UNO DE LOS REGISTROS, CONSTRUYE EL ADDRESS CONVERTER PARA CADA ARCHIVO, CARGA LOS ARCHIVOS FISICOS Y EXTRAE DE CADA REGISTRO LOS VALORES DE LOS DESCRIPTORES.

ADALD2 - CONSTRUYE Y CLASIFICA LA LISTA INVERTIDA PARA CADA DESCRIPTOR.

ADAHD1/ADAHD2 - SE UTILIZAN PARA LA MODIFICACION DE ARCHIVOS, CAMBIAN LA LONGITUD ESTANDARD DE UN CAMPO, CONVIERTEN UN CAMPO YA

## ADABAS

EXISTENTE A DESCRIPTOR Y VICEVERSA, AGREGAN NUEVOS CAMPOS AL ARCHIVO. NO EXISTEN UTILERIAS PARA BORRAR CAMPOS DE LOS ARCHIVOS, EN CASO DE REQUERIRLO SE TENDRIA QUE BORRAR EL ARCHIVO Y VOLVERLO A CREAR CON LOS NUEVOS CAMPOS.

ADAVUS - BORRA ARCHIVOS COMPLETOS DE LA BASE DE DATOS, INCREMENTA EL ESPACIO FISICO ASIGNADO A LA BASE DE DATOS, CAMBIA EL NUMERO DE ARCHIVO, REORGANIZA EL ESPACIO FISICO DE LA BASE DE DATOS.

ADAULD/ADADCU - CREAN ARCHIVOS SECUENCIALES ESTANDARD PARA ACTUALIZAR CUALQUIERA DE LOS ARCHIVOS DE LA BASE DE DATOS. EL ADAULD NOS SIRVE PARA CARGAR EL ARCHIVO OTRA VEZ Y EL ADADCU PARA DESCOMPRIMIR LOS ARCHIVOS.

ADAFIX - EFECTUA UN RESPALDO DE LA BASE DE DATOS Y LA DEVUELVE A SU ESTADO INICIAL.

ADARES - CREA EL RESPALDO DE PROTECCION LOG ( QUE CONTIENE LOS COMANDOS DE MODIFICACION DE ARCHIVOS E INFORMACION DE LOS USUARIOS), PARA REGENERAR A LA BASE DE DATOS DESPUES DE HABERLO CARGADO NUEVAMENTE.

ADAREP - SE UTILIZA PARA DAR INFORMACION DEL ESTADO ACTUAL DEL CONTENIDO DE LA BASE DE DATOS, COMO : CANTIDAD Y ESPACIO DISPUESTO PARA EL ASOCIADOR Y EL AREA DE DATOS, ESPACIO DISPONIBLE PARA EL ASOCIADOR Y AREA DE DATOS, INFORMACION DE PUNTOS DE REINICIO, Etc.



## ADABAS



RESULTA  
FISICO M3-

### ARCHIVOS

ADABAS SE COMPONE DE HASTA 255 ARCHIVOS, PARA EL MANEJO DE LA MISMA SE UTILIZAN 5 ARCHIVOS APROXIMADAMENTE, POR LO CUAL, NOS QUEDAN 250 ARCHIVOS PARA EL USUARIO.

COMO SE NOMBRO ANTERIORMENTE EXISTEN DOS TIPO DE PASSWORD DE ACCESO PARA LA PROTECCION DE LA INFORMACION Y TRES PASSWORDS DE SEGURIDAD APLICADOS A LOS ARCHIVOS DE ADABAS.

POR CADA ARCHIVO SE PUEDEN TENER UN MAXIMO DE 16.7 MILLONES DE REGISTROS, Y EL NUMERO MAXIMO DE CAMPOS POR REGISTRO ES DE 500.

EL NUMERO MAXIMO DE DESCRIPTORES ( LLAVES ) ES HASTA DE 200 POR ARCHIVO.

LOS TIPOS DE ACOPLAMIENTO DE ARCHIVOS SON:

- ACOPLAMIENTO FISICO
- ACOPLAMIENTO LOGICO

EL ACOPLAMIENTO FISICO ( SIMULA UNA FUNCION DE JOIN ) PERMITE RELACIONAR DOS ARCHIVOS PARA EFECTOS DE CONSULTA CREANDO UN TERCER ARCHIVO QUE CONTIENE LAS RELACIONES DE ELLOS, POR LO TANTO REQUIERE DE ESPACIO EN EL ASOCIADOR. NO ES MUY USADO PORQUE GASTA MUCHOS RECURSOS.

EL ACOPLAMIENTO LOGICO PERMITE RELACIONAR DOS O MAS ARCHIVOS, SE LOGRA CON UNA INSTRUCCION DE BUSQUEDA ( FIND ). ESTE TIPO DE ACOPLAMIENTO NO REQUIERE ESPACIO EN EL ASOCIADOR PERO SI TIEMPO DE I/O

METODO DE ACCESO DIRECTO ( ADAM ) : ACCESO RANDOM OPTIMIZADO QUE REQUIERE DE UN CAMPO DESCRIPTOR QUE TENGA VALORES UNICOS Y DEBE DE ESTAR DEFINIDO AL PRINCIPIO DEL REGISTRO. LA VENTAJA MAS IMPORTANTE DE ACCESO ES QUE NO USA LISTAS INVERTIDAS NI EL ADDRESS CONVERTER PARA ENCONTRAR LA INFORMACION.

# ADABAS

## REGISTROS

UN REGISTRO ES CADA UNO DE LOS MIEMBROS EN QUE SE DIVIDE UN ARCHIVO. CADA REGISTRO CONSTA DE UN MAXIMO DE 500 CAMPOS.

CADA UNO DE LOS REGISTROS ES IDENTIFICADO POR UN NUMERO DE SECUENCIA INTERNO (ISN: INTERNAL SEQUENCE NUMBER) QUE ES UNICO EN EL ARCHIVO.

EXISTEN DOS CAMPOS QUE VAN ACOMPAÑADOS A CADA REGISTRO:

- LONGITUD DEL REGISTRO
- ISN

EL TAMAÑO DE LOS REGISTROS ES DE UN MAXIMO DE 10 Kbytes SIN COMPRESION Y 3 Kbytes CON COMPRESION SIN TOMAR EN CUENTA EL CODIGO DE DESCOMPRESION.

PARA LOS NOMBRES DE CAMPO **ADABAS** MANEJA LA SIGUIENTES REGLAS:

- DEBE DE CONSISTIR DE 2 CARACTERES, EL PRIMERO ALFABETICO Y EL SEGUNDO ALFANUMERICO
- LAS UNICAS COMBINACIONES QUE NO PUEDEN SER USADAS PARA NOMBRES DE CAMPO SON DE E0 - E9, YA QUE SON LOS NOMBRES RESERVADOS PARA LAS MASCARAS DE EDICION.

LOS FORMATOS PARA LOS CAMPOS DE **ADABAS** SON:

TIPO DE FORMATO	DESCRIPCION	LONGITUD MAXIMA
F	DECIMAL PUNTO FIJO	4 Bytes
P	EMPACADO	14 Bytes
U	DESEMPACADO	27 Bytes
* A	ALFANUMERICO	253 Bytes
B	BINARIO	126 Bytes

\* NOTA: EN CASO DE QUE EL CAMPO SEA DESCRIPTOR SU LONGITUD MAXIMA DEBE SER DE 126 Bytes.

## ADABAS

LOS NIVELES DE LOS CAMPOS DE ADABAS PUEDEN SER DE 1 HASTA 7 Y SU SECUENCIA DEBE DE SER CONTINUA.

LOS TIPOS DE CAMPOS QUE SE PUEDEN MANEJAR EN EL REGISTRO SON LOS SIGUIENTES:

- ELEMENTAL
- AGRUPADO
- CAMPO MULTIPLE
- GRUPO PERIODICO

EL CAMPO ELEMENTAL TIENE FORMATO Y LONGITUD EN EL MISMO NIVEL MIENTRAS QUE EL DE GRUPO NO SE ESPECIFICA.

CAMPO MULTIPLE (MU) - PERMITE DE 1-191 VALORES DISTINTOS DE UN CAMPO. SI ES DEFINIDO COMO DESCRIPTOR MANEJA UNA LISTA INVERTIDA. NO RESPETA LA POSICIONALIDAD SI ES NULO.

GRUPO PERIODICO (PE) - PERMITE DE 1-99 OCURRENCIAS, SOLO SE PERMITE UN GRUPO PERIODICO POR NIVEL, ES DECIR NO SE PERMITEN GRUPOS PERIODICOS ANIDADOS. SI ES DEFINIDO COMO DESCRIPTOR CADA OCURRENCIA SERA UN DESCRIPTOR . SE RESPETA LA POSICION.

UN DESCRIPTOR ES EL NOMBRE QUE **ADABAS** LES DA A LAS LLAVES Y SON CRITERIOS DE BUSQUEDA EN ORDEN CRECIENTE.

TIPOS DE DESCRIPTORES:

- DESCRIPTORES
- SUPER-DESCRIPTORES
- SUB-DESCRIPTORES
- DESCRIPTORES FONETICOS

## ADABAS

LOS DESCRIPTORES ESTAN CLASIFICADOS DE DOS FORMAS GENERALES:

- NORMAL: PUEDEN TENER VALORES DUPLICADOS.
- UNICO : SOLO TIENE VALORES UNICOS Y SE DEFINE AL PRINCIPIO DEL REGISTRO.

LOS SUPER-DESCRIPTORES SON VIRTUALES, SE COMPONEN HASTA DE 5 ELEMENTOS DE VARIOS, DE LOS MISMOS O PARTES DE CAMPOS Y SE DEFINEN AL FINAL DEL ARCHIVO. SOLO GENERAN UNA LISTA INVERTIDA.

LOS SUB-DESCRIPTORES ES IGUAL QUE UN SUPER-DESCRIPTOR PERO PARA UN SOLO CAMPO (PUEDEN SER SOLO UNA PARTE DE LA LLAVE O TENER LA MISMA LONGITUD DEL CAMPO AL CUAL SE HACE REFERENCIA).

LOS DESCRIPTORES FONETICOS IDENTIFICAN REGISTROS CON CONTENIDO ALFANUMERICO Y LOS ENCUENTRA POR EL PARECIDO DEL SONIDO. FUNCIONA EN LENGUA ANGLOSAJONA.







## ADABAS

### ASOCIADOR

EL ASOCIADOR EL UNO DE LOS TRES CONJUNTOS DE DATOS DE ADABAS EL CUAL ESTA COMPUESTO POR EL ADDRESS CONVERTER Y LAS LISTAS INVERTIDAS.

#### ADDRESS CONVERTER:

EL ADDRESS CONVERTER ES LA RELACION QUE TIENE EL NUMERO DE BLOQUE DEL DATA STORAGE CON EL ISN DEL REGISTRO. A LOS NUMEROS DEL ADDRESS CONVERTER SE LES LLAMA RABN (RELATIVE ADABAS BLOCK NUMBER) Y CONTIENE LA DIRECCION FISICA DEL BLOQUE. EXISTE UN ADDRESS CONVERTER POR ARCHIVO.

#### LISTAS INVERTIDAS:

LAS LISTAS INVERTIDAS ALMACENAN EN ORDEN CRECIENTE LOS DIFERENTES VALORES DE UNA LLAVE, EL CAMPO DE CUENTA Y LOS ISN'S EN ORDEN ASCENDENTE.

HAY UNA LISTA LISTA INVERTIDA POR CADA DESCRIPTOR

DENTRO DEL ASOCIADOR TAMBIEN SE ENCUENTRAN:

- LA TABLA DE DEFINICION DE CAMPOS (FDT) EN LA CUAL ESTAN CONTENIDOS LOS NOMBRES Y ATRIBUTOS DE LOS CAMPOS (LONGITUD, COMPRESION, NIVEL, ETC...)

- LA DESCRIPCION DE ARCHIVOS EN LA CUAL SE DEFINE LA ESTRUCTURA FISICA DEL ARCHIVO Y SUS CARACTERISTICAS. ESTA DESCRIPCION SE DEFINE CUANDO SE DA DE ALTA EL ARCHIVO.

- LAS TABLAS DSST (DATA STORAGE SPACE TABLE) Y ASST (ASSOCIATOR STORAGE TABLE).

EN LAS SIGUIENTES PAGINAS SE MUESTRAN EJEMPLOS DE COMO SE VEN AFECTADOS EL ADDRESS CONVERTER, LAS LISTAS INVERTIDAS Y EL DATA STORAGE CUANDO SE EFECTUAN ACTUALIZACIONES EN DISTINTOS CAMPOS. \*



**ADDRESS  
CONVERTER**

PARA EL ISN 1	2
PARA EL ISN 2	1
PARA EL ISN 3	2
PARA EL ISN 4	1
PARA EL ISN 5	1
PARA EL ISN 6	4
PARA EL ISN 7	4
PARA EL ISN 8	3
PARA EL ISN 9	3
PARA EL ISN 10	2
PARA EL ISN 11	3

B L O C K  
1

ISN	COLOR	MARCA	MODELO
5	AMARILLO	FORD	SEDAN
2	AZUL	CHEVY	WAGON
4	VERDE	FORD	SEDAN

B L O C K  
2

3	ROJO	PONTIAC	DEPORTIVO
10	AMARILLO	CADILLAC	DEPORTIVO
1	AZUL	BMW	SEDAN

B L O C K  
3

8	ROJO	CHEVY	WAGON
9	VERDE	FORD	DEPORTIVO
11	AZUL	LINCOLN	SEDAN

B L O C K  
4

6	AMARILLO	BMW	SEDAN
7	ROJO	OLDSMOBILE	DEPORTIVO

**EJEMPLO DE LA FORMACION DEL ADDRESS CONVERTER**

## ADABAS

EN LAS PAGINAS 96 Y 97 SE MUESTRA COMO SE EFECTUAN ACTUALIZACIONES EN DISTINTOS CAMPOS QUE SON DESCRIPTORES.

EN LA PAGINA 98 SE MUESTRA COMO SE BORRA UN REGISTRO. CABE MENCIONAR QUE EN ESTE CASO SI EL ISN FUESE REUSABLE EN VEZ DE QUE QUEDE VACIO EL RABN QUE LE CORRESPONDIA DENTRO DEL ADDRESS CONVERTER , SE VOLVERIA A UTILIZAR CUANDO OTRO REGISTRO SE DIERA DE ALTA.

EN LA PAGINA 99 SE PUEDE VER QUE ES LO QUE PASA CUANDO SE EFECTUA UN CAMBIO A UN CAMPO QUE NO ES DESCRIPTOR.

EN LA PAGINA 100 SE MUESTRA CUANDO SE AGREGA UN NUEVO REGISTRO. COMO SE ACABA DE MENCIONAR SI EL ARCHIVO SE DEFINE CON SUS ISN'S REUSABLES EL ESPACIO QUE QUEDABA VACIO DENTRO DEL ADDRESS CONVERTER DEL EJEMPLO DE LA PAGINA 98 SE OCUPARIA CON EL RABN CORRESPONDIENTE A ESTE NUEVO REGISTRO.

LA PAGINA 101 MUESTRA LA ARQUITECTURA DEL ASOCIADOR CUANDO EXISTEN DESCRIPTORES, SUPERDESCRIPTORES, Y SUBDESCRIPTORES.

**LISTA INVERTIDA POR NOMBRE**

VALOR DE CAMPO DE LA LLAVE CUENTA ISN S

JINENEZ	1	11
JUAREZ	3	3,6,7
RAMIREZ	4	4,5,9,10
SOLIS	3	1,2,8

**LISTA INVERTIDA POR ANTIGUEDAD**

1	5	2,5,8,9,11
3	4	1,3,4,10
5	2	6,7

**ADDRESS  
CONVERTER**

ISN 1	3	B L O C K 1
ISN 2	4	
ISN 3	1	
ISN 4	2	
ISN 5	3	
ISN 6	1	B L O C K 2
ISN 7	2	
ISN 8	1	
ISN 9	3	B L O C K 3
ISN 10	4	
ISN 11	2	

**DATA STORAGE**

ISN	NOMBRE	SALARIO	ANTIGUEDAD
3	JUAREZ	25,000	3
8	SOLIS	20,000	1
6	JUAREZ	35,000	5

11	JINENEZ	28,000	1
4	RAMIREZ	26,000	3
7	JUAREZ	33,500	5

5	RAMIREZ	20,000	1
1	SOLIS	21,000	3
9	RAMIREZ	26,000	1

**B L O C K  
4**

2	SOLIS	25,000	1
10	RAMIREZ	33,000	3

LISTA INVERTIDA POR NOMBRE, LISTA INVERTIDA POR ANTIGUEDAD, ADDRESS CONVERTER Y DATA STORAGE ORIGINALES

ADDRESS  
CONVERTER

DATA STORAGE

LISTA INVERTIDA POR NOMBRE

JIMENEZ	1	11	
JUAREZ	3	3,6,7	
LOPEZ	1	5	
RAMIREZ	3	4,9,10	
SOLIS	3	1,2,8	

LISTA INVERTIDA POR ANTIGUEDAD

1	5	2,5,8,9,11
3	4	1,3,4,10
5	2	6,7

3
4
1
2
3
1
2
1
3
4
2

B L O C K  
1

B L O C K  
2

B L O C K  
3

B L O C K  
4

ISN	NOMBRE	SALARIO	ANTIGUEDAD
3	JUAREZ	25,000	3
8	SOLIS	20,000	1
6	JUAREZ	35,000	5

11	JIMENEZ	28,000	1
4	RAMIREZ	26,000	3
7	JUAREZ	33,500	5

5	LOPEZ	20,000	1
1	SOLIS	21,000	3
9	RAMIREZ	26,000	1

2	SOLIS	25,000	1
10	RAMIREZ	33,000	3

CUANDO EL NOMBRE DEL ISN 5 ES CAMBIADO POR 'LOPEZ'

EN ESTE CASO, EL ISN 5 QUE ANTERIORMENTE TENIA EL NOMBRE DE 'RAMIREZ' ES CAMBIADO POR 'LOPEZ' EN EL DATA STORAGE, COMO SE APRECIA, EL UNICO CAMBIO QUE SE PRESENTA ES DENTRO DE LA LISTA INVERTIDA POR NOMBRE, DEBIDO AL CAMBIO DE VALOR EN LA LLAVE, SE INTRODUCE EL VALOR DE 'LOPEZ' Y EL ISN CORRESPONDIENTE (5) SE BORRA DEL VALOR 'RAMIREZ' Y EL CAMPO DE CUENTA SE DECREMENTA EN UNO.

ADDRESS  
CONVERTIER

DATA STORAGE

LISTA INVERTIDA POR NOMBRE

JIMENEZ	1	11
JUAREZ	3	3,6,7
LOPEZ	1	5
RAMIREZ	3	4,9,10
SOLIS	3	1,2,8

LISTA INVERTIDA POR ANTIGUEDAD

1	4	5,8,9,11
3	5	1,2,3,4,10
5	2	6,7

3
4
1
2
3
1
2
1
3
4
2

BLOCK  
1

BLOCK  
2

BLOCK  
3

BLOCK  
4

ISN	NOMBRE	SALARIO	ANTIGUEDAD
3	JUAREZ	25,000	3
8	SOLIS	20,000	1
6	JUAREZ	35,000	5

11	JIMENEZ	28,000	1
4	RAMIREZ	26,000	3
7	JUAREZ	33,500	5

5	LOPEZ	20,000	1
1	SOLIS	21,000	3
9	RAMIREZ	26,000	1

2	SOLIS	25,000	3
10	RAMIREZ	33,000	3

CUANDO LA ANTIGUEDAD DEL ISN 2 ES CAMBIADA POR '3' PARA ESTE CASO, EL ISN 2 QUE TENIA ANTIGUEDAD 1 EN EL DATA STORAGE ES CAMBIADO POR 3, COMO SE MUESTRA, SOLO SE PRESENTA CAMBIO DENTRO DE LA LISTA INVERTIDA POR ANTIGUEDAD, DEBIDO AL CAMBIO DE VALOR DE LA LLAVE, AHORA COMO YA EXISTE VALOR, SE INCREMENTA EL CAMPO DE CUENTA EN 1 Y SE BORRA EL ISN 2 DE LA LISTA DE ISN'S DEL VALOR '1'.

**ADDRESS  
CONVERTER**

**DATA STORAGE**

**LISTA INVERTIDA POR NOMBRE**

JINENEZ	1	11
JUAREZ	3	3,6,7
LOPEZ	1	5
RAMIREZ	3	4,9,10
SOLIS	2	1,2

**LISTA INVERTIDA POR ANTIGUEDAD**

1	3	5,9,11
3	5	1,2,3,4,10
5	2	6,7

3
4
1
2
3
1
2
3
4
2

B L O C K  
1

B L O C K  
2

B L O C K  
3

B L O C K  
4

ISN	NOMBRE	SALARIO	ANTIGUEDAD
3	JUAREZ	25,000	3
6	JUAREZ	35,000	5

11	JINENEZ	20,000	1
4	RAMIREZ	26,000	3
7	JUAREZ	33,500	5

5	LOPEZ	20,000	1
1	SOLIS	21,000	3
9	RAMIREZ	26,000	1

2	SOLIS	25,000	3
10	RAMIREZ	33,000	3

**CUANDO EL ISN 8 ES BORRADO**

EN ESTE CASO EL ISN 8 DESAPARECE DEL BLOCK 1 Y, POR LO TANTO, EN EL ADDRESS CONVERTER QUEDA VACIO EL DASH, SE BORRA EL ISN 8 DE LA LISTA INVERTIDA POR NOMBRE Y POR ANTIGUEDAD DECREMENTANDO LOS CAMBIOS DE CUENTA DE CADA LISTA EN 1.

ADDRESS  
CONVERTER

DATA STORAGE

LISTA INVERTIDA POR NOMBRE

JIMENEZ	1	11
JUAREZ	3	3,6,7
LOPEZ	1	5
RAMIREZ	3	4,9,10
SOLIS	2	1,2

LISTA INVERTIDA POR ANTIGUEDAD

1	3	5,9,11
3	5	1,2,3,4,10
5	2	6,7

3
4
1
2
3
1
2
3
4
2

B L O C K  
1

B L O C K  
2

B L O C K  
3

B L O C K  
4

ISN	NOMBRE	SALARIO	ANTIGUEDAD
3	JUAREZ	25,000	3
6	JUAREZ	38,000	5

11	JIMENEZ	20,000	1
4	RAMIREZ	26,000	3
7	JUAREZ	33,500	5

5	LOPEZ	20,000	1
1	SOLIS	21,000	3
9	RAMIREZ	26,000	1

2	SOLIS	25,000	3
10	RAMIREZ	33,000	3

CUANDO EL SALARIO DEL ISN 6 ES CAMBIADO POR '38,000'  
EN ESTE CASO COMO EL CAMPO SALARIO NO ES DESCRIPTOR NO HAY CAMBIOS EN EL ASOCIADOR SOLO EN EL DATA STORAGE.

**ADDRESS  
CONVERTER**

**DATA STORAGE**

**LISTA INVERTIDA POR NOMBRE**

JIMENEZ	1	11
JUAREZ	3	3,6,7
LOPEZ	2	5,12
RAMIREZ	3	4,9,10
SOLIS	2	1,2

**LISTA INVERTIDA POR ANTIGUEDAD**

1	4	5,9,11,12
3	5	1,2,3,4,10
5	2	6,7

3
4
1
2
3
1
2
3
4
2
1

B L O C K  
1

B L O C K  
2

B L O C K  
3

B L O C K  
4

ISN	NOMBRE	SALARIO	ANTIGUEDAD
3	JUAREZ	25,000	3
6	JUAREZ	38,000	5
12	LOPEZ	29,500	1

11	JIMENEZ	28,000	1
4	RAMIREZ	26,000	3
7	JUAREZ	33,500	5

5	LOPEZ	20,000	1
1	SOLIS	21,000	3
9	RAMIREZ	26,000	1

2	SOLIS	25,000	3
10	RAMIREZ	33,000	3

CUANDO SE AGREGA EL SIGUIENTE REGISTRO EN EL BLOCK 1  
CON ISN 12 :LOUIS, 29,500 , 1.

EN ESTE CASO SE AGREGA EL REGISTRO CON ISN 12, SE ANADE DENTRO DE LAS LISTAS INVERTIDAS Y DEL ADDRESS CONVERTER



**ADDRESS  
CONVERTER**

**DATA STORAGE**

**LISTA INVERTIDA POR COLOR**

AMARILLO	3	5,6,10
AZUL	3	1,2,11
ROJO	3	3,7,8
VERDE	2	4,9

**LISTA INVERTIDA POR MODELO**

DEPORTIVO	4	3,7,9,10
SEDAN	5	1,4,5,6,11
WAGON	2	2,8

**LISTA INVERTIDA DEL SUPER-  
DESCRIPTOR COLOR/MODELO**

AMARILLO DEPORTIVO	1	10
AMARILLO SPORT	2	5,6
AZUL SEDAN	2	1,11
AZUL WAGON	1	2
ROJO DEPORTIVO	2	3,7
ROJO WAGON	1	8
VERDE DEPORTIVO	1	9
VERDE SEDAN	1	4

**LISTA INVERTIDA DEL SUB-DE.**

**PRIMER BYTE DE MARCA**

B	2	1,6
C	3	2,8,10
F	3	4,5,9
L	1	11
O	1	7
P	1	3

2
1
2
1
1
4
4
3
3
2
3

B L O C K  
1

B L O C K  
2

B L O C K  
3

B L O C K  
4

ISN	COLOR	MARCA	MODELO
5	AMARILLO	FORD	SEDAN
2	AZUL	CHEVY	WAGON
4	VERDE	FORD	SEDAN

3	ROJO	PONTIAC	DEPORTIVO
10	AMARILLO	CADILLAC	DEPORTIVO
1	AZUL	BMW	SEDAN

8	ROJO	CHEVY	WAGON
9	VERDE	FORD	DEPORTIVO
11	AZUL	LINCOLN	SEDAN

6	AMARILLO	BMW	SEDAN
7	ROJO	OLDSMOBILE	DEPORTIVO

**EJEMPLO DE LA FORMACION DE LISTAS INVERTIDAS DE  
DESCRIPTORES, SUPERDESCRIPTORES Y SUB-DESCRIPTORES**

# ADABAS

## TIPOS DE ACCESO

EXISTEN DOS TIPOS DE ACCESO DE DATOS EN ADABAS EL SECUENCIAL Y EL ALEATORIO.

### ACCESO SECUENCIAL

EN ADABAS LA LECTURA EN ORDEN SECUENCIAL PUEDE HACERSE POR TRES TIPOS:

- READ FISICO
- READ POR ISN
- READ LOGICO

Ej.

ISN	NOMBRE	OCUPACION-1	OCUPACION-2
5	SOLIS	OPERADOR	
3	BRAVO	PROGRAMADOR	ANALISTA
1	JIMENEZ	ENTRENADOR	
2	ZAPATA	OPERADOR	PROGRAMADOR
4	MARTINEZ	ANALISTA	

SECUENCIA FISICA: 5, 3, 1, 2, 4

SECUENCIA POR ISN'S: 1, 2, 3, 4, 5

SECUENCIA LOGICA: 3, 1, 4, 5, 2  
(POR NOMBRE)

SECUENCIA LOGICA: 3, 4, 1, 2, 5, 2, 3  
(POR OCUPACION)

## ADABAS

- READ FISICO: ES EL MAS RAPIDO Y NO GASTA ACCESO AL DISCO YA QUE ESTE LO HACE EN EL ORDEN EN EL QUE SE ENCUENTRAN LOS REGISTROS EN EL DATA STORAGE. NO MANEJA NINGUN CRITERIO DE ORDENAMIENTO.

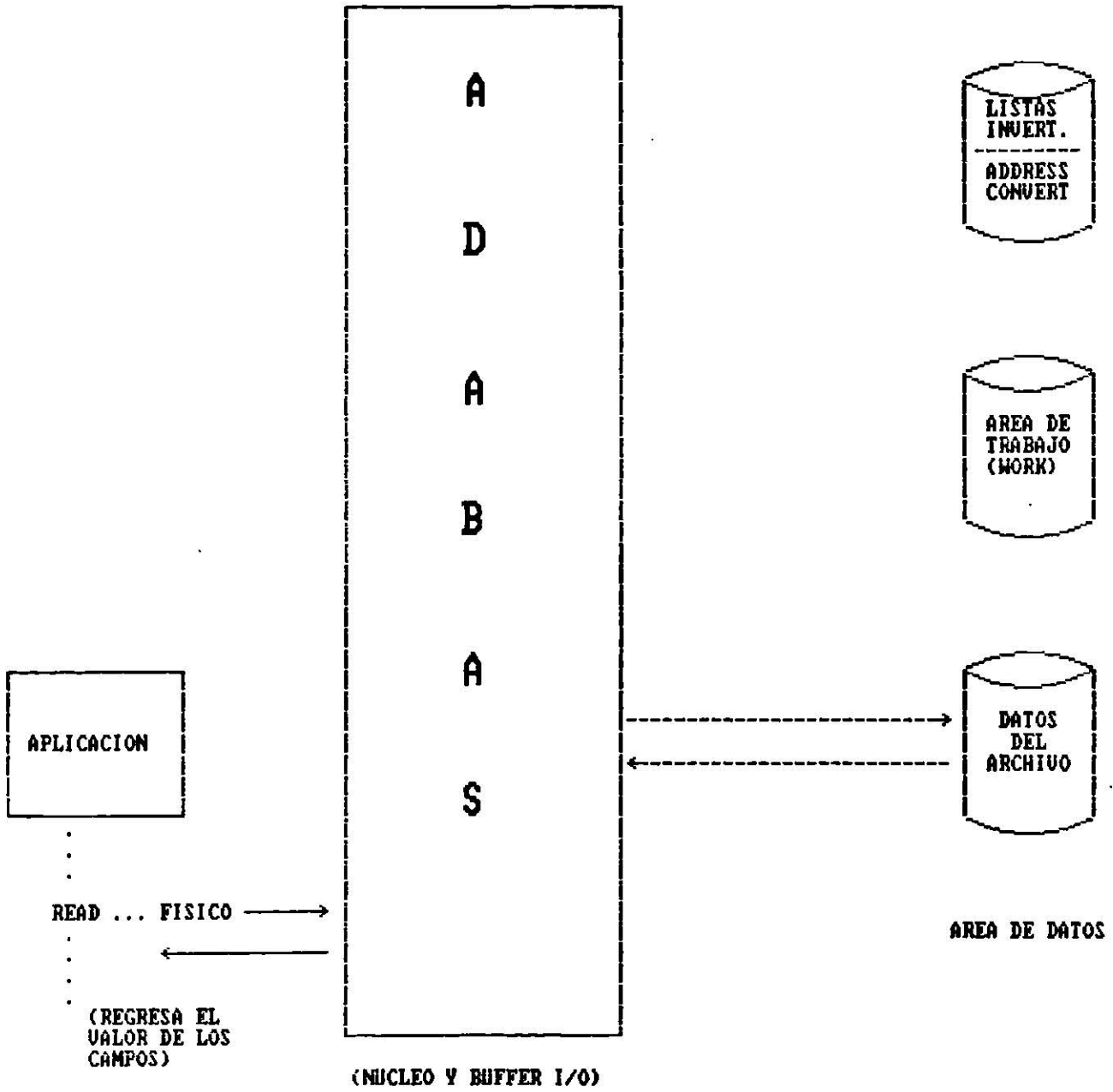
- READ POR ISN: ESTA LECTURA LA HACE EN EL ORDEN ASCENDENTE DE LOS ISN'S. ES MAS COSTOSO QUE EL READ FISICO YA QUE UTILIZA EL ADDRESS CONVERTER.

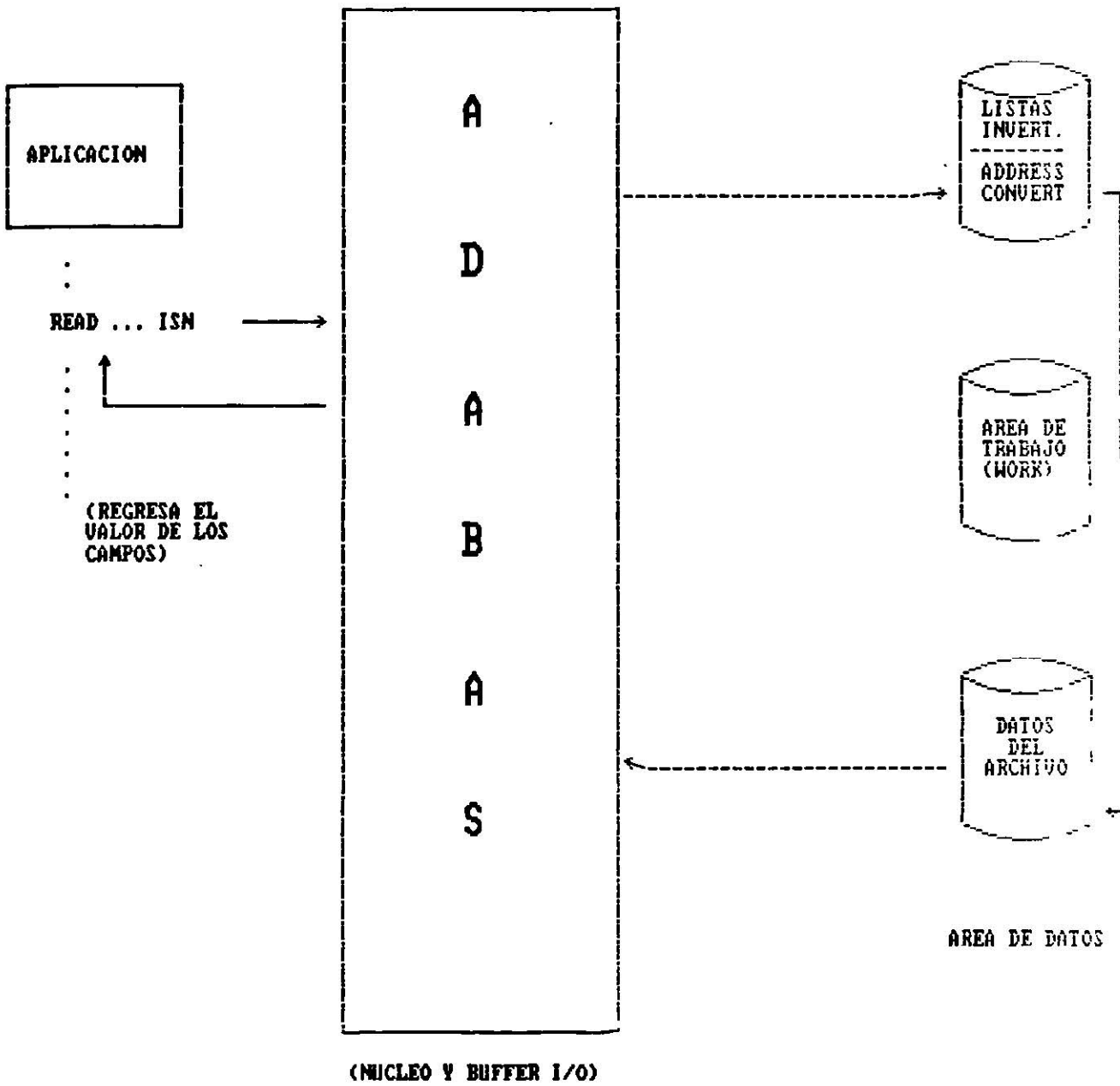
- READ LOGICO: ESTA LECTURA ES LA QUE GASTA MAS RECURSOS EN LA BASE DE DATOS, YA QUE UTILIZA LAS LISTAS INVERTIDAS PARA IDENTIFICAR LOS ISN'S DE LOS REGISTROS SELECCIONADOS Y DE AHI ACCESA EL ADDRESS CONVERTER PARA OBTENER LA DIRECCION DEL BLOCK EN DONDE SE ENCUENTRAN LOS REGISTROS.

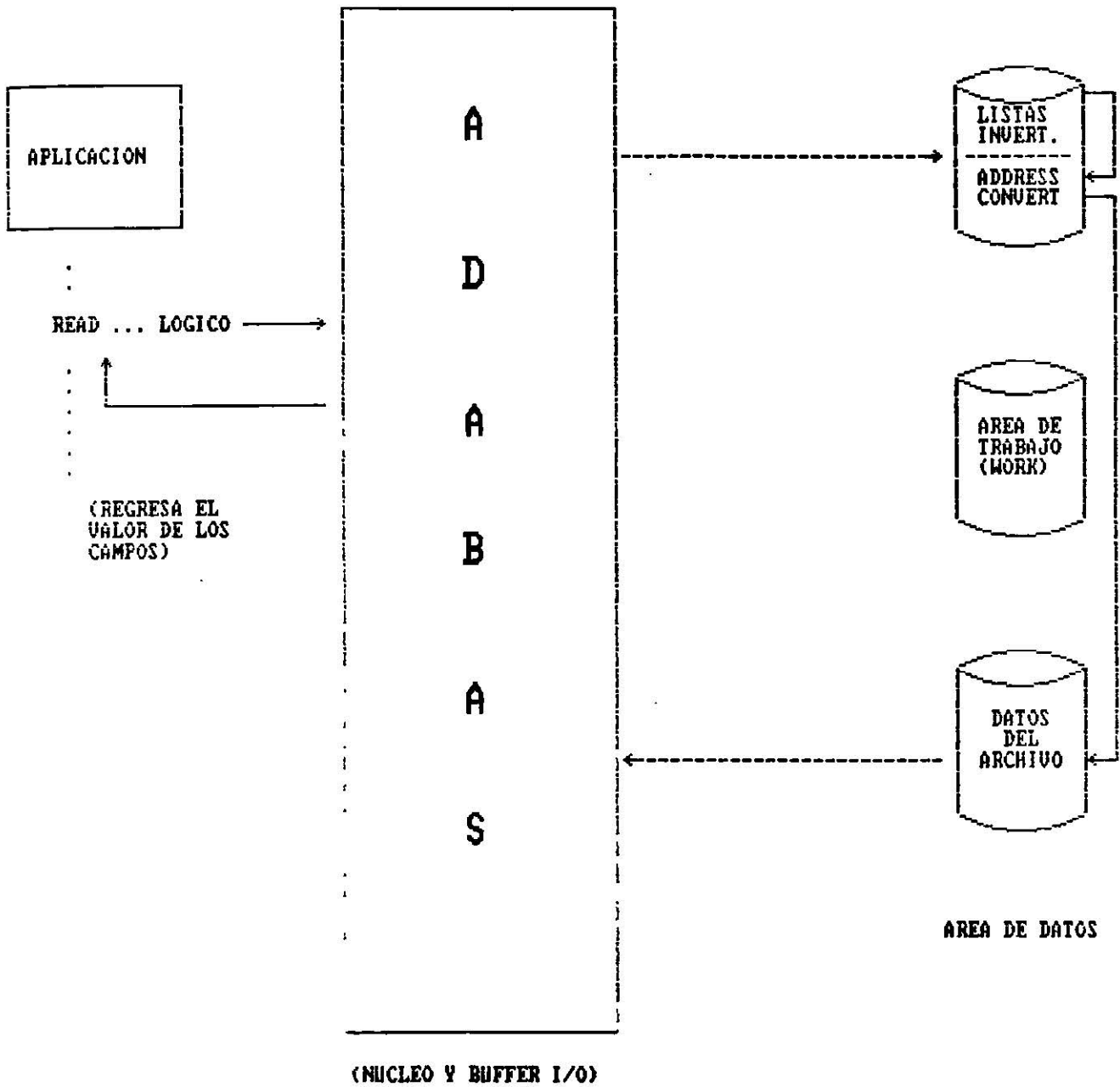
EN LAS PAGINAS SIGUIENTES SE MUESTRAN LAS GRAFICAS DE CADA LECTURA.

READ FISICO

ASOCIADOR







## ADABAS

### ACCESO ALEATORIO

ESTE TIPO DE ACCESO SE LLEVA A CABO POR MEDIO DE BUSQUEDAS CON DESCRIPTORES.

PARA LOCALIZAR UN REGISTRO USANDO UN DESCRIPTOR ACUDE A LA LISTA INVERTIDA, TOMA EL ISN, VA AL ADDRESS CONVERTER PARA BUSCAR EL NUMERO DE BLOQUE DONDE SE ENCUENTRA EL DATO EN EL DATA STORAGE.

PARA LOCALIZAR UN REGISTRO USANDO DOS DESCRIPTORES, TOMA LA PRIMERA LISTA INVERTIDA, BUSCA LOS ISN'S Y LOS GUARDA EN EL AREA DE DATOS, DESPUES VA A LA LISTA INVERTIDA DEL SEGUNDO DESCRIPTOR, BUSCA LOS ISN'S Y TAMBIEN LOS GUARDA EN EL AREA DE DATOS, ENTONCES SE ELIMINAN LOS REGISTROS QUE NO TIENEN LAS MISMAS CARACTERISTICAS Y CON LOS REGISTROS RESULTANTES SE GENERA UN TERCERA LISTA INVERTIDA Y EN BASE AL ADDRESS CONVERTER VA AL DATA STORAGE A BUSCAR LOS REGISTROS QUE CUMPLAN CON LA CONDICION DE AMBOS DESCRIPTORES.

ESTE ACCESO SE HACE POR MEDIO DE BUSQUEDAS CON LA INSTRUCCION FIND Y LA INSTRUCCION GET . EN EL FIND SE ENCUENTRA LA CANTIDAD DE REGISTROS Y EL GET POR MEDIO DE EL ISN TRAE EL REGISTRO. LA INSTRUCCION GET USA LA LISTA DE ISN'S DEL FIND Y LEE LOS REGISTROS UNO A LA VEZ.

EL ACCESO ALEATORIO PUEDE SER DE DIFERENTES TIPOS:

- CRITERIO DE BUSQUEDA SIMPLE: SOLO MANEJA UN DESCRIPTOR, POR LO TANTO SOLO MANEJA UNA LISTA INVERTIDA.

- CRITERIO DE BUSQUEDA COMPLEJA: ESTE MANEJA MAS DE UN DESCRIPTOR, POR LO TANTO MANEJA VARIAS LISTAS INVERTIDAS.

EN ADABAS LA LECTURA DE ACCESO ALEATORIO PUEDE HACERSE POR:

- FIND: SE EFECTUA EN BASE A CRITERIOS Y CONSTRUYE UNA LISTA DE ISN'S BASADA EN ESOS CRITERIOS. LA LISTA DE ISN'S LA GUARDA EN EL AREA DE TRABAJO (WORK AREA) CON UN IDENTIFICADOR.

- FIND CON SORT: ESTE NOS PRESENTA LOS DATOS EN UN CIERTO ORDEN Y SE PUEDE CLASIFICAR HASTA POR 3 DESCRIPTORES DE UN MISMO ARCHIVO. LA CLASIFICACION LA HACE EN EL AREA DE TRABAJO.

## ADABAS

- GET POR UN ESPECIFICO ISN: ESTE TIPO DE ACCESO LO HACE POR MEDIO DE UN DETERMINADO ISN DADO POR EL USUARIO.

- FIND CON GET: BUSCA EL CONJUNTO DE ISN'S Y ENTONCES HACE LA LECTURA POR MEDIO DEL GET.

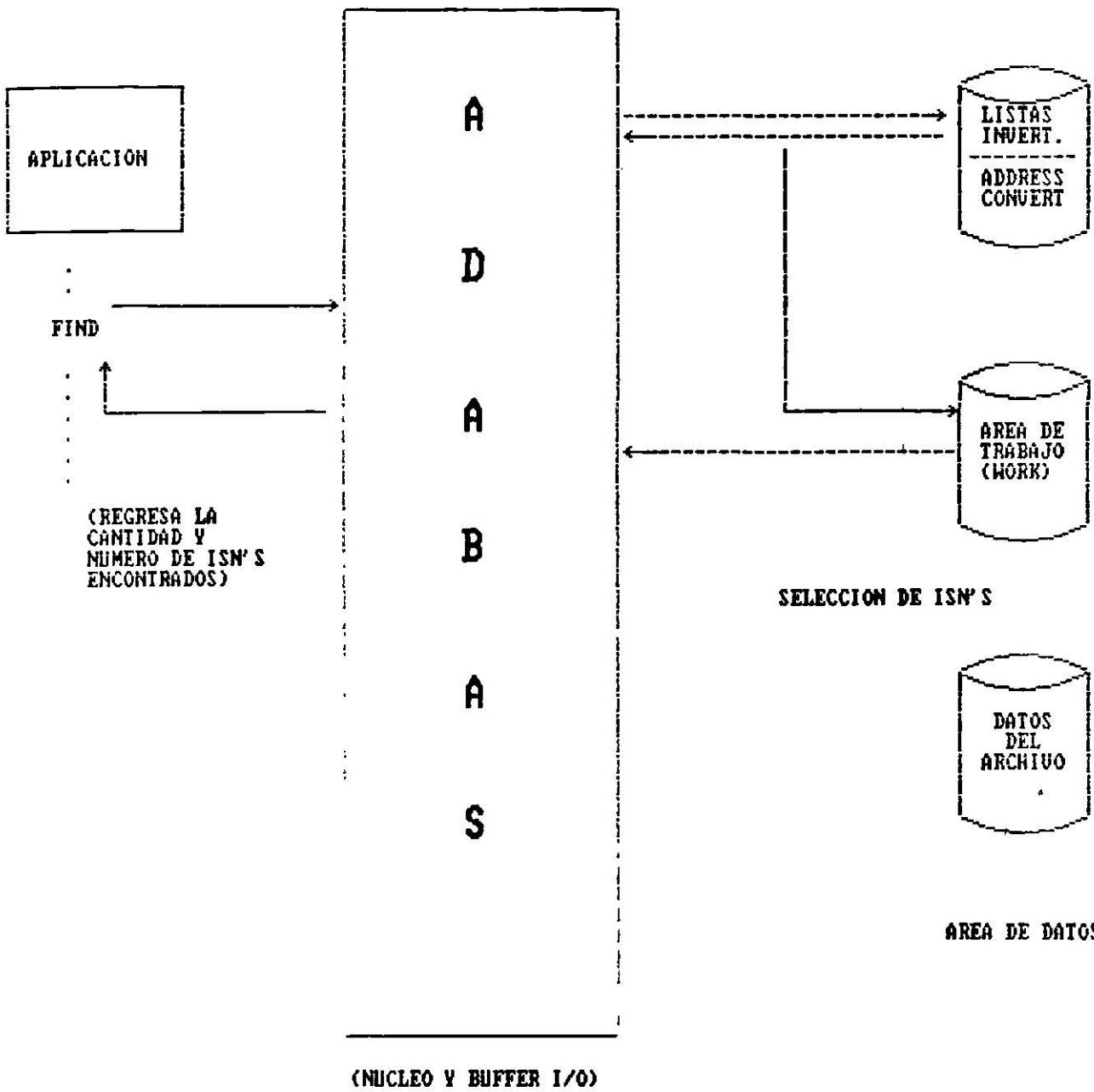
EXISTE ADEMAS OTRO TIPO DE ACCESO A LA BASE DE DATOS QUE ES EL ACCESO AL ASOCIADOR EN EL CUAL SE TOMAN LOS VALORES DE LAS LISTAS INVERTIDAS, LLAMADO HISTOGRAM, ESTE TIPO DE ACCESO REGRESA LOS VALORES DE LOS DESCRIPTORES Y LA CANTIDAD DE CADA VALOR ( CAMPO DE CUENTA).

EN LAS PAGINAS SIGUIENTES SE MUESTRAS LAS GRAFICAS DE ACCESO ALEATORIO.

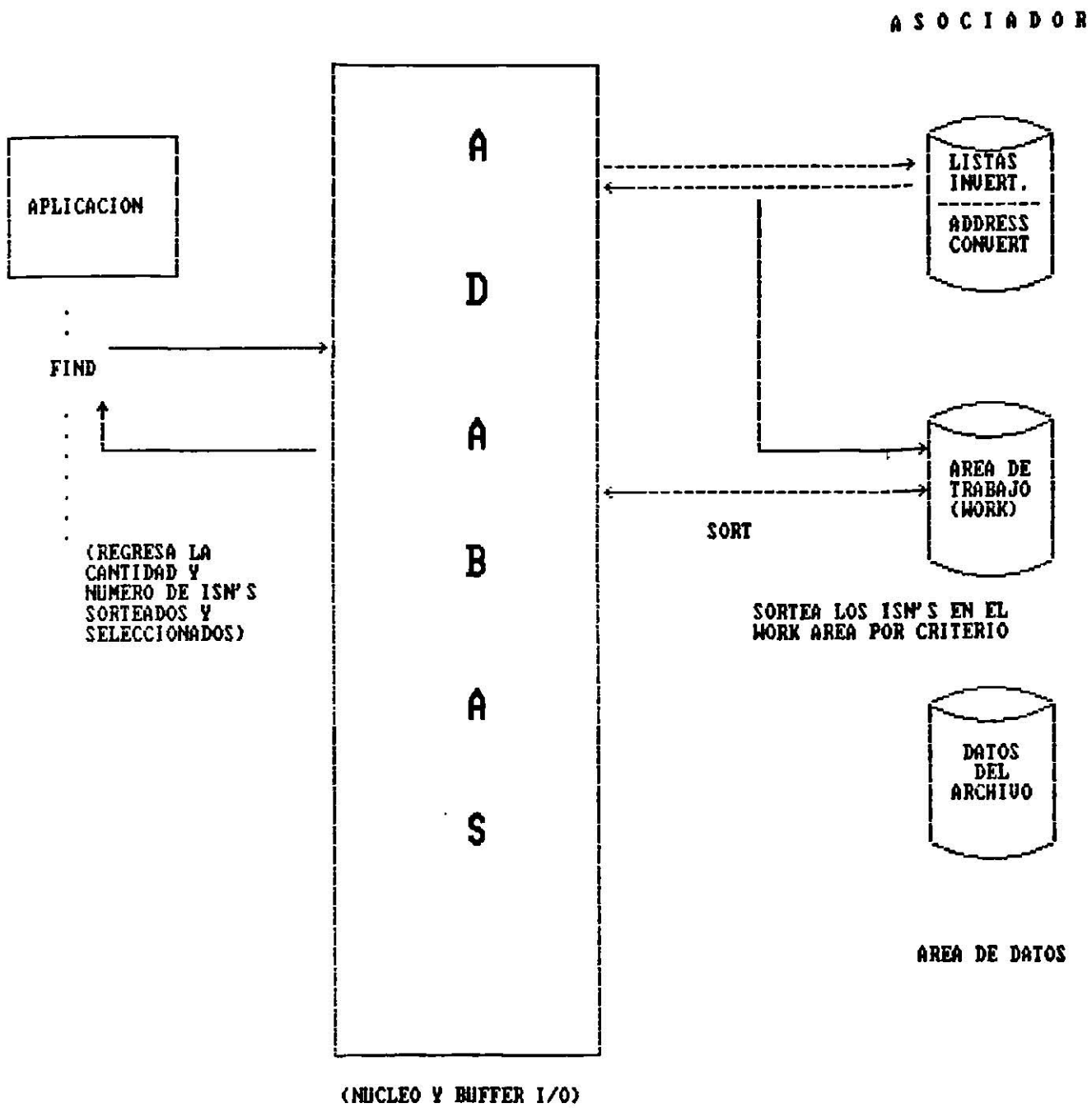


FIND

ASOCIADOR

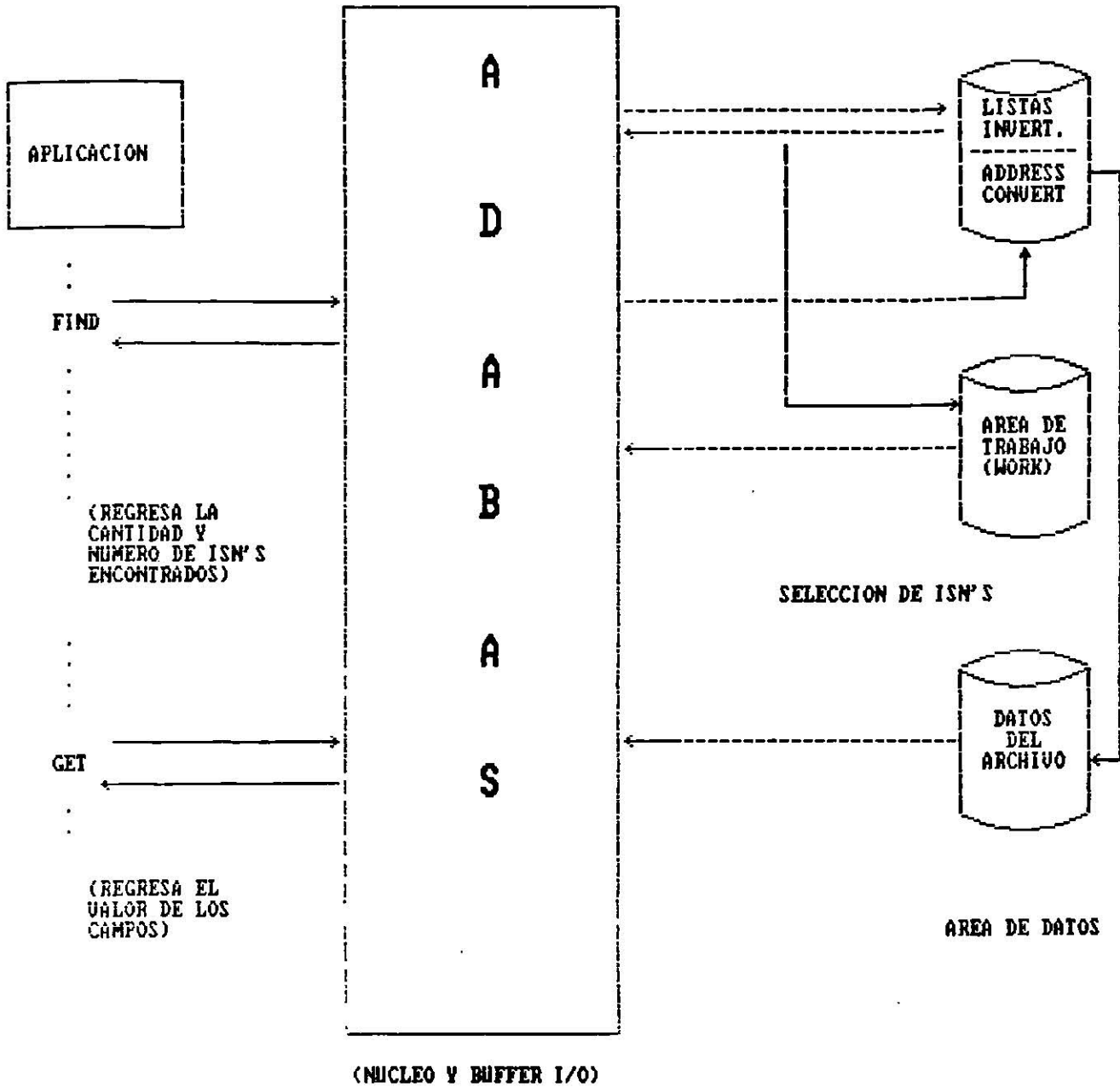


FIND CON SORT

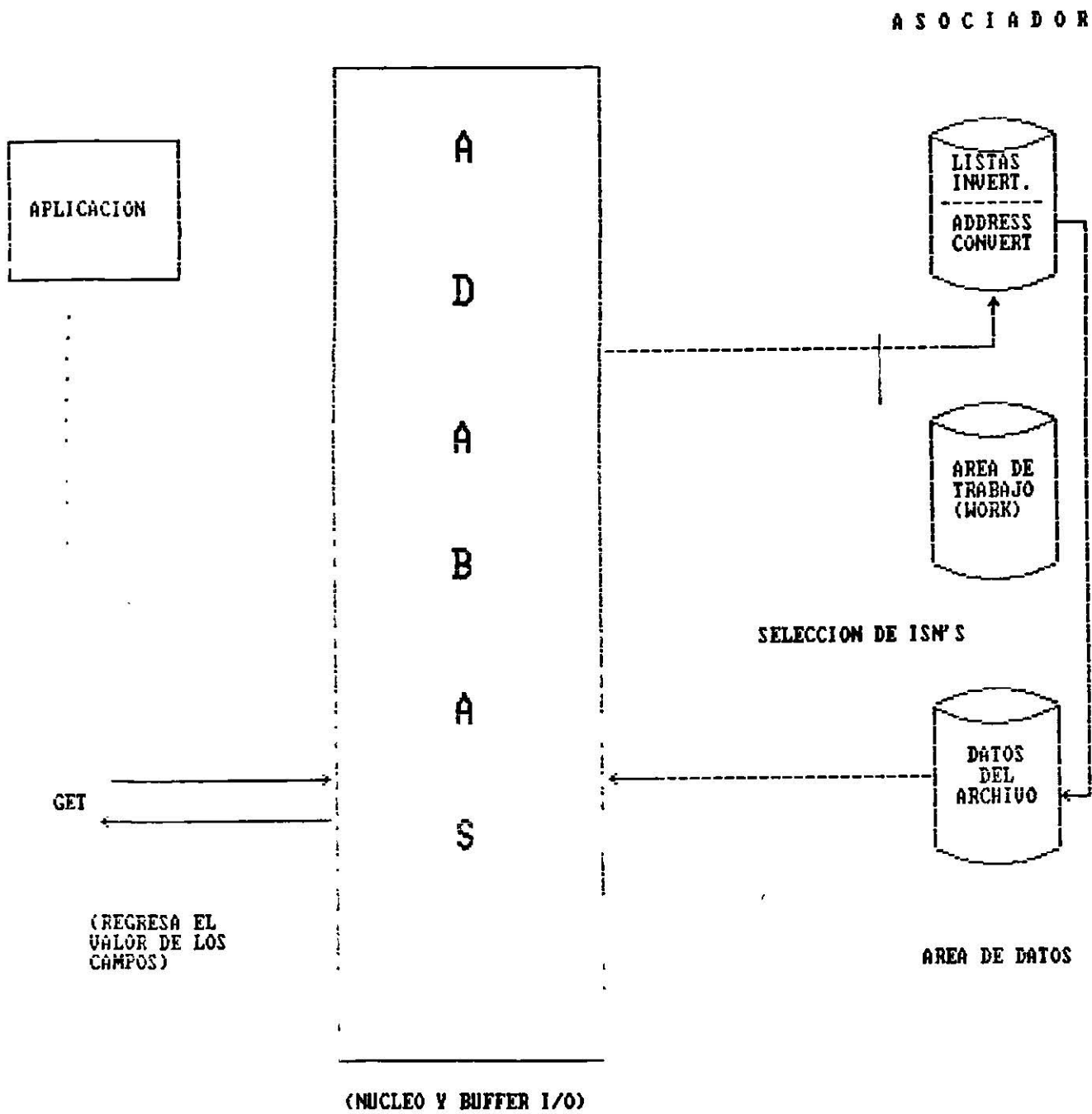


FIND CON GET

A S O C I A D O R

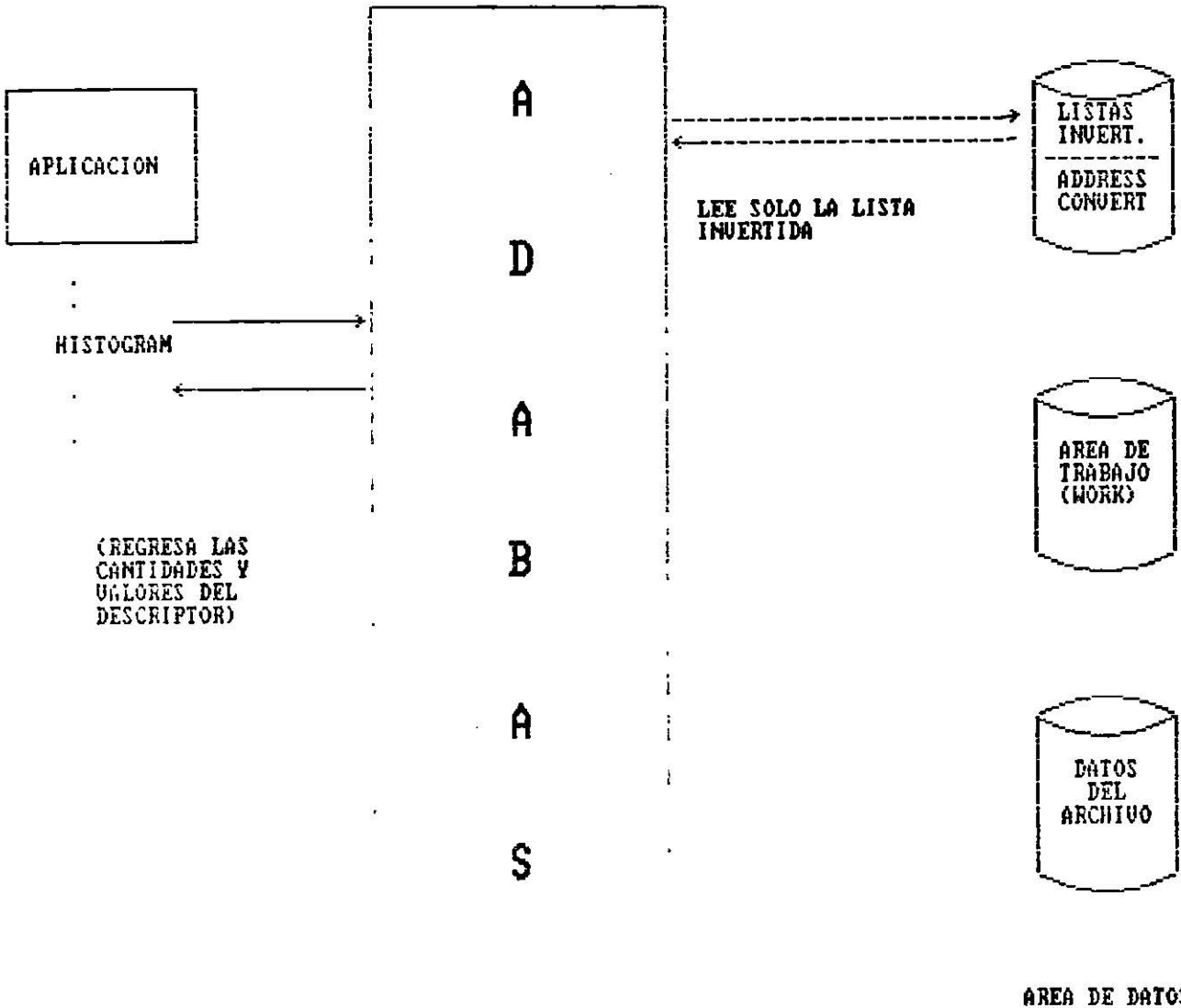


GET CON ISN'S SELECCIONADOS



HISTOGRAM (READ - LISTAS INVERTIDAS)

A S O C I A D O R



(NUCLEO Y BUFFER I/O)

## ADABAS

### CONSIDERACIONES SOBRE ACCESOS

PARA HACER ACTUALIZACIONES DE REGISTROS DENTRO DE LA BASE DE DATOS LOS REGISTROS PASAN A ESTADO DE ESPERA (HOLD) PARA CADA USUARIO. EN EL CASO EN QUE OTRO USUARIO QUIERA ACTUALIZAR UN REGISTRO QUE SE ENCUENTRA EN HOLD TENDRA QUE ESPERAR A QUE EL PRIMER USUARIO QUE LO ESTA UTILIZANDO LO LIBERE O QUE **ADABAS** LO LIBERE CUANDO SE TERMINE EL TIEMPO LIMITE DE TRANSACCION (TTL) PARA EVITAR DEADLOCK . AL PONER UN REGISTRO EN HOLD CONSERVAMOS LA INTEGRIDAD DE LA INFORMACION.

PARA SABER LA RAZON DE EXITO O FRACASO DE UNA TRANSACCION ADABAS UTILIZA LOS CODIGOS DE RESPUESTA , PARA EL CASO YA MENCIONADO, LOS CODIGOS DE RESPUESTA MAS FRECUENTES SON:

- 145: INDICA QUE EL REGISTRO ES DE USO EXCLUSIVO PARA EL CASO DE ACTUALIZACION.
- 009: INDICA QUE SE TERMINO EL TIEMPO LIMITE DE NO ACTIVIDAD Y **ADABAS** LIBERA EL REGISTRO.

OTROS CODIGOS COMUNES SON:

- 000: INDICA QUE LA OPERACION ES CORRECTA.
- 148: INDICA QUE LA BASE DE DATOS NO SE ENCUENTRA ACTIVA.

PARA INDICAR CUANDO TERMINA UNA TRANSACCION , YA SEA DE ALTA, BAJA, O MODIFICACION DE LOS REGISTROS SE UTILIZA LA INSTRUCCION ET (END TRANSACTION) ,LA CUAL, LIBERA TODOS LOS REGISTROS QUE SE ENCONTRABAN EN HOLD POR EL USUARIO Y EJECUTA TODAS LAS ACTUALIZACIONES.

OTRA INSTRUCCION QUE SE UTILIZA EN ESTOS CASOS ES LA INSTRUCCION BT (BACKOUT TRANSACTION) QUE SE ENCARGA TAMBIEN DE LIBERAR LOS REGISTROS QUE SE ENCONTRABAN EN HOLD PERO SIN EFECTUAR LAS ACTUALIZACIONES QUE SE HICIERON EN LA TRANSACCION.

## ADABAS

### REINICIO Y RECUPERACION DE INFORMACION DE LA BASE DE DATOS

EL PROPOSITO DEL REINICIO Y RECUPERACION DE INFORMACION DE LA BASE DE DATOS ES ASEGURAR LA INTEGRIDAD, ES DECIR, QUE SE CONSERVEN LAS AREAS Y LIMITES CORRECTOS ENTRE ARCHIVOS Y EVITAR EL HACER UN REPROCESO.

#### TIPOS DE USUARIOS DE ADABAS:

- SOLAMENTE DE ACCESO: ESTE USUARIO SOLO PUEDE LEER INFORMACION, NO ESTA AUTORIZADO PARA HACER ACTUALIZACIONES.
- FILE CLUSTER: (CONJUNTO DE ARCHIVOS QUE ESTAN DEDICADOS A UN USUARIO) MANEJA CIERTO NUMERO DE ARCHIVOS DONDE EL USUARIO SE ENCARGA DE ORGANIZARLOS PARA HACER UN CHECK-POINT (ADABAS NO LO HACE).
- ET LOGICO: ES PARA USUARIOS QUE RESPETAN LA LOGICA DE LA TRANSACCION, ADABAS HACE AUTOMATICAMENTE EL CHECK-POINT. TIENE DOS TIEMPOS LIMITES : TIEMPO DE NO ACTIVIDAD Y TIEMPO DE TRANSACCION (ESTOS TIPOS DE USUARIO SON LOS MAS COMUNES).
- CONTROL EXCLUSIVO: ES UN USUARIO DE ET LOGICO, PERO TODO UN ARCHIVO ESTA DEDICADO AL USUARIO Y NO TIENE RESTRICCIONES NI LIMITES DE TIEMPO.

#### TIEMPO LIMITES

SON PARAMETROS QUE SIEMPRE ESTAN PRESENTES EN LA BASE DE DATOS Y QUE AYUDAN A RESOLVER LA SITUACION DE DEADLOCK ( SITUACION DE INTERLOCK ), ESTOS PARAMETROS SE FIJAN EN LA INSTALACION, PERO SE PUEDEN MODIFICAR DINAMICAMENTE.

1.- TIEMPO LIMITE DE TRANSACCION (TTL) : SE DEFINE EN SEGUNDOS. CONSISTE EN EL TIEMPO QUE PUEDE TRANSCURRIR ENTRE EL PRIMER REGISTRO EN HOLD Y EL END OF TRANSACTION.

2.- TIEMPO LIMITE DE NO-ACTIVIDAD: TAMBIEN SE DEFINE EN SEGUNDOS. CONSISTE EN EL TIEMPO QUE PUEDE TRANSCURRIR ENTRE COMANDOS Y LA BASE DE DATOS.

# ADABAS

## REINICIALIZACION DE DATOS

PERMITE A LOS PROGRAMAS AUTO-REINICIALIZARSE ATRAVES DE ALMACENAMIENTO ESPECIFICADO POR EL USUARIO DE MODO QUE PUEDA RECONOCER LA INFORMACION Y ACTUAR EN CONSECUENCIA.

EL GET TRANSACTION DATA SE UTILIZA PARA RECUPERAR LA INFORMACION EN EL SISTEMA DE ARCHIVOS Y SE PUEDE EJECUTAR EN CUALQUIER MOMENTO.

EL SIGUIENTE EJEMPLO MUESTRA LOS DISTINTOS CASOS EN LOS QUE PUEDEN REINICIALIZARSE LOS DATOS:

Ej.

NO.	NO. PARTE	DESCRIPCION	CANTIDAD
1	FR78-14	PUERTAS	123,000
2	6X39-1	RINES	45,234
3	518M	LLANTAS	230,456
:	:	:	:
:	:	:	:
47	R14Z	RADIADORES	740,568
48	RSX360	TANQUES DE GAS	40,000

- A) CADA LINEA ES DEFINIDA COMO UNA TRANSACCION LOGICA.  
SI EL SISTEMA FALLA DURANTE EL REGISTRO 47, EL OPERADOR COMIENZA EN EL REGISTRO 47.
- B) HASTA EL FINAL ES DEFINIDA COMO UNA TRANSACCION LOGICA.  
SI EL SISTEMA FALLA DURANTE EL REGISTRO 47, EL OPERADOR DEBERA EMPEZAR EN EL REGISTRO 1.
- C) CADA 10 REGISTROS SE DEFINE UNA TRANSACCION LOGICA.  
SI EL SISTEMA FALLA DURANTE EL REGISTRO 47, EL OPERADOR DEBERA EMPEZAR EN EL REGISTRO 41.



## ADABAS

**ADABAS TIENE DOS TIPOS DE RECUPERACION DE INFORMACION:**

- AUTOMATICA
- MANUAL

LA RECUPERACION AUTOMATICA SE LLAMA AUTOBACKOUT, ES INVOCADA AL INICIO DE LA SESION DE **ADABAS**. EL USUARIO RECUPERARIA INFORMACION ACTUALIZADA HASTA EL ULTIMO END TRANSACTION.

LA RECUPERACION MANUAL SE HARIA EN CASO DE QUE EL DISCO DONDE RESIDE LA BASE DE DATOS ESTUVIERA DAÑADO. **ADABAS** ESTA PROVISTA DE UNA IMAGEN DE LA BASE DE DATOS (DUMP) QUE SUPLIRIA A LA DAÑADA. CON RESPECTO A LA INFORMACION QUE EL USUARIO GUARDA EN LA BASE DE DATOS, TAMBIEN ES RECUPERADA POR QUE **ADABAS** GUARDA UN RESPALDO LLAMADO PROTECTION LOG, ESTE RESPALDO SE PUEDE CARGAR POR MEDIO DE UNA UTILERIA.

A N E X O

## A N E X O

COMO UNA INFORMACION ADICIONAL PRESENTAMOS UNA COMPARACION DE ALGUNOS SISTEMAS DE MANEJO DE BASES DE DATOS MUNDIALMENTE RECONOCIDAS CON RESPECTO AL ACCESO Y EL MANEJO DE DATOS, SEGURIDAD, TRANSFERENCIA CLASIFICACION Y EQUIPO.

PARA HACER ESTA COMPARACION SE ELIGIERON , ENTRE OTRAS, LOS SIGUIENTES MANEJADORES DE BASES DE DATOS: IMS DE IBM, TOTAL DE CINCOM, SYSTEM 2000 DE INTEL-MRI E IDMS DE CULLINANE. EN LA TABLA DE LA PAG. ??? SE ENCUENTRAN ALGUNAS CARACTERISTICAS DE ALGUNOS DE LOS SISTEMAS DE MANEJO DE BASES DE DATOS DISPONIBLES EN EL MERCADO, ESTA TABLA SE HIZO CON EL PROPOSITO DE DARLE AL LECTOR UNA MAYOR VISION DE LAS ESTRUCTURAS Y TIPOS DE ACCESO QUE PRODUCEN LAS CASAS DE SOFTWARE QUE OPERAN PRINCIPALMENTE EN COMPUTADORAS IBM 360, 370, 30XX Y 43XX.

LOS TIPOS DE ACCESO MAS UTILIZADOS SON LOS DE TIPO RED, JERARQUICA E INVERTIDA. ENTRE LOS SISTEMAS DE MANEJO DE BASES DE DATOS QUE UTILIZAN EL ACCESO TIPO RED ENCONTRAMOS A TOTAL, IMS Y DMS DE BURROGHS CABE MENCIONAR QUE LA MAYORIA DE LOS SISTEMAS SE APEGAN AL ESTANDAR DE RED CODASYL COMO IDS II DE HONEYWELL, DMS 1100 DE UNIVAC, IDMS DE CULLINANE, DBMS/10 DE DEC Y PHOLAS DE PHILLIPS, ESTA ESTANDARIZACION HACE MAS VIABLE LA TRANSFERENCIA DE PROGRAMAS DE APLICACION DE UN SISTEMA DE MANEJO DE BASES DE DATOS A OTRO, YA QUE LA INVERSION MONETARIA DE SOFTWARE PARA ACOPLAR PROGRAMAS DE APLICACION DE UN SISTEMA DE MANEJO DE BASES DE DATOS PARTICULAR Y LA DIFICULTAD O EL COSTO DE TRANSFERIRLOS A OTRO ES TAN GRANDE EN MUCHAS INSTALACIONES ACTUALES, QUE LA TRANSFERENCIA A CUALQUIER BASE DE DATOS ES CASI IMPOSIBLE, A EXCEPCION DE CODASYL A CODASYL.

LA POSIBILIDAD DE TRANSFERENCIA DE UN SISTEMA DE MANEJO DE BASES DE DATOS JERARQUICO (COMO IMS) O DE UNA DE RED (COMO TOTAL) NO ES TAN FACIL POR LO GENERAL, COMO LA TRANSFERENCIA DE UN SISTEMA DE MANEJO DE BASES DE DATOS INVERTIDO( COMO LO ES ADABAS ) O DE LOS RELACIONALES QUE SE ENCUENTRAN EN LA TABLA DE LA PAG. 123.

NUMEROSAS BASES DE DATOS PRESENTAN LAS FACILIDADES PARA INDEXAMIENTO INVERTIDO PARA UN ACCESO MAS FLEXIBLE Y EFECTIVO EN CUANTO A COSTOS, A UN SUBCONJUNTO DEL BANCO DE DATOS QUE SE SELECCIONA EN BASE A CLASIFICACIONES COMPLEJAS. LOS EXPONENTES CLASICOS SON LOS SISTEMAS DE MANEJO DE BASES DE DATOS INVERTIDOS COMO LO SON ADABAS, SISTEM 2000 Y DATACOM/ DB QUE PROPORCIONAN PODEROSOS LENGUAJES DE CONSULTA Y OTROS SISTEMAS DE MANEJO DE BASES DE DATOS INCORPORAN CIERTO TIPO DE CAPACIDAD DE INVERSION.

## A N E X O

UNO DE LOS PRINCIPALES OBJETIVOS DE UN SISTEMA DE MANEJO DE BASES DE DATOS ES PROPORCIONAR UN ALTO GRADO DE SEGURIDAD A LOS DATOS, IMS CUENTA CON NIVELES DE PROTECCION, CODASYL INCLUYE UN REPERTORIO EXTENSO DE MECANISMOS DE SEGURIDAD, ADEMAS DE LOS MECANISMOS DE LOS SUBESQUEMAS QUE A SU VEZ SON MAS FLEXIBLES QUE LOS DE IMS, LOS SISTEMAS INVERTIDOS TIENEN ESTRUCTURAS DE DIRECTORIOS DEL BANCO DE DATOS QUE PUEDEN SER MUY UTILES EN LA IMPLEMENTACION DE CONTROLES DE SEGURIDAD A NIVEL DE LOS CAMPOS Y A UN NIVEL DE LOS VALORES DE LOS CAMPOS, ADABAS ES UNA DE LOS POCOS SISTEMAS DE MANEJO DE BASES DE DATOS QUE PROPORCIONAN MECANISMOS DE SEGURIDAD POR CODIFICACION/DECODIFICACION (CIFRAMIENTO) , SOLO UNOS CUANTOS SISTEMAS DE MANEJO DE BASES DE DATOS SENSIBLES HAN JUSTIFICADO TAL GRADO DE SEGURIDAD Y SU COSTO CORRESPONDIENTE.

EN CUANTO A BASES DE DATOS RELACIONALES QUE SON LAS DE MAS AUGE EN LA ACTUALIDAD, INVESTIGAMOS LAS SIGUIENTES: ORACLE, PROGRESS Y DB2 DE IBM.

LA MAYORIA DE LAS BASES DE DATOS RELACIONALES TIENE LAS SIGUIENTES CARACTERISTICAS:

- INDEPENDENCIA DE DATOS
- SEGURIDAD E INTEGRIDAD DE LOS DATOS
- FACILIDADES DE RECUPERACION DE INFORMACION
- CONECTABILIDAD
- PORTABILIDAD
- DICCIONARIO DE DATOS
- LENGUAJE DE CUARTA GENERACION

LA ELECCION DE MANEJADORES DE BASE DE DATOS DEPENDERA DE LAS NECESIDADES DEL USUARIO Y DEL EQUIPO CON EL QUE DISPONGA.

CON ORACLE SE PUEDEN CREAR APLICACIONES EN UN EXTENSO GRUPO DE TRABAJO YA SEA EN REDES LOCALES O REMOTAS, ORACLE CUENTA CON LOS SIGUIENTES SERVIDORES: OS/2, UNIX, XENIX, BANYAN VINES, DEC VAX/VMS Y OTRAS MINIS, IBM MVS Y IBM VM.

ORACLE TIENE GRANDES FACILIDADES PARA CONECTAR VARIOS EQUIPOS SIENDO ESTO TRASPARENTE PARA EL USUARIO. SUS SERVIDORES ABREN UNA ARQUITECTURA QUE PERMITE VIRTUALMENTE A CUALQUIER COMPANIA DE SOFTWARE DE PC PROVEER UNA INTERFASE CON ELLOS.

## A N E X O

EL LENGUAJE DE CUARTA GENERACION DE ORACLE TIENE LAS SIGUIENTES HERRAMIENTAS:

- SQL\*FORMS SISTEMA DE DESARROLLO DE APLICACIONES INTERACTIVO
- SQL\*REPORTWRITER GENERADOR DE REPORTES
- SQL\*MENU GENERADOR DE MENUS
- SQL\*PLUS EL ESTANDAR ANSI DEL SQL QUERY, HERRAMIENTA DE ADMINISTRACION CON EXTENSIONES DE LENGUAJE PARA AD HOC QUERY Y ANALISIS DE DATOS.
- SQL\*LOADER CONVERTIDOR PARA FORMATO ORACLE
- PRO\*C PRECOMPILADOR EN LENGUAJE "C" E INTERFASE DE LLAMADAS EXTERNAS POR SUBROUTINAS.

ORACLE PUEDE HACER INTERFASE CON OTRAS BASES DE DATOS (p.e. DBASE), HOJAS DE CALCULO (p.e. LOTUS 1-2-3) Y OTROS PAQUETES, ADEMAS CUENTA CON UN CURSO DE ADIESTRAMIENTO PARA SQL.

EL MANEJADOR DE BASE DE DATOS PROGRESS CUENTA CON UN LENGUAJE DE CUARTA GENERACION QUE EVITA TENER QUE HACER PROCESOS DE I/O, ACCESOS DE ARCHIVOS, LAYOUT DE PANTALLAS Y REPORTES, RECUPERACIONES DE INFORMACION, REDUCE CONSIDERABLEMENTE EL CICLO DE EDICION-COMPILACION-EJECUCION.

PROGRESS TIENE PROTECCION AUTOMATICA DE ARCHIVOS Y REGISTROS QUE ASEGURAN LA INTEGRIDAD DE LOS DATOS Y PERMITE UN DESARROLLO RAPIDO DE APLICACIONES MULTI-USUARIO, ALMACENAMIENTO DE LONGITUD VARIABLE QUE MANTIENE CONTINUAMENTE LA COMPRESION DE LA BASE DE DATOS, DANDO AHORRO DE ESPACIO EN DISCO Y PROCESOS MAS RAPIDOS.

PROGRESS PUEDE SOPORTAR UNA GRAN CANTIDAD DE PLATAFORMAS DE CORRIDA, ENTRE LOS CUALES ESTAN VAX/MVS, MS-DOS, UNIX, XENIX, ULTRIX, AIX, A/UX, CTOS/BTOS Y PROTOCOLOS LAN (TALES COMO NETBIOS, TCP/IP, DECNET, SPX/IPX, etc). ADEMAS SE PUEDE DISTRIBUIR A TRAVES DE UNA AMBIENTE DE UNA RED HETEROGENEA, POR EJEMPLO DOS Y XENIX PUEDEN ACCESAR UNA BASE DE DATOS QUE SE CENTRA EN UNIX, PERMITIENDO QUE EL CPU GRANDE ACTUE COMO SERVIDOR DE LA BASE DE DATOS.

PROGRESS TIENE UNA HERRAMIENTA LLAMADA FAST TRACK QUE ES UN CONSTRUCTOR DE APLICACIONES A TRAVES DE OPCIONES DE MENU, ENTRADAS DE COMANDOS Y TECNICAS DE PUNTO Y BUSQUEDA, ADEMAS PROVEE EL SQL ESTANDAR EN ANSI Y UN DISPOSITIVO LLAMADO HOST LANGUAGE CALL (HLC) PARA LAS INTERFASES, TIENE EDITOR CHECADOR DE SINTAXIS, DICCIONARIO DE DATOS (QUE AL IGUAL QUE EN ADABAS, PROVEE UNA INTEGRIDAD REFERENCIAL QUE ES AUTOMATICAMENTE INCORPORADA); FORMATEADOR AUTOMATICO DE PANTALLAS Y LA FACILIDAD DE UN HELP INTERACTIVO.

## A N E X O

LA RECUPERACION DE INFORMACION EN CASO DE CAIDA DEL SISTEMA, SE HACE AUTOMATICAMENTE RECHAZANDO LAS TRANSACCIONES INCOMPLETAS, PRESERVANDO ASI LA INTEGRIDAD DE LOS DATOS.

POR ULTIMO, HABLAREMOS DEL MANEJADOR DE BASE DE DATOS DE LA IBM, EL DB2, EL CUAL PUEDE SER USADO PARA IMPLEMENTAR SISTEMAS DE SOPORTE DE DECISIONES, ASI COMO, APLICACIONES TRADICIONALES Y TAMBIEN DONDE HAYA REQUERIMIENTOS DE APLICACIONES Y ESTRUCTURAS DE DATOS QUE ESTAN SUJETAS A FRECUENTES CAMBIOS, PERMITIENDO GRAN PRODUCTIVIDAD PARA PROGRAMADORES Y USUARIOS.

DB2 TRABAJA EN EQUIPOS IBM SISTEMA/ 370, PARA AMBIENTES VM/SP, VM/XA, MVS/SP, MVS/XA, MVS/ESA, VSE, IMV/VS, etc.

DENTRO DE LAS FUNCIONES, FACILIDADES Y CARACTERISTICAS DE DB2 ENCONTRAMOS LAS SIGUIENTES:

- PROVEE UNA COMPLETA CAPACIDAD DE RECUPERACION DE INFORMACION, EN CASO DE FALLA DEL SISTEMA, FALLA DE PROGRAMAS DE APLICACION O DE ALMACENAMIENTO.
- MULTIPLES USUARIOS PUEDEN ACCESAR Y HACER CAMBIOS A LOS DATOS AL MISMO TIEMPO ( AÑADIR, MODIFICAR O BORRAR), DEBIDO A QUE GARANTIZA LA CONSISTENCIA DE DATOS.
- ASEGURA QUE TODOS LOS ACCESOS Y MANIPULACION DE DATOS ( Y OTROS RECURSOS) ESTEN EN COMBINACION CON USUARIOS Y PROGRAMADORES, PERMITIENDO ESPECIFICACIONES DE SEGURIDAD DE DATOS BASADO HASTA EN EL CONTENIDO.
- CONTIENE UN CATALOGO ACCESIBLE PARA EL USUARIO, EL CUAL ES UN ALMACEN ACTIVO QUE CONTIENE INFORMACION ACERCA DE TODO EL SISTEMA.
- SELECCION DE TRAYECTORIAS DE ACCESO EJECUTADAS AUTOMATICAMENTE POR EL SISTEMA.
- INDEPENDENCIA DE DATOS QUE PERMITEN MUCHOS CAMBIOS LOGICOS Y FISICOS.
- EL LENGUAJE SQL QUE PROVEE FUNCIONES COMO PROMEDIOS, MINIMOS, MAXIMOS, CONTADORES, etc PARA DAR MAS PRODUCTIVIDAD AL PROGRAMA. EL SQL ( STRUCTURED QUERY LANGUAGE) ES UN LENGUAJE DE ALTO NIVEL DE ACCESO DE DATOS, Y ES USADO PARA PROGRAMAR; ESTE LENGUAJE ES EL COMPONENTE CLAVE DEL SAA (SYSTEMS APPLICATION ARCHITECTURE).
- UTILERIAS QUE PUEDEN SER INVOCADAS EN-LINEA Y PUEDEN SER EJECUTADAS SIN REQUERIR QUE LA BASE DE DATOS ESTE ACTIVA.

## A N E X O

- PROGRAMAS TRASPARENTES PARA EL USUARIO, QUE OFRECEN ASISTENCIA A USUARIOS QUE EMIGRAN DE DB2 A ARCHIVOS VSAM, Y PARA QUIENES NECESITAN TRASPORTAR APLICACIONES ENTRE AMBIENTES IBM QUE ESTEN COMPATIBLES CON EL SAA (PROGRAMA COMUN DE INTERFASE PARA BASE DE DATOS).
- BASE DE DATOS PARTICIONADAS, CON RAPIDAS RECUPERACIONES PARCIALES QUE PERMITEN MAYOR DISPONIBILIDAD.
- FACILIDADES EN LINEA DE ASITENCIA DEL HELP PARA TODAS LAS CATEGORIAS DE USUARIO.
- LA INTERFASE DL/1 BATCH QUE SOPORTA ACCESOS DE DB2 CON INTEGRIDAD DE LECTURA Y ESCRITURA.
- RAPIDOS SORTEOS DE GRANDES VOLUMENES DE DATOS Y CREACION DE INDICES QUE DAN MAYOR FLEXIBILIDAD.

DE ESTA MANERA, SE HAN EXPUESTO ALGUNAS DE LAS CARACTERISTAS DE LOS MANEJADORES DE BASES DE RELACIONES, COMO SE PUEDE OBSERVAR LA MAYORIA CUENTAN CON CARACTERISTICAS SIMILARES, LAS VENTAJAS DE CADA BASE DE DATOS DEPENDERAN, COMO SE MENCIONO ANTERIORMENTE DE LAS NECESIDADES DEL USUARIO.



ANEXO

PROVEEDOR	NOMBRE	CLASIFICACION	COMPUTADORA USADA	MINIMO DE MEMORIA Y SIST. OPERATIVO	LENGUAJE(S) DE PROG. ANFITRION(ES)	ESTRUCTURAS DE DATOS
BURROUGHS	DMS-II	RED	SERIE B 1700 -7800	80 K PALABRAS MCP	COBOL, PL/1, ALGOL	LISTAS ENCADENAMIENTO DE ARCHIVOS
CONTROL DATA	DMS-170	CODASYL DBTG	CYBER 70 CDC 6000	58K PALABRAS ESCOPE	ALGOL ENSAMBLADOR	RED DBTG
CINCOM	TOTAL	RED	BURROUGHS 2500 -4800 IBM 360, 370 IBM SYSTEM 3 HONEYWELL 200, 2000 UNIVAC SERIES 70, 9400/9700 NCR CENT. 101 CDC 6000 CYBER 70-74, 170-175 DEC PDP 11	MCPU, 80 K PA- LABRAS 3K A 32K BYTES DOS, OS, US1, US2 MOD1, MOD2 OS/2000 OS4, OS7, OS9  B1-B4  ESCOPE, KRONOS, NOS RSX-11D, IAS	COBOL, RPG II, PL/1, FORTRAN, BAL EN LA MAYO- RIA DE LAS COMPUTADORAS	RED
CULLINANE	IDMS	CODASYL DBTG	IBM 360, 370, 303X, 303XX	DOS, OS, US1, US2, 55K	COBOL, FORTRAN, PL/1	RED DBTG
DEC	DBMS/10	CODASYL DBTG	DEC PDP 10	DEC-10 MONITOR	COBOL, FORTRAN	RED DBTG
HONEYWELL	IDS/II	PREDECESOR ANILLADO DEL CODASYL DBTG	H400 H60/6000	64K PALABRAS GCOS	COBOL PARA EL DHL Y EL LDD	ANILLADA (RED)
IBM	CICS	SIS. DE CON- TROL DE CO- MUNICACIONES INTERACTIVAS	IBM 360, 370, 303X, 43XX	DOS 48K OS/US 2000 K	COBOL, PL/1 BAL	ACOPLAMIENTO CON IMS Y CONUENCIO- NAL (ALEATORIO SECUENCIAL)
IBM	IMS2 IMS/US	JERARQUICO	IBM 360/370 MODELOS 40 Y DEL 145 EN ADELANTE, 303X, 43XX	DOS 98K BYTES IMS 2 DB: 128K IMS 2 DB/DC: 512K IMS/US DB: 2000K IMS/US DB/DC: MAS DE 2000K	FORTRAN, COBOL, PL/1 Y BAL	JERARQUICO ENCADENAMIENTO DE ARCHIVOS
IBM	GIS/2	RECOPILACION DE INF. GENERACION DE REPORTES	IBM 360, 370, 303X, 43XX	OS/US 500K		ACOPLAMIENTO CON IMS Y CONUENCIO- NAL (SECUENCIAL ALEATORIO-SECUEN- CIAL INDIZADO)



ANEXO

PROVEEDOR	NOMBRE	CLASIFICACION	COMPUTADORA USADA	MINIMO DE MEMORIA Y SIST. OPERATIVO	LENGUAJE(S) DE PROG. ANFITRION(ES)	ESTRUCTURAS DE DATOS
INTEL-MIR	SYSTEM 2000	ARBOREO INVERTIDO	IBM 360, 370, 43XX CDC 6000 CYBER 70, 170 UNIVAC 1100	DOS OS/US SCOPE/KRONOS 20KP DE 60 BYTES EXEC 8 31KP	COBOL, FORTRAN, PL/1 Y BAL	ARBOREO INVERTIDO
PHILLIPS	PHOLAS	CODASYL DBTG	PHILLIPS P1000	64K, P1000 OS		RED DBTG
PHI (INFORMATICS)	METABAS	ARBOREO INVERTIDO	IBM 360/370	OS	COBOL, PL/1 FORTRAN Y BAL	ARBOREO INVERTIDO
SOFTWARE AG	ADABAS	ORIENTACION DE RED INVERTIDA	IBM 360, 370, 380X SIEMENS 4004 UNIVAC 9000 ICL SYSTEM 4 DEC 11	OS/DOS 110K	COBOL, FORTRAN, PL/1, BAL Y NATURAL	ORIENTACION DE RED INVERTIDA
UNIVAC	DMS 1100	CODASYL DBTG	SERIE UNIVAC 1100	60 PALABRAS DE 36 BITS	COBOL, PL/1, FORTRAN	RED DBTG
APPLICATIONS SOFTWARE	ASI-ST	SISTEMA GENERALIZADO PARA MANEJO DE ARCHIVOS	IBM 360, 370	DOS 44K, DOS/US OS 65K OS/US1, OS/US2	SUSTITUYE A LOS LENG. DE PROGRAMACION ANTERIORES	CONVENCIONAL, ARCHIVOS IMS Y TOTAL
INFORMATICS	MARK IV	SISTEMA GENERALIZADO PARA MANEJO DE ARCHIVOS	IBM 360/25 EN ADELANTE IBM 370/125 EN ADELANTE UNIVAC 90 SIEMENS 4004	DOS, OS, OS/US1 OS/US2 48K TDOS, DOS, OS/4 65K	SUSTITUYE A LOS LENG. DE PROGRAMACION ANTERIORES	CONVENCIONAL, ARCHIVOS IMS Y TOTAL
PROGRAMA PRODUCTS	DATA ANALYZER	SISTEMA GENERALIZADO PARA MANEJO DE ARCHIVOS	IBM 360/370	40K DOS 85K OS	SUSTITUYE A LOS LENG. DE PROGRAMACION ANTERIORES	CONVENCIONAL, ARCHIVOS IMS Y TOTAL

# G L O S A R I O

## GLOSARIO

### **ADABAS (ADAPTABLE DATA BASE SYSTEM):**

SISTEMA DE MANEJO DE BASE DE DATOS RELACIONAL DE ACCESO SIMPLIFICADO Y ALTA INTEGRIDAD DE DATOS CON UN EFECTIVO CONTROL DE SEGURIDAD.

### **ADDRESS CONVERTER (CONVERTIDOR DE DIRECCIONES):**

ES EL AREA DENTRO DEL ASOCIADOR EN LA QUE SE ENCUENTRA LA RELACION QUE TIENE EL NUMERO DE BLOQUE EN DONDE SE ENCUENTRA EL REGISTRO CON SU DETERMINADO ISN.

### **ARCHIVO:**

CONJUNTO DE REGISTROS QUE TIENEN RELACION ENTRE SI.

### **ASOCIADOR:**

ES UNO DE LOS GRUPOS DE DATOS DE ADABAS EN EL CUAL SE ENCUENTRAN LAS LISTAS INVERTIDAS FORMADAS POR LOS DESCRIPTORES Y EL CONVERTIDOR DE DIRECCIONES. SE ENCARGA DE TENER LA RELACION ENTRE LOS ISN'S, BLOQUES, REGISTROS Y DESCRIPTORES ENTRE SI.

### **BASE DE DATOS:**

AREA ESTRUCTURADA DE INFORMACIONES DENTRO DEL CPU QUE SON SISTEMATICAMENTE REGISTRADAS, BUSCADAS, ACCESADAS Y ACTUALIZADAS.

### **BIT:**

ESTA FORMADO POR LAS PALABRAS BINARY DIGIT (DIGITO BINARIO), QUE COMO SU NOMBRE LO INDICA ES UNA UNIDAD DE INFORMACION QUE PUEDE ADOPTAR DOS VALORES O ESTADOS DISTINTOS. ES LA UNIDAD MINIMA DE INFORMACION.

### **BLOQUE (BLOCK):**

UNIDAD FISICA DE INFORMACION QUE ESTA FORMADA POR UN CONJUNTO DE REGISTROS.

### **BT (BACKOUT TRANSACTION):**

INSTRUCCION DE ADABAS Y NATURAL QUE SE ENCARGA DE LIBERAR REGISTROS QUE SE ENCUENTRAN EN HOLD SIN EFECTUAR LAS ACTUALIZACIONES QUE SE HICIERON DENTRO DE LA TRANSACCION.

### **BUFFER:**

AREA DE ALMACENAMIENTO TEMPORAL QUE VARIA DE ACUERDO A COMO ESTA SE DEFINA Y SE ENCARGA DE CONTROLAR EL MANEJO DE LAS ENTRADAS Y SALIDAS DE INFORMACION.

### **BYTE:**

UNIDAD DE INFORMACION COMPUESTA POR BITS.

## GLOSARIO

### **CAMPO:**

UNIDAD DE INFORMACION EN LA CUAL SE DIVIDEN LOS REGISTROS DE UN ARCHIVO.

### **CAMPO MULTIPLE(MU):**

CONJUNTO DE CAMPOS REPETITIVOS DENTRO DE UN REGISTRO EN ADABAS EL CUAL PUEDE CONTENER DE 1 A 191 VALORES DISTINTOS DENTRO DE UN CAMPO.

### **CAMPO PERIODICO (PE):**

ES UN CONJUNTO DE CAMPOS REPETITIVOS DENTRO DE UN REGISTRO EN ADABAS EL CUAL PUEDE CONTENER VARIOS CAMPOS DENTRO DE UN CAMPO. PERMITE DE 1 A 99 OCURRENCIAS.

### **COMANDO:**

INSTRUCCIONES DENTRO DE NATURAL O ADABAS QUE SE UTILIZAN PARA EJECUTAR FUNCIONES RELACIONADAS CON LA CREACION, MANIPULACION, ALMACENAMIENTO Y EJECUCION DE PROGRAMAS.

### **COMPILADOR:**

PROGRAMA DE TRADUCCION QUE CONVIERTE INSTRUCCIONES DE ALTO NIVEL EN UN JUEGO DE INSTRUCCIONES BINARIAS PARA SU EJECUCION.

### **CPU:**

MODULO ENCARGADO DE BUSCAR, DECODIFICAR Y EJECUTAR INSTRUCCIONES DENTRO DE LA COMPUTADORA.

### **DESCRIPTOR:**

LLAVE DE ACCESO A UN ARCHIVO, EL CUAL, PUEDE SER UNICO O PUEDE TENER VALORES DUPLICADOS DEPENDIENDO DE COMO SE REQUIERA.

### **DESCRIPTOR FONETICO:**

LLAVE DE ACCESO A UN ARCHIVO QUE IDENTIFICA REGISTROS CON CONTENIDO ALFANUMERICO Y LOS ENCUENTRA POR EL PARECIDO DEL SONIDO.

### **EDITOR:**

PROGRAMA DESIGNADO PARA LA INTRODUCCION DE TEXTO DENTRO DEL NATURAL, EN EL CUAL, SE PUEDE MODIFICAR, BORRAR O INSERTAR LINEAS DENTRO DE UN PROGRAMA.

### **EFC (EMPTY FIELD COUNT):**

CAMPO DENTRO DE UN REGISTRO EN ADABAS QUE SE ENCARGA DE MARCAR LOS CAMPOS NULOS NO CONTIGUOS DENTRO DE UN REGISTRO.

## GLOSARIO

### **ENCRIPCION:**

METODO DE CODIFICACION QUE SE UTILIZA POR RAZONES DE SEGURIDAD. GENERALMENTE SE USAN EN TECNICAS DE CIFRADO MUY COMPLEJAS.

### **ESTATUTOS:**

INSTRUCCIONES DE LAS QUE SE FORMA UN PROGRAMA EN NATURAL.

### **ET (END TRANSACTION):**

INSTRUCCION DE ADABAS Y NATURAL QUE SE ENCARGA DE LIBERAR REGISTROS QUE SE ENCUENTRAN EN HOLD Y EJECUTAR TODAS LAS ACTUALIZACIONES QUE SE HICIERON EN LA TRANSACCION.

### **FORMATO:**

FORMA O DISPOSICION EN LA QUE SE ENCUENTRAN LOS DATOS.

### **ILB (INCLUSIVE LENGTH BYTE):**

PRIMER BYTE DENTRO DE UN CAMPO DE UN ARCHIVO ADABAS QUE GUARDA LA LONGITUD DEL CAMPO.

### **INTERLOCK:**

SITUACION QUE SE PRESENTA CUANDO DOS O MAS USUARIOS DE UNA BASE DE DATOS QUIEREN ACCESAR UN MISMO CONJUNTO DE REGISTROS Y CAUSA PERDIDA DE TIEMPO.

### **ISN (INTERNAL SEQUENCE NUMBER):**

NUMERO QUE LLEVA LA SECUENCIA INTERNA DE LOS REGISTROS DENTRO DE UN ARCHIVO ADABAS.

### **Kbyte (Kilobyte):**

UNIDAD DE INFORMACION QUE CONTIENE 1,024 Bytes.

### **LISTA INVERTIDA:**

AREA DENTRO DEL ASOCIADOR EN DONDE SE ALMACENAN EN ORDEN CRECIENTE LOS DIFERENTES VALORES DEL DESCRIPTOR, EL CAMPO DE CUENTA DE ESTE Y LOS ISN'S QUE CUMPLEN CON ESE VALOR.

### **LLAVE ADAM:**

LLAVE DE ACCESO UNICA A UN ARCHIVO.

### **NATURAL:**

LENGUAJE DE PROGRAMACION DE CUARTA GENERACION QUE SETA DISEÑADO PARA EL MANEJO DEL SISTEMA DE BASE DE DATOS ADABAS.

## GLOSARIO

**NUCLEO:**

PARTE PRINCIPAL DE LA BASE DE DATOS QUE SE ENCARGA DE ANALIZAR LA SINTAXIS Y CONGRUENCIA DE LAS INSTRUCCIONES.

**PADDING:**

FACTOR DE EXPANSION DE REGISTROS QUE PERMITE AGRANDAR LA LONGITUD DE LOS CAMPOS.

**PASSWORD:**

CLAVE SECRETA DE UN USUARIO QUE SE UTILIZA PARA DIFERENTES FINES.

**PREDICT:**

SISTEMA DE MANTENIMIENTO DE DICCIONARIO DE DATOS EL CUAL CONTIENE TODA LA INFORMACION DE LA ORGANIZACION DE LA BASE DE DATOS.

**PROGRAMA:**

CONJUNTO DE INSTRUCCIONES LOGICAS PARA OBTENER UN RESULTADO.

**RABN (RELATIVE ADABAS BLOCK NUMBER):**

NUMERO DENTRO DEL ADDRESS CONVERTER QUE CONTIENE LA DIRECCION FISICA DE UN BLOQUE.

**REGISTRO:**

MIEMBRO EN EL QUE SE DIVIDE UN ARCHIVO, EL CUAL, ESTA COMPUESTO POR CAMPOS.

**SISTEMA:**

CONJUNTO DE PROGRAMAS QUE TIENEN UN PROPOSITO EN COMUN.

**SUB-DESCRIPTOR:**

DESCRIPTOR COMPUESTO POR UN SOLO CAMPO O PARTES DEL MISMO CAMPO.

**SUPER-DESCRIPTOR:**

DESCRIPTOR COMPUESTO POR HASTA 5 CAMPOS DIFERENTES.

**TTL (TRANSACTION TIME LIMIT):**

ES EL TIEMPO QUE TRANSCURRE ENTRE EL PRIMER REGISTRO EN HOLD Y EL END TRANSACTION (ET).

## GLOSARIO

### **USER-VIEW(VISTAS LOGICAS):**

ARCHIVOS CONCEPTUALES QUE SON SUBCONJUNTOS DE ARCHIVOS FISICOS DE ADABAS.

### **UTILERIAS (UTILITIES):**

PROGRAMAS DENTRO DE ADABAS QUE SE ENCARGAN DE HACER CIERTAS FUNCIONES PARA EL MANEJO Y DESARROLLO DE ESTOS.

# A G R A D E C I M I E N T O S



## **AGRADECIMIENTOS**

AL DEPARTAMENTO DE SISTEMAS DEL BANCO MERCANTIL DEL NORTE, S.N.C, ESPECIALMENTE AL ING. ISRAEL HERAS Y AL ING. JOSE HUTCHINSON POR EL APOYO E INTERES BRINDADO, Y A SOFTWARE AG DE MEXICO, POR TODAS LAS FACILIDADES PROPORCIONADAS PARA LA REALIZACION DE LA PRESENTE TESIS.

GRACIAS A NUESTRO ASESOR EL LIC. OSCAR AGUILAR Y AL ING. RAUL MONTEMAYOR POR SUS UTILES COMENTARIOS EN LAS REVISIONES PREPARATORIAS DE ESTE TRABAJO.

## B I B L I O G R A F I A

## **BIBLIOGRAFIA**

### **ADABAS MESSAGES AND CODES MANUAL**

Noviembre, 1983  
SOFTWARE AG of North America, Inc. and  
SOFTWARE AG, Darmstadt, West Germany.

### **ADABAS CONCEPTS AND FACILITIES**

Diciembre, 1983  
SOFTWARE AG of North America, Inc. and  
SOFTWARE AG, Darmstadt, West Germany.

### **COM-LETE INTRODUCTION**

SOFTWARE AG of North America, Inc. and  
SOFTWARE AG, Darmstadt, West Germany.

### **NATURAL version 1.2 REFERENCE MANUAL**

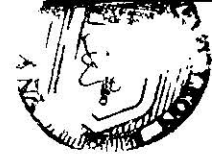
Octubre, 1984  
SOFTWARE AG of North America, Inc. and  
SOFTWARE AG, Darmstadt, West Germany.

### **NATURAL APPLICATIONS WORKSHOP**

Septiembre, 1982  
SOFTWARE AG of North America, Inc. and  
SOFTWARE AG, Darmstadt, West Germany.

### **PREDICT REFERENCE MANUAL version 2.2**

Agosto, 1986 y Abril, 1987  
SOFTWARE AG of North America, Inc. and  
SOFTWARE AG, Darmstadt, West Germany.



UNIVERSIDAD DE  
PACIFICO-MATEMATICAS  
BIBLIOTECA

**BIBLIOGRAFIA**

**GLOSARIO DE COMPUTACION**

Tercera Edicion, 1984  
Alan Freedman  
Mc. Graw Hill

**SISTEMAS DE ADMINISTRACION DE BANCOS DE DATOS**

Cuarta Edicion, 1983  
Alfonso F. Cardenas  
Limusa  
}

**INTRODUCCION AL PROCESAMIENTO DE DATOS PARA LOS NEGOCIOS**

Segunda Edicion, 1983  
Lawrence S. Orillia  
Mc. Graw Hill

