UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

IMPLEMENTACION DE UN SISTEMA DE MANEJO DE BASES DE DATOS (DMS II BURROUGHS)

T E S I S
PARA OBTENER EL TITULO DE
A C T U A R I O

JORGE VIERA HARO

MEXICO, D. F. 1984

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

INDICE PAG.

I INTRODUCCION.

EN LO QUE SIGUE SE TRATARA DE UN PAQUETE DE MANEJO DE BASES DE DATOS, A SABER, DMS II DE BURROUGHS. DADA LA NECESIDAD DE UTILIZAR LAS TECNICAS DE PROCESO DE DATOS QUE MAS EFICIENTES SE MUESTREN, ES IMPERATIVO CONOCERLAS A FONDO, SI NO SE QUIERE COMPROMETER EL EXITO DE LAS APLICACIONES.

EL PROVEEDOR HA DEMOSTRADO EN LA MAYORIA DE SUS PRODUCTOS UN ALTO NIVEL DE INGENIO E INDEPENDENCIA DE LA MANERA CONVENCIONAL, COMO TRATAMOS DE HACER EVIDENTE EN ESTE TRABAJO, PARA EL CASO DE DMS II.

DADA LA DIFICULTAD DE ENTENDER PLENAMENTE UN SISTEMA DE LA COMPLEJIDAD DE DMS II, TAMBIEN ES DIFICIL HACER UN USO EFICIENTE DE SUS POSIBILIDADES. ES PROPOSITO DE ESTE TRABAJO SERVIR COMO INTRODUCCION A DMS II, RESCATANDO LAS IDEAS FUNDAMENTALES DE SU IMPLEMENTACION, Y SENALANDO LAS IMPLICACIONES QUE EN USO DE RECURSOS LAS APLICACIONES CONLLEVAN. ASI, DE ENTRE LAS DISTINTAS ALTERNATIVAS PARA EL DISENO DE UNA APLICACION, EL USUARIO PODRA ESCOGER LAS QUE LE OFREZCAN MAYORES VENTAJAS, Y AUN IMITAR EL DETALLE DEL MANEJO INTERNO EN LAS APLICACIONES QUE SIN DMS II DESARROLLE.

DMS II FUE ANUNCIADO INICIALMENTE EN OCTUBRE DE 1974, HABIENDO SIDO DESARROLLADO PARA REMPLAZAR A DM6700, UN PAQUETE QUE EMPLEABA UN ENFOQUE RELATIVAMENTE CONVENCIONAL. EL PROVEEDOR SENALA QUE CON MENOR USO DE MEMORIA, DMS II SOBRE DM6700 ES MAS EFICIENTE EN OCHO VECES CUANDO MENOS, SIENDO POSIBLE ACCESAR UN REGISTRO DE BASE DE DATOS TAN RAPIDO COMO LO FUESE MEDIANTE UN PROGRAMA DE APLICACION.

ESTO LO BASA DMS II EN EL USO DE RUTINAS REENTRANTES, QUE OPERAN COMO INTRINSECOS DEL SISTEMA MIENTRAS UNA BASE ESTE ABIERTA. TALES RUTINAS SE GENERAN A LA MEDIDA DE LAS ESTRUCTURAS DEFINIDAS Y SE RECONSTITUYEN A CADA CAMBIO DE DEFINICION. ESTO REPRESENTA UN COMPROMISO ENTRE LA INTEGRACION DE LA DEFINICION DE DATOS A LOS PROGRAMAS Y LA INTERPRETACION DE LOS FORMATOS A TIEMPO DE EJECUCION. TAMBIEN FUE MODIFICADO EL SISTEMA OPERATIVO PARA CONTROLAR LA CONCURRENCIA Y LA ASIGNACION DINAMICA DE RECURSOS.

LAS DIFERENTES VISTAS LOGICAS QUE DE LA BASE TIENEN LOS USUARIOS SON RECONOCIDAS POR LOS COMPILADORES DE APLICACION, Y AUN PUEDEN LLEVAR CAMBIOS A LAS RUTINAS REENTRANTES.

LAS RESTRICCIONES DE SEGURIDAD SE BASAN EN EL MANEJO DE CLAVES DE USUARIO ORIGINAL DEL SISTEMA, CON ARCHIVOS GUARDIANES ESPECIFICANDO LOS TIPOS DE ACCESO PERMITIDOS POR PROGRAMA Y USUARIO.

SE POSIBILITA LA REORGANIZACION DE LOS DATOS, YA SEA OBLIGADA POR CAMBIO DE DEFINICION O PARA RECUPERAR ESPACIO. ENTRE OTRAS UTILERIAS SE TIENEN CARGA, RESPALDO, IMPRESION Y CONSULTA DE LA BASE, Y TODAS SE GENERAN A LA MEDIDA DE LAS ESTRUCTURAS DEFINIDAS.

ENTRE LAS ALTERNATIVAS QUE EL USUARIO TIENE PARA EL DISENO DE UNA BASE, ESTA EL USAR ESTRUCTURAS JERAROUICAS. EFECTIVAMENTE ES MAS INTUITIVO ENTENDER UNA ESTRUCTURA INCLUIDA EN OTRA, Y DMS II PERMITE LA DIFINICION DE TAL TIPO DE ESTRUCTURAS. SIN EMBARGO, SU UTILIZACION ES ENGORROSA EN PROGRAMAS DE APLICACION. POR ELLO DESCRIBIMOS Y EJEMPLIFICAMOS UNA TECNICA CON LA CUAL SE CONSIGUEN LOS MISMOS Y AUN MEJORES RESULTADOS, USANDO ESTRUCTURAS SECUENCIALES CON INDICE.

A ESTA INTRODUCCION LE SIGUEN ANTECEDENTES, CON RAZON DE SER, ESTRUCTURA Y MODELOS DE BASES DE DATOS. EL EQUIPO BURROUGHS B6700 ES DESCRITO COMO PERTENECIENTE A LA TERCERA GENERACION.

PARA DESCRIBIR DMS II, LO DESCRIBIMOS DESDE SUS COMPONENTES FUNDAMENTALES, Y DETALLAMOS CADA UNA DE LAS ESTRUCTURAS FISICAS RESULTANTES EN LAS BASES.

SEGUIMOS CON CONSIDERACIONES SOBRE EL DISENO DE BASES DE DATOS, PARA TERMINAR CON DETALLES DE IMPLEMENTACION, CONCLUSIONES Y BIBLIOGRAFIA.

II ANTECEDENTES.

II.1 POR QUE DE BASES DE DATOS.

FRECUENTEMENTE SE SENALAN DOS RAZONES COMO PRINCIPALES RESPONSABLES DEL INCREMENTO EN EL COSTO DE USO DE UN COMPUTADOR. LA PRIMERA ES QUE LOS SISTEMAS DE APLICACION ESTAN SUJETOS A CAMBIOS : EN UN LAPSO CORTO, UNA ORGANIZACION PUEDE ENCONTRARSE CON QUE EL PROBLEMA ATACADO EN UNA APLICACION HA CAMBIADO. ALTERNATIVAMENTE, LAS PREGUNTAS QUE LA GERENCIA HACE PARA SU PLANEACION Y CONTROL PUEDEN CAMBIAR, INDEPENDIENTEMENTE DE CUALQUIER CAMBIO EN LOS PROCESOS BASICOS. ASI, LA ESPERANZA DE VIDA DE UNA APLICACION ES DE TRES A CINCO ANOS. EN ESTE PERIODO, SE HA DADO SUFICIENTE CAMBIO PARA HACER OBSOLETO EL DISENO PREVIO Y NECESARIO EL INVERTIR EN UNO NUEVO. EN OCASIONES EL COSTO DE MANTENIMIENTO Y SOPORTE RESULTA DEL DOBLE DEL DE LOS COSTOS DE DISENO Y DESARROLLO ORIGINALES.

LA SEGUNDA RAZON ES EL COSTO DEL PERSONAL. LAS APLICACIONES CRECEN EN COMPLEJIDAD, LO QUE INCREMENTA LA CANTIDAD DE CONOCIMIENTO QUE DE ELLAS SE HA DE TENER PARA UNA IMPLEMENTACION EFECTIVA. EL PERSONAL DE COMPUTACION COMPETENTE, Y ADEMAS INTERESADO EN LAS APLICACIONES ES ESCASO, Y LA TASA DE ROTACION ES ELEVADA.

TRADICIONALMENTE, EL METODO EMPLEADO EN DESARROLLAR PROGRAMAS DE APLICACION SIGUE LAS SIGUIENTES FASES :

A) RECONOCIMIENTO POR PARTE DEL USUARIO, DE QUE PARA UN PROBLEMA PUEDE USAR LA COMPUTADORA.

B) LO DISCUTE CON UN ANALISTA DE SISTEMAS.

C) EL ANALISTA DESARROLLA EL REQUERIMIENTO DE DATOS. ESTO ES, DETERMINA :

   1) DATOS RELEVANTES.
   2) DIMENSIONES DE ELEMENTOS DE DATOS.
   3) LA REPRESENTACION DE LOS VARIOS ELEMENTOS DE DATOS, TAMANOS DE REGISTROS, ETC.
   4) LA ESTRATEGIA DE ALMACENAMIENTO Y EL COMPROMISO ENTRE ESPACIO A OCUPAR Y TIEMPO DE PROCESO.
   5) LA ESTRATEGIA DE CONTROL DE EJECUCIONES CONTRA FALLAS, Y DE RECUPERACION.

D) EL ANALISTA DE SISTEMAS IDENTIFICA LAS DISTINTAS TRANSACCIONES O UNIDADES DE TRABAJO EN TERMINOS DE PROCESOS LOGICOS DE LOS DATOS.

E) EL ANALISTA DETERMINA LA VALIDACION A EFECTUARSE SOBRE LAS ENTRADAS, Y LA LOGICA DE LAS TRANSACCIONES.

F) SE PASA A CODIFICACION EN UN LENGUAJE DE APLICACION.

G) SE REVISA, PRUEBA Y CORRIGE EL PROGRAMA.

SOLO DESPUES DE TODOS ESTOS PASOS SON VISIBLES AL USUARIO LOS PRIMEROS RESULTADOS. SI EN SU DESARROLLO LOS POSTULADOS DEL PROBLEMA HAN CAMBIADO, SE HAN DE REPETIR LOS PASOS ANTERIORES UNA Y OTRA VEZ, HASTA LOS RESULTADOS DESEADOS.

EL PROBLEMA DEL ALTO COSTO ASOCIADO CON ESTE METODO ES QUE CADA ANALISTA ES RESPONSABLE DE IDENTIFICAR : LOS DATOS RELEVANTES, SU REPRESENTACION, LOS TAMANOS DE LOS ELEMENTOS, ETC. ; LA ESTRATEGIA DE ALMACENAMIENTO Y LECTURA HA SER USADA ; EL COMPROMISO ENTRE LOS COSTOS DE ALMACENAMIENTO Y DE LECTURA, Y LOS MECANISMOS DE CONTROL DE EJECUCION Y DE RECUPERACION DE FALLAS, EN LAS DIFERENTES APLICACIONES. RESULTAN OPTIMAS HASTA LO MEJOR DE SU HABILIDAD, PERO COMO GENERALMENTE TIENE UNA PANORAMICA MUY ESTRECHA DEL PROBLEMA Y DE LA NATURALEZA DE SOLICITUDES POSTERIORES, ESTOS DATOS PUEDEN SER INACCESIBLES PARA CUALQUIER OTRO PROGRAMA, A MENOS QUE ESTEN DISPONIBLES LAS DESCRIPCIONES PRECISAS DE LOS DATOS, DE LOS ALGORITMOS DE LECTURA Y ALMACENAMIENTO, Y DE LOS MECANISMOS DE CONTROL DE PROCESO Y RECUPERACION. POR OTRO LADO, ESTOS ESTAN FRECUENTEMENTE TAN LIGADOS AL PROGRAMA URIGINAL QUE SI SE ADAPTAN PARA SER USADOS POR OTRO PROGRAMA AQUEL HA DE CORREGIRSE. EN MUCHOS CASOS, NO SE DISPONE DE DOCUMENTACION ADECUADA EN RELACION A ESTOS FACTORES, CON LO QUE SE IMPONE UN ESTUDIO PREVIO. AL TERMINAR TAL ESTUDIO, EN ALGUNOS CASOS LA NECESIDAD DE LA INFORMACION YA PASO DE CRITICA A IRRELEVANTE.

ADEMAS, NORMALMENTE EL PROGRAMADOR ESTA INADECUADAMENTE ENTRENADO PARA EVALUAR LAS DIFERENTES ESTRATEGIAS DE PROGRAMACION, Y PARA DESARROLLAR UN SISTEMA DE CONTROL DE PROCESO Y DE RECUPERACION DE FALLAS. ESTO ULTIMO COMPROMETE LA INTEGRIDAD DE LOS DATOS, AL OBLIGAR A REPETIR PROCESOS.

UN TIPICO PROGRAMA DE APLICACION CONSTA DE 6 PARTES DISTINTAS :

A) ALGUNA FORMA DE DESCRIPCION DE LOS DATOS. EN COBOL, SE FORMALIZA EN LA DIVISION DE DATOS.

B) ALGUNA LOGICA RELACIONADA CON VALIDACION DE ENTRADAS, QUE CALIFICA LOS DATOS QUE ENTRAN A PROCESO.

C) LA LOGICA DE PROCESO DE TRANSACCIONES Y DE CALCULOS GENERALES.

D) LAS FUNCIONES DE "OBTIENE" Y "DA", CON DISPOSITIVOS DE ALMACENAMIENTO SECUNDARIO, QUE PONEN DISPONIBLES A LOS DATOS PARA EL PROCESO DE TRANSACCIONES.

E) UNA FASE DE ORGANIZACION Y SALIDA DE DATOS PRODUCIDOS.

F) LA LOGICA DE CONTROL DE PROCESO Y DE RECUPERACION DE PROGRAMA Y DE DATOS.

EN LA MAYORIA DE LAS APLICACIONES COMERCIALES, SUCEDE QUE LA DESCRIPCION DE DATOS, EL SEGUIMIENTO Y RECUPERACION DE TRANSACCIONES Y LAS FUNCIONES DE "OBTIENE" Y "DA" CONSUMEN MAS TIEMPO Y CODIGO QUE LA VALIDACION DE ENTRADAS Y LA LOGICA DE TRANSACIONES. ESTO ES, EL PROGRAMADOR SE OCUPA MENOS DEL PROBLEMA QUE SE INTENTA RESOLVER QUE EN FUNCIONES DE AMBIENTE, ESPECIALMENTE EN MANTENIMIENTO Y CORRECCION.

CON ESTO EN MENTE, SE PROPONE QUE SE PUEDE OBTENER LA REDUCCION DEL COSTO DE MANTENIMIENTO SEPARANDO ESTOS ASPECTOS DE LA PROGRAMACION DE APLICACION.

EL ENFOQUE CORRIENTE CONSISTE EN INTEGRAR LOS DATOS Y LA LOGICA PARA UNA APLICACION ESPECIFICA DE MANERA QUE SE MAXIMICE LA EFICIENCIA INTERNA DEL PROCESO, LO QUE LLEVA A SERIOS PROBLEMAS, COMO LOS SIGUIENTES :

EL EXCESIVO COSTO DE EFECTUAR CAMBIOS EN LA APLICACION, COMO EN LAS CARACTERISTICAS DE LOS DATOS MANEJADOS, QUE SE REQUIERAN OTRAS ENTRADAS O SALIDAS, O CAMBIOS TECNOLOGICOS. COMO EJEMPLOS DE CAMBIOS EN CARACTERISTICAS DE LOS DATOS TENEMOS QUE SE REQUIERAN MAS DIGITOS EN UNA CANTIDAD, QUE UNA CLAVE ANTES NUMERICA INCLUYA CARACTERES ALFABETICOS, O QUE UN DATO SE CONSIDERE DE SOBRA. TAMBIEN, PUEDE SER NECESARIO UN DATO NO CONTEMPLADO EN LA APLICACION ORIGINAL, LO QUE REPRESENTARIA OTRA ENTRADA U OTRA SALIDA. Y POR ULTIMO, LOS CAMBIOS TECNOLOGICOS NO SON SOLO APROVECHABLES SINO AUN OBLIGADOS, POR LOS COSTOS DE MANTENIMIENTO DE LOS EQUIPOS OBSOLETOS, DE UNIDADES DE ALMACENAMIENTO POR EJEMPLO.

DIFICULTAD PARA TRANSFERIR Y HACER ACCESIBLES LOS DATOS ENTRE PROCESOS DIFERENTES.

ALTOS COSTOS DE DISENO Y REDISENO, DADO QUE SE HA DE ESCOGER DE ENTRE LAS DISTINTAS ALTERNATIVAS DE ACOMODO FISICO PARA LA INFORMACION, QUE SIEMPRE NECESITARA REACOMODO.

Y EL QUE LAS MEJORAS Y MODIFICACIONES SE TOMAN DEMASIADO TIEMPO.

DISTINTOS ENFOQUES HAN SIDO APLICADOS A LA IMPLEMENTACION DE UN SERVICIO DE DATOS EN UNA ORGANIZACION MERCANTIL. SE HAN INTENTADO VARIAS TECNICAS DE ORGANIZACION Y DE PROCEDIMIENTOS DE USUARIO, COMO EL ESTABLECER ESTANDARES Y CONVENCIONES EN DOCUMENTACION Y PROGRAMACION, DESARROLLAR MODULOS ESTANDARES DE ACCESO COMUN, PARA SISTEMAS DE ALMACENAMIENTO Y RECUPERACION, QUE PUDIESEN SER COPIADOS POR CADA PROGRAMADOR, Y EL ELABORAR UN MANEJADOR DE ARCHIVOS, CON MANTENIMIENTO Y ORGANIZACION CENTRALIZADOS, Y CONVENCIONES DE USO POR LOS PROGRAMADORES. OTRO ENFOQUE LO CONSTITUYEN LAS BASES DE DATOS.

LAS BASES DE DATOS PROPORCIONAN UN CONTROL CENTRALIZADO DE LA INFORMACION. LA INFORMACION HA DE ADMINISTRARSE COMO UN RECURSO, A MODO DE QUE ESTE DISPONIBLE CON BUENOS MARGENES DE CONFIABILIDAD Y OPORTUNIDAD.

COMO VENTAJAS DEL CONTROL CENTRALIZADO DE LA INFORMACION, SE MENCIONAN :

1) POSIBLE REDUCCION DE LA REDUNDANCIA.

2) CONSISTENCIA DE LA INFORMACION : SE PUEDE UTILIZAR LA REDUNDANCIA PARA DETECTAR Y CORREGIR ERRORES.

3) DISPONIBLIDAD DE LA INFORMACION : POR LA INTEGRACION DE LOS ARCHIVOS DE LAS DIFERENTES APLICACIONES, POR EL ESTABLECIMIENTO EXPLICITO DE LOS DERECHOS DE CONSULTA DE LA INFORMACION A LOS DIFERENTES NIVELES DE USUARIO, Y POR LA OPORTUNA SATISFACCION DE SOLICITUDES NO ANTICIPADAS.

4) ESTANDARIZACION, COMO POR EJEMPLO DE UNIDADES EN LAS CANTIDADES Y EN LOS CODIGOS A UTILIZARSE.

5) SEGURIDAD, AL ESTABLECERSE RESTRICCIONES EN LOS ACCESOS.

6) PROTECCION DE LA INTEGRIDAD DE LA INFORMACION, CON MECANISMOS PRESERVADORES DE LA DESTRUCCION.

UN SISTEMA DE MANEJO DE BASES DE DATOS PUEDE DEFINIRSE COMO UN SISTEMA DE PROGRAMACION PARA EL MANEJO Y MANTENIMIENTO DE DATOS EN UNA ESRUCTURA NO REDUNDANTE, CON EL PROPOSITO DE SER PROCESADOS POR APLICACIONES MULTIPLES. SE PERSIGUE UNA MAYOR EFECTIVIDAD EN EL PROCESO DE LOS DATOS, CONSIDERANDO LAS RELACIONES QUE EXISTAN ENTRE ELLOS, Y LIBERANDO AL PROGRAMADOR DE APLICACION DE CONSIDERACIONES TALES COMO SU LOCALIZACION, REPRESENTACION FISICA Y METODO DE ACCESO.

EL PRIMER CONCEPTO QUE CARACTERIZA LA NATURALEZA DE LOS SISTEMAS DE BASES DE DATOS ES QUE LOS DATOS HAN DE SER GRABADOS UNA SOLA VEZ, PARA SER COMPARTIDOS POR LOS VARIOS SUBSISTEMAS Y USUARIOS DEL SISTEMA. PARA SATISFACER LAS NECESIDADES DE TODOS LOS USUARIOS SIN LA PRESENCIA DE DUPLICACION DE DATOS, ESTOS SE HAN DE ESTRUCTURAR Y ORGANIZAR. LA INFORMACION NO DEBE GRABARSE EN ARCHIVOS LOCALES, DISENADOS PARA APLICACIONES ESPECIFICAS.

EL CONCEPTO DE BASE DE DATOS ES MOTIVADO TAMBIEN CUANDO SE CAPTURAN Y GRABAN DATOS AL MOMENTO QUE SE SUCEDEN EN EL DESEMPENO DE LA ORGANIZACION LOS EVENTOS QUE DAN LUGAR A LAS CORRESPONDIENTES TRANSACCIONES. EN ESTE IDEAL DE PROCESO EN TIEMPO REAL, EL OBJETIVO ES ASEGURAR QUE LOS ARCHIVOS SEAN ACTUALIZADOS A MEDIDA QUE LOS EVENTOS OCURRAN, Y QUE LA COMPUTADORA SEA CAPAZ DE COMUNICAR INFORMACION RELEVANTE EN MUCHOS LUGARES AL MISMO TIEMPO, DEJANDO ATRAS LOS PROBLEMAS DE COMUNICACION DE LAS GRANDES ORGANIZACIONES. EL SISTEMA DE BASE DE DATOS NECESITA ASEGURAR QUE LOS DATOS SON EXACTOS Y ESTEN DISPONIBLES, SIN DESCUIDAR LA SEGURIDAD Y CONFIDENCIALIDAD DE LA INFORMACION.

UN SISTEMA DE MANEJO DE DATOS, SEA O NO GENERALIZADO A BASES DE DATOS, EN PRINCIPIO RELEVA DE RESPONSABILIDAD SOBRE LAS FUNCIONES DE "OBTIENE" Y "DA" DE Y A ESTRUCTURAS DE ARCHIVOS, VALIDACION DE DATOS DE ENTRADA, Y PROCESOS DE TRAZA DE MODIFICACIONES Y DE RECUPERACION. EN SEGUNDO LUGAR, PRESENTA LA OPORTUNIDAD DE DEFINIR ESTRUCTURAS MAS VERSATILES QUE LAS DE LOS ARCHIVOS TRADICIONALES, CON RELACIONES Y NIVELES DE DATOS, CUYO MANTENIMIENTO SE HACE TRANSPARENTE AL USUARIO.

UN SISTEMA GENERALIZADO DE MANEJO DE BASES DE DATOS HA DE INCORPORAR LA MAYORIA DE LAS SIGUIENTES FACILIDADES :

INDEPENDENCIA ENTRE LOS LENGUAJES DE APLICACION Y DE CONTROL.

SOPORTE DE LOS LENGUAJES DE APLICACION USADOS CON ANTERIORIDAD EN LA INSTALACION.

PROGRAMAS DE UTILERIA PARA FACILITAR LA CREACION Y EL MANTENIMIENTO DE LAS BASES.

POSIBILIDAD DE REORGANIZACION DE LOS DATOS.

SEGURIDAD DE LOS DATOS Y LIMITACION DE ACCESO.

REARRANQUE AUTOMATICO EN CASO DE FALLA DEL SISTEMA, O RECUPERACION MANUAL DE LAS APLICACIONES CON ESFUERZO MINIMO.

POSIBILIDAD EN PRIMER LUGAR, DE ESCOGER ENTRE LAS ALTERNATIVAS DE USO DE RECURSOS, POR EJEMPLO EL ASIGNAR MAS MEMORIA DE EJECUCION PARA MEJORAR EL TIEMPO DE RESPUESTA, Y LA POSIBILIDAD DE AJUSTE FINO, COMO EL DETERMINAR LA CANTIDAD DE MEMORIA DE QUE SE PODRA DISPONER.

ADEMAS, SE PUEDE CONTAR CON CAPACIDAD DE PROCESO DE TRANSACCIONES, CON UN SUBSISTEMA DE CONSULTA, Y CON UN GENERADOR DE REPORTES.

II.2 ESTRUCTURA DE UNA BASE DE DATOS.

SE PUEDEN DISTINGUIR TRES NIVELES : INTERNO, EXTERNO Y CONCEPTUAL. EN EL NIVEL INTERNO SE ENCUENTRAN LAS ESTRUCTURAS DE ALMACENAMIENTO BASICO, MIENTRAS QUE EN EL EXTERNO SE ENCUENTRAN LAS DIVERSAS IMAGENES QUE DE LA BASE DE DATOS TIENEN LOS USUARIOS PARA LA SATISFACCION DE SUS SOLICITUDES. LA IMAGEN GLOBAL DE LA BASE DE DATOS INTEGRA EN EL NIVEL CONCEPTUAL LAS IMAGENES A LOS USUARIOS CON LOS PROCEDIMIENTOS DE AUTORIZACION Y VALIDACION, EN LO QUE TIENDE A SER LA DESCRIPCION DE LA ORGANIZACION EN CUANTO AL FLUJO INTERNO DE LOS DATOS, LOS DIFERENTES USOS DE LA INFORMACION Y SU CONTROL.

II.3 MODELOS DE BASES DE DATOS.

LAS DISTINTAS REPRESENTACIONES DE ESTRUCTURAS ENTRE LOS DATOS QUE CONSTITUIRIAN LOS ARCHIVOS TRADICIONALES, EN EL CONTEXTO DE BASES DE DATOS HAN DADO ORIGEN A DIVERSOS MODELOS, DE ENTRE LOS CUALES LOS MAS REPRESENTATIVOS SON EL JERARQUICO, EL DE REDES Y EL RELACIONAL. EN ESTA DISCUSION, UNA ENTIDAD REPRESENTARA INFORMACION, Y UNA RELACION SERA UNA CORRESPONDENCIA LOGICA ENTRE ENTIDADES.

MODELO JERARQUICO.

UN MODELO JERARQUICO ESTA CONSTITUIDO POR UN CONJUNTO DE ENTIDADES Y POR UNO DE RELACIONES, QUE ESTABLECE UNA JERARQUIA ENTRE LAS ENTIDADES. NO PUEDE HABER MAS DE UNA RELACION ENTRE CUALESQUIERA DOS ENTIDADES ; LAS RELACIONES SON DE UNA ENTIDAD DE MAS JERARQUIA A MUCHAS DE MENOR, Y POR CADA ENTIDAD EXISTE UNA Y SOLO UNA ENTIDAD EN EL NIVEL INMEDIATO SUPERIOR, EXCEPTUANDO LA DE MAYOR JERARQUIA, LLAMADA RAIZ.

SE PRESENTA UNA ESTRUCTURA DE ARBOL EN LOS DATOS, QUE ES UTILIZADA PARA LA RECUPERACION DE LA INFORMACION, PASANDO DE LAS ENTIDADES PADRES A LAS ENTIDADES HIJOS, DE ESTAS A LAS HIJOS DE ESTAS, ETC., HASTA QUE NO QUEDAN ENTIDADES DE MENOR JERARQUIA, O BIEN SE RECUPERAN LAS ENTIDADES DE ACUERDO A VALORES DESEADOS DE LOS ELEMENTOS DE DATOS.

MODELO DE REDES.

EN ESTE MODELO SE TIENEN UN CONJUNTO DE ENTIDADES Y OTRO DE RELACIONES, ESTABLECIENDO PERTENENCIAS ENTRE LAS ENTIDADES.

TIPICAMENTE, DADA UNA ENTIDAD PROPIETARIO ES POSIBLE PROCESAR TODAS LAS ENTIDADES MIEMBRO ; DADO UN MIEMBRO, ES POSIBLE PROCESAR LA ENTIDAD PROPIETARIO, Y DADO UN MIEMBRO, ES POSIBLE PROCESAR OTROS MIEMBROS DEL MISMO PROPIETARIO. EL PROCESO PUEDE EFECTUARSE SEGUN EL ORDEN DEL VALOR DE ALGUNO DE LOS ELEMENTOS DE DATOS, O EN ALGUN OTRO ORDEN, COMO EL DE PRIMERAS ENTRADAS PRIMERAS SALIDAS, O ULTIMAS ENTRADAS PRIMERAS SALIDAS.

SE PRESENTAN DOS RESTRICCIONES PARA UNA RELACION DADA : EL QUE UNA ENTIDAD NO PUEDE SER MIEMBRO O PROPIETARIO DE SI MISMA, Y EL QUE UN PROPIETARIO NO PUEDE TENER COMO MIEMBRO DOS VECES A UNA MISMA ENTIDAD.

MODELO RELACIONAL.

EN ESTE MODELO SE PRESENTA AL USUARIO SOLAMENTE INFORMACION ORIGINAL Y COMO SE RELACIONAN ENTRE SI SUS DATOS, SIN PRESENTAR ESTRUCTURAS, DE MODO QUE SE TIENEN LAS ENTIDADES ORIGINALES Y ENTIDADES RELACION, LAS QUE TIENEN DATOS DE LAS ENTIDADES QUE RELACIONAN. LAS ENTIDADES RELACION PUEDEN ESTAR DEFINIDAS PARA TANTAS ENTIDADES COMO SE DESEE.

SE TIENE CON ESTE MODELO INMUNIDAD DE LAS APLICACIONES ANTE CAMBIOS EN LA ESTRUCTURA FISICA DE ALMACENAMIENTO DE LA INFORMACION Y EN LA ESTRATEGIA DE ACCESO ; ES MAS SIMPLE DE MANEJAR QUE LOS MODELOS JERARQUICO Y DE RED, Y ALGUNAS FORMAS DE RECUPERACION DE LA INFORMACION SON MAS FACILES. ESTO PARA EL USUARIO, PUES LA IMPLEMENTACION PARA USO COMERCIAL RESULTA COMPLEJA, LO QUE SE TRADUCE EN ALTOS COSTOS DE OPERACION. SE HA DE SOPORTAR EN CIRCUITOS Y NO EN PROGRAMACION.

III AMBIENTE.

III.1 CARACTERISTICAS COMUNES A LOS SISTEMAS OPERATIVOS DE TERCERA GENERACION.

LA CONCURRENCIA, O EXISTENCIA POSIBLE DE VARIOS PROCESOS PARALELOS.
LA ASIGNACION AUTOMATICA DE RECURSOS.

EL COMPARTIR RECURSOS, SIENDO USADOS SIMULTANEAMENTE POR MAS DE UN PROCESO, SEA UN TIPO DE RECURSO O INFORMACION.

EL MULTIPLEXADO, O LA DIVISION DEL TIEMPO EN INTERVALOS DISJUNTOS Y LA ASIGNACION DE UNA UNIDAD DE TIEMPO A LO MAS A UN PROCESO DURANTE CADA INTERVALO. CON EL MULTIPLEXADO ENCONTRAMOS MULTIPROGRAMACION, CON VARIOS PROGRAMAS O PARTES DE ELLOS EN MEMORIA EN UN MOMENTO DADO ; MULTIACCESO, CUANDO AL SISTEMA O A PARTE DE EL MUCHOS USUARIOS TIENEN ACCESO SIMULTANEAMENTE, COMO UN SISTEMA DE TIEMPO COMPARTIDO ; MULTITAREA, CUANDO SE SOPORTA MAS DE UN PROCESO ACTIVO, Y MULTIPROCESO, CUANDO SE TIENEN VARIOS PROCESADORES CENTRALES, Y ESTAN ACTIVOS Y CONCURRENTES VARIOS PROCESOS.

EL ACCESO CONVERSACIONAL REMOTO, CUANDO MUCHOS USUARIOS INTERACTUAN EN LINEA CON SUS PROCESOS.

LA INDETERMINACION, O CALIDAD DE IMPREDECIBLE DEL ORDEN EN QUE OCURRIRAN CUALES EVENTOS.

Y EL ALMACENAMIENTO INDEFINIDO, DE INFORMACION DE USUARIOS. A ESTA ULTIMA CARACTERISTICA SE DEBE LA NECESIDAD DE CONTAR CON UN SISTEMA DE ARCHIVO, CON GARANTIAS EN CUANTO A LA SEGURIDAD DE LA INFORMACION CONTRA FALLAS DEL SISTEMA O ERRORES DEL USUARIO, Y CON CONTROL DE ACCESO, CONTRA LECTURA O ESCRITURA SIN AUTORIZACION.

EL DESARROLLO DE SISTEMAS DE ARCHIVOS PASO POR UNA ERA DE SISTEMAS PRE-ARCHIVOS, EN LA CUAL NINGUNA INFORMACION ERA ALMACENADA PERMANENTEMENTE ; UNA ERA DE SISTEMAS TEMPRANOS DE ARCHIVOS, EN LA QUE LA CAPACIDAD DE ALMACENAMIENTO ESTABA LIMITADA A LA CINTA DEL SISTEMA ; ERA DE SISTEMAS DE BIBLIOTECA, CON CINTAS DE INFORMACION Y UN INDICE, Y LA ERA DE SISTEMAS GENERALIZADOS DE ARCHIVO.

ENTRE LAS CARACTERISTICAS DESEABLES EN UN SISTEMA DE ARCHIVO, SE RECONOCEN EL QUE LAS OPERACIONES DE ENTRADA Y SALIDA SE REDUZCAN EN LO POSIBLE EN NUMERO, EL QUE SE CUENTE CON FLEXIBILIDAD IRRESTRICTA ENTRE LOS TAMANOS DE REGISTRO LOGICO Y DE BLOQUE FISICO, EL QUE EL ACOMODO DE ESPACIO DE ARCHIVO SEA AUTOMATICO, EL QUE SEA DINAMICO, Y QUE LA NOMENCLATURA DE LOS ARCHIVOS SEA FLEXIBLE.

III.2 BURROUGHS B6700.

ESTE EQUIPO INCLUYE MODULOS DE MEMORIA CENTRAL, USUALMENTE DE 65,536 PALABRAS, CADA UNA CON 48 DIGITOS BINARIOS ACCESIBLES, MAS TRES DE TIPO DE PALABRA. PUEDE INCLUIR HASTA TRES PROCESADORES CENTRALES Y HASTA TRES DE ENTRADA-SALIDA. LA INFORMACION SE TRANSFIERE ENTRE LA MEMORIA Y LOS PROCESADORES, MEDIANTE UNOS CABLES DENOMINADOS "BUS DE MEMORIA". SE MANDAN INSTRUCCIONES DE CONTROL ENTRE LOS DISTINTOS TIPOS DE PROCESADOR POR LOS CABLES QUE CONSTITUYEN EL "BUS DE BUSQUEDA". SI HAY SOLO UN PROCESADOR CENTRAL Y UNO DE ENTRADA-SALIDA, SOLO UNO PUEDE TOMAR O GUARDAR DE MEMORIA A LA VEZ.

LOS PERIFERICOS NO ESTAN CONECTADOS DIRECTAMENTE EN EL PROCESADOR DE ENTRADA-SALIDA, SINO A UN CONTROLADOR ESPECIFICO PARA CADA TIPO DE PERIFERICO, QUE SE CONECTA A SU VEZ AL "BUS DE CONTROLADORES".

LOS PERIFERICOS LENTOS, COMO PARA TARJETAS, CINTAS DE PAPEL E IMPRESORAS, REQUIEREN CADA UNO UN CONTROLADOR. SE TIENE SOLO UNA RUTA A UNA UNIDAD EN ESTA CATEGORIA, Y CADA PERIFERICO TIENE UN NUMERO ASIGNADO ENTRE 0 Y 255, QUE DA UN MAXIMO DE 256 PERIFERICOS. LOS PERIFERICOS RAPIDOS, COMO DISCOS FIJOS, PAQUETES DE DISCOS Y CINTAS, PUEDEN TENER MAS DE UNA RUTA USANDO UN "INTERCAMBIADOR".

ENTRE LOS PERIFERICOS QUE SE INCLUYEN TENEMOS LECTORA DE TARJETAS, PERFORADORA DE TARJETAS, UNIDADES DE CINTA MAGNETICA, LECTORA DE CINTA DE PAPEL, PERFORADORA DE CINTA DE PAPEL, IMPRESORAS, PAQUETES DE DISCOS, UNIDADES DE ALMACENAMIENTO EN DISCO, CONSOLAS, Y TERMINALES.

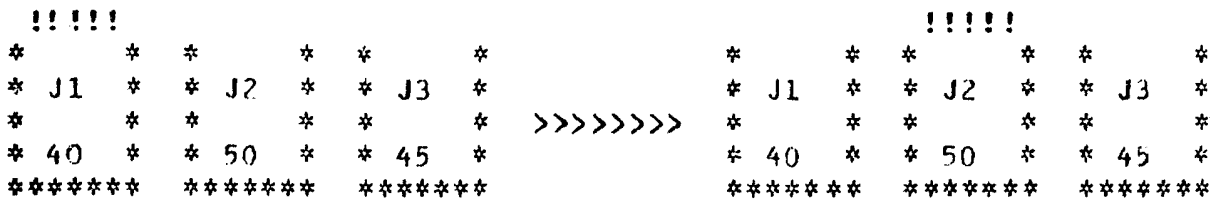
III.2.1 OPERACION DEL SISTEMA B6700.

CONTROL DE TAREAS Y TRABAJOS.

LOS TRABAJOS SE CATEGORIZAN DEPENDIENDO SOLO DE SUS CARACTERISTICAS LOGICAS O CLASE DE SERVICIO REQUERIDO, SEGUN SEAN PEQUENOS, GRANDES, MAS GRANDES, CON TERMINO MUY RAPIDO, CON O SIN INTERFERENCIA DEL OPERADOR, ETC. CONSTITUYENDOSE COLAS, DE DONDE PASAN A EJECUCION Y A SOLICITAR RECURSOS AL SISTEMA OPERATIVO. SE PUEDEN ESTABLECER NIVELES DE MULTIPROGRAMACION PARA BUSCAR LA EJECUCION MAS PRONTA DE MAS TRABAJOS, COMO LO SON LOS LIMITES DE TRABAJOS EN EJECUCION DE CADA UNA DE LAS COLAS. OTROS ATRIBUTOS DE COLAS SON LA PRIORIDAD DEFAULT Y EL LIMITE DE PRIORIDAD, LOS LIMITES MAXIMOS DE TIEMPOS TOTALES DE ENTRADA/SALIDA, DE TIEMPO DE PROCESO, DE LINEAS IMPRESAS Y DE TARJETAS PERFORADAS ; EL LAPSO A TRANSCURRIR ANTES DE SELECCIONAR PARA EJECUCION OTRO TRABAJO DE LA MISMA COLA, Y EL TIPO DE ACOMODO EN MEMORIA DE INTERCAMBIO.

LA MULTIPROGRAMACION ES IMPLEMENTADA POR AMBAS PROGRAMACION Y CIRCUITERIA. SE TIENEN VARIOS PROGRAMAS AVANZANDO AL MISMO TIEMPO, HACIENDO USO DE DIFERENTES RECURSOS A LA VEZ. POR PROGRAMACION, EL MCP MANTIENE TABLAS CONCERNIENTES A LOS DIFERENTES TRABAJOS A CADA MOMENTO.

POR EJEMPLO, TOMEMOS LOS TRABAJOS J1(40), J2(50) Y J3(45) COMO LISTOS PARA CORRER. UN PROCESADOR ESTA EJECUTANDO EN EL STACK PARA J1 Y CUANDO J1 SOLICITA UNA OPERACION DE ENTRADA-SALIDA POR LA CUAL HA DE ESPERAR, DEJARA EL CONTROL Y SE TRABAJARA EN EL STACK PARA J2. CUANDO SE COMPLETE LA ENTRADA-SALIDA PARA J1, J1 SE PONDRA EN LA COLA DE TRABAJOS LISTOS.



EN EL CASO DEL MULTIPROCESO, SE TRATA A LOS PROCESADORES SOLAMENTE COMO RECURSOS, SIN PRECEDENCIA ENTRE ELLOS. EL SISTEMA OPERATIVO TRANSFERIRA CONTROL A CUALQUIERA DE ELLOS SI UN STACK ESTA LISTO. PERO TAMBIEN PODRA DECIDIR QUE UNO CORRA EN ESTADO DE CONTROL, SIN PERMITIR LAS INTERRUPCIONES, POR EJEMPLO AL INICIAR UNA OPERACION DE ENTRADA- SALIDA, O AL DARLE CONTROL A OTRO STACK.

PARA LA IMPLEMENTACION DE MEMORIA VIRTUAL, EL CODIGO Y LOS DATOS SE DIVIDEN EN SEGMENTOS DE LONGITUD VARIABLE. CADA SEGMENTO SE ACOMODA EN MEMORIA REAL SI SE PUEDE ENCONTRAR UN ESPACIO SUFICIENTEMENTE GRANDE.

UN PROGRAMA TRABAJANDO TENDRA UN DESCRIPTOR EN SU AREA DE TRABAJO O STACK, APUNTANDO A CADA SEGMENTO DE CODIGO O DE DATOS, EL CUAL SE MARCARA SEGUN EL SEGMENTO ESTE PRESENTE O AUSENTE. SI ESTA AUSENTE, SE TRAERA DE DISCO ; SI NO HAY ESPACIO EN MEMORIA SE HARA DISPONIBLE UN AREA MANDANDO EL SEGMENTO A DISCO Y MARCANDO EL DESCRIPTOR CORRESPONDIENTE COMO AUSENTE. EL CODIGO Y LOS DATOS DE SOLO LECTURA NO SON MODIFICABLES Y SE ENCUENTRAN EN EL ARCHIVO DE CODIGO, POR LO QUE NO ES NECESARIO ESCRIBIRLOS DE NUEVO.

EN UN AMBIENTE DE COMUNICACION DE DATOS, ES FRECUENTE EL QUE UN PROGRAMA NO SE NECESITE EN MEMORIA EN LAPSOS CONSIDERABLES, POR EJEMPLO AL ESPERAR RESPUESTA DE UNA TERMINAL. PARA APROVECHAR ESTE HECHO, SE TIENE EL MODO INTERCAMBIO, CON UN AREA FIJA EN DISCO, A DONDE Y DE DONDE SERA MOVIDO EL PROGRAMA, SEGUN UN ATRIBUTO DE LA TAREA O TRABAJO, LLAMADO DE SUBESPACIOS : CON VALOR DE 0, NO SE USA INTERCAMBIO. CON 1, EL CODIGO NO ENTRA AL INTERCAMBIO. CON 2, EN EL INTERCAMBIO SE INCLUYEN LOS DATOS Y EL STACK, Y TAMBIEN EL CODIGO SI PERTENECE A LA BIBLIOTECA DEL PROPIO USUARIO, Y CON VALOR DE 3, TODAS LAS AREAS SE TRABAJAN EN ESPACIO DE INTERCAMBIO.

SEGURIDAD DEL SISTEMA.

CADA TRABAJO DEBE DE TENER ASOCIADA UNA CLAVE DE USUARIO, Y SE SUMINISTRA UNA CONTRASENA, QUE SE MANTIENE SECRETA A LOS DEMAS USUARIOS. LOS ARCHIVOS EN EL SISTEMA PERTENECEN A UNA CLAVE DE USUARIO, Y PUEDEN SER PRIVADOS, CON ACCESO SOLO AL PROPIO USUARIO O A CLAVES PRIVILEGIADAS, DE CLASE A, CON ACCESO A OTROS USUARIOS SUJETA A RESTRICCIONES DE ENTRADA/SALIDA, O DE CLASE B, CON ACCESO CONTROLADO CON UN ARCHIVO GUARDIAN. EL ACCESO A UN ARCHIVO PUEDE RESTRINGIRSE A SOLO LECTURA, SOLO ESCRITURA, O PROHIBIRSE LECTURA Y ESCRITURA.

CONTROL DE ARCHIVOS.

EN UN MOMENTO DADO, SOLO ESTARAN EN LINEA CIERTOS PAQUETES Y CINTAS. EL CONTROL DE ARCHIVOS CONSISTE EN SU ACOMODO EN PAQUETES, CINTAS Y UNIDADES DE DISCO, Y EN EL MANTENIMIENTO DE BIBLIOTECAS, COPIANDO ARCHIVOS DE UN MEDIO DE ALMACENAMIENTO A OTRO. SE USA EL CONCEPTO DE FAMILIA, COMO AGREGADO DE ALMACENAMIENTO MASIVO. SE TIENEN FAMILIAS DE PAQUETES, DE DISCOS CON CABEZA DE LECTURA/ESCRITURA POR PISTA, Y DE CINTAS. SE PUEDE ASIGNAR UNA FAMILIA A UNA COLA DE TRABAJOS DE MODO DE QUE SOLO ELLA SEA NECESARIA EN LINEA. LOS ARCHIVOS EN DIFERENTES FAMILIAS PUEDEN TENER EL MISMO NOMBRE. NO SE PERMITE DUPLICAR NOMBRES EN LA MISMA FAMILIA. PARA NOMBRE DE UN ARCHIVO SE PUEDEN USAR HASTA 14 IDENTIFICADORES, CADA UNO HASTA DE 17 CARACTERES, SEPARADOS POR DIAGONALES. SE PRESENTA ASI PARA LOS ARCHIVOS UNA ESTRUCTURA DE ARBOL, CON LOS IDENTIFICADORES DE SUS NOMBRES. PRESENTAREMOS MAS ADELANTE UNA ESTRUCTURA ALTERNA, CON AHORRO DE ESPACIO Y OTRAS VENTAJAS, CUMPLIENDO DE IGUAL MANERA CON LA REPRESENTACION.

ACOMODO DE ESPACIO EN DISCO.

LOS DISCOS Y PAQUETES ESTAN FISICAMENTE DIVIDIDOS EN SEGMENTOS Y SECTORES DE 180 CARACTERES, DE DONDE LA CIRCUITERIA LEE Y ESCRIBE.

DADO UN ARCHIVO FISICO, SERA DIVIDIDO POR EL SISTEMA OPERATIVO BAJO CONTROL DE PROGRAMA EN FILAS O AREAS. EN DISCO SE TIENE UN ENCABEZADO DE ARCHIVO, EL CUAL ES MANTENIDO POR EL SISTEMA Y APUNTADO POR EL DIRECTORIO DE ARCHIVOS EN DISCO, QUE CONTIENE UN APUNTADOR A CADA FILA DEDICADA AL ARCHIVO. CADA FILA DE ARCHIVO ES UN AREA SEPARADA. GENERALMENTE EL SISTEMA DISTRIBUYE LAS FILAS ENTRE TODOS LOS PAQUETES DE LA FAMILIA. DENTRO DE UN VOLUMEN, SE DAN LAS AREAS EN EL ESPACIO MINIMO DISPONIBLE SUFICIENTE PARA CONTENERLAS. LAS TABLAS DE ESPACIO SE MENTIENEN EN MEMORIA O EN MEMORIA VIRTUAL. ASI, LAS LOCALIDADES REALES DE LOS ARCHIVOS NO SON CONOCIDAS NI REQUERIDAS POR EL USUARIO.

LOS ARCHIVOS DE UN PAQUETE PUEDEN ACOMODARSE EN MODO CILINDRO : CADA FILA DEL ARCHIVO ES CONTENIDA DENTRO DE LIMITES DE CILINDRO, DE FORMA QUE SI LA FILA ES MENOR QUE UN CILINDRO PUEDE SER LEIDA SIN MOVIMIENTO DEL BRAZO.

ESTRUCTURA DE PROGRAMA.

EL CODIGO ESTA DIVIDIDO EN SEGMENTOS PARA FACILITAR LA IMPLEMENTACION DE MEMORIA VIRTUAL.

EL ARCHIVO DE CODIGO INCLUYE : SEGMENTO CERO, SEGMENTOS DE CODIGO, INFORMACION OPCIONAL DE LIGA, INFORMACION OPCIONAL DE NUMEROS DE LINEA, Y DICCIONARIO DE SEGMENTOS.

EL SEGMENTO CERO CONTIENE INFORMACION PARA EL SISTEMA OPERATIVO, CANTIDAD ESTIMADA DE MEMORIA A USAR, NIVEL Y TIPO DEL COMPILADOR. EL DICCIONARIO DE SEGMENTOS CONTIENE APUNTADORES AL PRINCIPIO DE CADA SEGMENTO DE CODIGO, LAS LONGITUDES DE LOS SEGMENTOS, Y UN APUNTADOR AL PRINCIPIO DEL CODIGO DE INICIALIZACION. LA INFORMACION DE LIGA ES PARA SI SE HA DE UNIR A OTRO CODIGO. LA INFORMACION DE LINEAS ES UNA TABLA DE POSICIONES RELATIVAS DE CODIGOS FUENTE Y DE MAQUINA, PARA RELACIONAR MAS FACILMENTE LOS PUNTOS DE ABORTO CON UN LISTADO DE COMPILACION.

LA INICIALIZACION DE PROGRAMA ES COMO SIGUE : EL SISTEMA LEE EL SEGMENTO CERO, ENCUENTRA LA ESTIMACION DE MEMORIA, ETC. SI HAY MEMORIA DISPONIBLE, EL PROGRAMA SERA ARRANCADO ; SI NO SERA PUESTO EN TURNO.

PARA ARRANCAR UNA TAREA, SE CONSTRUYE UN STACK DENOMINADO D1,Y SE LE COPIA EL DICCIONARIO DE SEGMENTOS. CADA ENTRADA ES UN DESCRIPTOR, QUE INICIALMENTE APUNTARA A CODIGO AUSENTE. LOS SEGMENTOS DE CODIGO SERAN LLAMADOS DE DISCO CUANDO SEAN NECESARIOS.

SE ENCUENTRA UN AREA DE MEMORIA CONTIGUA, O STACK D2, QUE SIRVE COMO ALMACENAMIENTO DE DATOS PARA ESTE PROGRAMA, COMO AREA DE TRABAJO PARA RESULTADOS TEMPORALES DE PROCESO, COMO ALMACENAMIENTO DE HISTORIA DEL PROGRAMA, COMO QUE RUTINAS LLAMARON A CUALES OTRAS, Y COMO ALMACENAMIENTO DE ATRIBUTOS DE LA TAREA, COMO LA ESTIMACION DE MEMORIA.

EL CODIGO DE INICIALIZACION CONSTRUIRA EL RESTO DEL STACK.

EN EL STACK D2 SE TENDRAN DATOS COMO : PALABRAS DE CALCULOS, ALGUNAS AREAS PEQUENAS DE REGISTROS, Y LOS ELEMENTOS ESTATICOS, QUE SE TENDRAN AQUI PARA APROVECHAR EL QUE ESTA AREA NO SE PASA A MEMORIA VIRTUAL.

FUERA DEL STACK D2 SE TENDRAN DATOS APUNTADOS POR DESCRIPTORES DE DATOS O DE CADENAS EN EL STACK. ESTOS DESCRIPTORES TIENEN DIRECCION, LONGITUD Y MARCA DE PRESENCIA DE SUS DATOS O CADENAS.

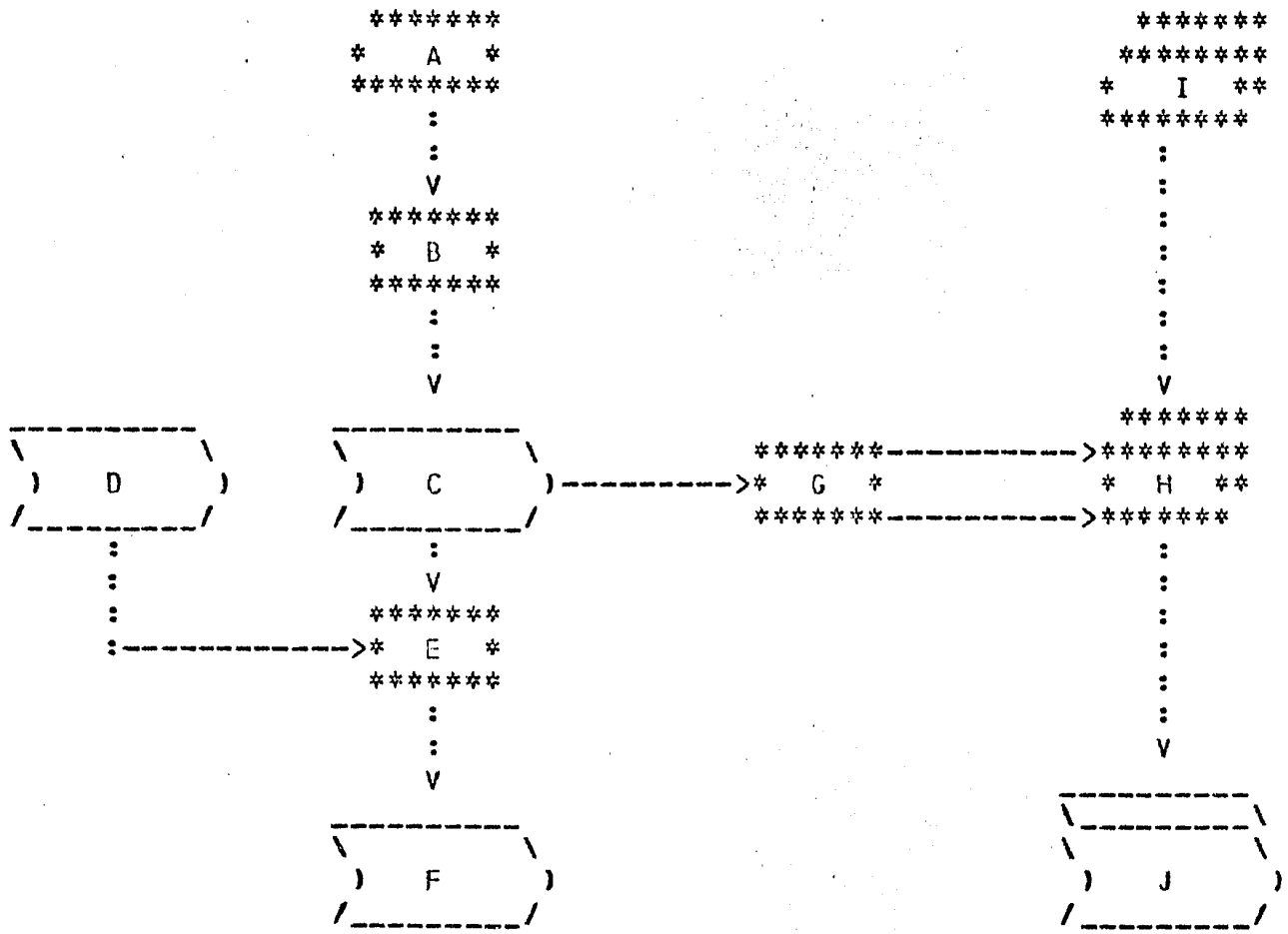
ENTRE OTRAS ESTRUCTURAS SE TIENEN TABLAS MULTIDIMENSIONALES. EN UNA TABLA DE DOS DIMENSIONES, EL DESCRIPTOR EN EL STACK APUNTA A OTRA LISTA DE DESCRIPTORES, LOS CUALES A SU VEZ APUNTAN A LOS RENGLONES DE LA TABLA.

IV DMS II.

IV.1 DESCRIPCION DE DMS II.

ESTE ES UN SISTEMA DE MANEJO DE BASES DE DATOS QUE OPERA EN COMPUTADORAS BURROUGHS 86700, 6800, 7700, Y 7800, BAJO EL SISTEMA OPERATIVO MCP. DA ACCESO COMUN A LA BASE DE DATOS A PROGRAMAS OPERANDO EN LOTEO, PROCESO DE TRANSACCIONES, ENTRADA REMOTA DE TRABAJOS Y DE TIEMPO COMPARTIDO.

COMPONENTES DE DMS II.



A. UN FUENTE EN LENGUAJE DE DEFINICION DE DATOS Y ESTRUCTURAS. ES RESPONSABILIDAD DEL ADMINISTRADOR. ES TOMADO POR

B. EL COMPILADOR DEL LENGUAJE DE DEFINICION, QUE CREA UN

C. ARCHIVO DE DESCRIPCION EN DISCO, CONTENIENDO TODAS LAS CARACTERISTICAS ESTRUCTURALES DE LA BASE DE DATOS.

D. EL SIMBOLICO DE BASE DE DATOS ES UN GRUPO DE PROCEDIMIENTOS Y POSTULADOS EN DMALGOL, QUE SERA TOMADO JUNTO CON EL ARCHIVO DESCRIPTOR POR

E. EL COMPILADOR DE DMALGOL QUE GENERARA

F. RUTINAS DE ACCESO ESPECIALES PARA CADA ESTRUCTURA, QUE SERAN CARGADAS A TIEMPO DE APERTURA DE LA BASE DE DATOS.

EL ARCHIVO DESCRIPTOR TAMBIEN ES CONSULTADO POR

G. LA INTERFASE DE BASE DE DATOS, QUE ACTUA ENTRE LOS COMPILADORES

H. DE COBOL O ALGOL, PARA CHECAR QUE EN EL

I. FUENTE, RESPONSABILIDAD DEL PROGRAMADOR DE APLICACION, LOS NOMBRES CITADOS PARA LOS ITEMS SEAN CONOCIDOS POR EL SISTEMA, Y QUE LAS OPERACIONES REALIZADAS EN ELLOS NO ESTEN EN CONFLICTO CON SUS PROPIEDADES.

J. EL CODIGO DE TIEMPO DE EJECUCION PRODUCIDO POR LOS COMPILADORES CONTIENE REFERENCIAS A LOS PROCEDIMIENTOS DEL SIMBOLICO DE BASE DE DATOS.

IV.2 OPERACION DE DMS II.

PARA OPERACION DEL SISTEMA SE TIENEN DOS COMPONENTES FUNDAMENTALES : UN LENGUAJE DE DEFINICION DE DATOS Y ESTRUCTURAS Y UNA INTERFASE DESDE LENGUAJES DE APLICACION PARA EL USO DE LA BASE. TAMBIEN SE INCLUYEN UN CONJUNTO DE UTILERIAS PARA TRAZA DE MODIFICACIONES Y DE RECUPERACION DE TRANSACCIONES FALLIDAS, Y UTILERIAS PARA LA CARGA, VACIADO Y REORGANIZACION DE LA BASE DE DATOS.

IV.2.1 DEFINICION DE DATOS Y ESTRUCTURAS.

DASDL ES LA HERRAMIENTA USADA POR EL ADMINISTRADOR DE BASE DE DATOS PARA DESCRIBIR LAS BASES AL SISTEMA.

EL DISENO TOTAL DE LA BASE DE DATOS ES RESPONSABILIDAD DEL ADMINISTRADOR. ESTE HA DE ENTENDER LOS REQUERIMIENTOS DE TODOS LOS USUARIOS DE LA BASE, ANALIZAR LOS DIFERENTES USOS, Y PRODUCIR UNA DESCRIPCION DE LOS DATOS CAPAZ DE CUMPLIR CON TODAS LAS NECESIDADES.

DEBE TAMBIEN DETERMINAR CUALES REQUIEREN OPTIMIZACION PARA AUMENTAR LA EFICIENCIA TOTAL. DADO QUE DASDL DA FLEXIBILIDAD Y MUCHAS SOLUCIONES ALTERNAS A CUALQUIER PROBLEMA, EL ADMINISTRADOR PUEDE OPTIMIZAR ALGUNOS USOS DE LA BASE, REDUCIENDO RAPIDEZ DE RESPUESTA EN OTRAS APLICACIONES, O INCREMENTANDO LOS REQUERIMIENTOS DE ALMACENAMIENTO SECUNDARIO. DEBE PUES FAMILIARIZARSE HASTA EL EXTREMO CON DASDL, A FIN DE PRODUCIR UN DISENO QUE NO SOLO SOPORTE LAS SOLICITUDES DE LOS USUARIOS, SINO AUN OPTIMICE EL USO DE LOS DISTINTOS RECURSOS.

A PARTIR DE LAS ESPECIFICACIONES DEL USUARIO EN LENGUAJE DE DATOS Y ESTRUCTURAS SE CREA UN ARCHIVO DESCRIPTOR EN DISCO, CON LA DESCRIPCION COMPLETA DE LAS CARACTERISTICAS ESTRUCTURALES DE LA BASE DE DATOS, QUE ES ACCESADO POR EL COMPILADOR DE APLICACION, QUE INSERTA UNA DESCRIPCION DE LAS PORCIONES INVOCADAS EN LOS PROGRAMAS, Y CON EL QUE SE CREAN RUTINAS DE ACCESO ESPECIALIZADAS. TAMBIEN ES USADO POR LAS UTILERIAS DE REORGANIZACION, VACIADO E IMPRESION.

EL LENGUAJE DE DEFINICION DE DATOS Y ESTRUCTURAS RECONOCE TRES ENTIDADES BASICAS DENTRO DE UNA BASE DE DATOS. LA TERMINOLOGIA DIFIERE UN TANTO DE LA DE CODASYL. SON LLAMADAS : "CONJUNTOS DE DATOS", "ORDENES", E "ITEMS". UN CONJUNTO DE DATOS ES UN ARCHIVO LOGICO, O COLECCION DE REGISTROS INTERRELACIONADOS, QUE CONTIENEN LA INFORMACION REAL DE LA BASE DE DATOS. LOS CONJUNTOS DE DATOS PUEDEN SER "INCLUIDOS" O "DISJUNTOS". UN CONJUNTO DE DATOS INCLUIDO ES UN ELEMENTO DE OTRO CONJUNTO DE DATOS, CON LO QUE SE TIENE UNA RELACION JERARQUICA ENTRE UN REGISTRO MAESTRO Y UNO DETALLE, O ENTRE UNO PROPIETARIO Y OTRO MIEMBRO. UN CONJUNTO DE DATOS DISJUNTO ES UNA ESTRUCTURA DE DATOS AISLADA, QUE PUEDE ACTUAR COMO RAIZ EN UNA ESTRUCTURA DE ARBOL, O PUEDE LIGARSE A OTRO CONJUNTO DISJUNTO PARA FORMAR ESTRUCTURAS DE ANILLO Y DE RED.

EJEMPLO :

```
CURSOS-UNIV DATA SET (
          NOMBREDECURSO ALPHA(24);
          PROFESOR IS IN PERSONAL-UNIV COUNTED
                    OCCURS 3 TIMES;
          LIBROS UNORDERED DATA SET (
                    CLAVE-BIBLIOTECA NUMBER(9);
                    TITULO ALPHA(6) NULL IS BLANKS;
                    AUTOR ALPHA(30));
          )
          ;
LIBROS-CIENCIAS SUBSET OF LIBROS UNORDERED LIST
          DATA CLAVE-BIBLIOTECA;
PERSONAL-UNIV DATA
          (
          NOMBRE GROUP(
                    APELLIDOS ALPHA(30);
                    NOMBRE-PILA ALPHA(10)) REQUIRED;
          REGISTRO-FEDERAL ALPHA(9) REQUIRED UNIQUE;
          )
          ;
REGISTRO-EMPLEADO SET OF PERSONAL-UNIV
          KEY IS REGISTRO-FEDERAL I-S NO DUPLICATES;
```

"CURSOS-UNIV" Y "PERSONAL-UNIV" APARECEN COMO CONJUNTOS DE DATOS DISJUNTOS, Y UN EJEMPLO DE LIGA LO CONSTITUYE "PROFESOR". "LIBROS" APARECE COMO CONJUNTO DE DATOS INCLUIDO, SIENDO SU PROPIETARIO "CURSOS-UNIV". LA INFORMACION SE TIENE EN DOS ARCHIVOS, DE CURSOS Y DE PERSONAL. POR CADA REGISTRO DE CURSOS SE TIENE UN ARCHIVO DE LIBROS, EL NOMBRE DEL CURSO, Y TRES APUNTADORES AL ARCHIVO DE PERSONAL PARA LOS PROFESORES. POR PROGRAMA SE MANTIENE UN ACCESO A LOS LIBROS DE CIENCIAS, Y SE ACCESA AL PERSONAL EN ORDEN DE REGISTRO FEDERAL.

SE PUEDEN DEFINIR Y CONTROLAR LA VERIFICACION DE INFORMACION EN LA BASE DE DATOS, POR MEDIO DEL LENGUAJE DE DEFINICION DE DATOS Y ESTRUCTURAS. A NIVEL DE ITEM SE PUEDE CONTROLAR LA INTEGRIDAD DE LA INFORMACION CON LA OPCION DE REQUERIDO, EN LA DESCRIPCION DE REGISTRO, QUE ESPECIFICA QUE EL ITEM DESIGNADO HA DE ESTAR PRESENTE EN CADA NUEVO REGISTRO ANADIDO AL CONJUNTO DE DATOS, Y EN CADA REGISTRO ACTUALIZADO. LA OPCION DE REQUERIDOS TODOS OBLIGA LA PRESENCIA DE TODOS LOS ITEMS PARA LOS QUE LA OPCION ES VALIDA. TAMBIEN SE PUEDEN DECLARAR LOS ITEMS COMO VERIFICADOS, LO QUE CAUSA QUE EL SISTEMA AUTOMATICAMENTE PRUEBE POR LAS CONDICIONES ESTABLECIDAS EN LA CLAUSULA DE VERIFICACION, ANTES DE ALMACENAR EL REGISTRO QUE LO CONTENGA.

SE PUEDEN DEFINIR DATOS GLOBALES, CON INFORMACION ESTADISTICA O DE RESUMEN RELACIONADA CON UN CONJUNTO DE DATOS O CON LA TOTALIDAD DE LA BASE. LOS ITEMS GLOBALES PUEDEN REPRESENTAR TOTALES DE CONTROL, DE DISPERSION, O POBLACIONES, Y PUEDEN USARSE CON PROPOSITOS DE CONTROL O DE CONSULTA.

SE PUEDE OBTENER COMPACTACION DE DATOS EN DISCO O PAQUETE USANDO UN CONJUNTO DE DATOS "COMPACTO" O A TRAVES DEL USO DE REGISTROS DE FORMATO VARIABLE. SE USA UNA DEFINICION DE TIPO PARA EL NUMERO MAXIMO DE TIPOS DE REGISTRO DE FORMATO VARIABLE PERMITIDOS PARA CADA CONJUNTO DE DATOS. SE CONSTITUYE UNA PARTE FIJA CON LOS ITEMS COMUNES A TODOS LOS TIPOS DE REGISTRO. DMS II MANTIENE UN INDICADOR DE FORMATO, Y CADA REGISTRO FISICO ESCRITO EN EL CONJUNTO DE DATOS INCLUYE AUTOMATICAMENTE LA PARTE FIJA, ANEXANDOSE LA PARTE VARIABLE CORRESPONDIENTE A MEDIDA QUE SE ESCRIBEN LOS REGISTROS DE LOS DISTINTOS TIPOS.

LOS REGISTROS FISICOS SE RELACIONAN LOGICAMENTE USANDO "ORDENES", "SUBORDENES" Y "LIGAS". LAS DEFINICIONES DE ORDEN DAN LUGAR A TABLAS DE LLAVES ; DESCRIBEN LAS LLAVES PARA EL ACCESO Y LOS METODOS DE RECUPERACION DE REGISTROS, MAS VARIAS OPCIONES PARA LA ORGANIZACION DEL CONJUNTO DE DATOS, COMO ORDEN DE LOS REGISTROS, PRESENCIA DE LLAVES DUPLICADAS, ETC. CADA ORDEN O TABLA DE LLAVES CONTIENE UNA RUTA A LA BASE POR CADA REGISTRO, Y PUEDE HABER CUALQUIER NUMERO DE ORDENES ASOCIADOS CON UN CONJUNTO DE DATOS DADO, PARA CONTAR CON MULTIPLES ENTRADAS A LOS REGISTROS. PARA CUANDO SE REQUIERE ACCESO RAPIDO A INFORMACION DE LLAVES, SE PUEDEN ALMACENAR ITEMS DE DATOS EN LOS INDICES.

SE PUEDEN DEFINIR ORDENES PARCIALES DE INFORMACION, PERMITIENDO LA RECUPERACION DE MIEMBROS DE UN CIERTO CONJUNTO DE DATOS SELECCIONADOS SEGUN UN CRITERIO ESPECIFICO.

TODOS LOS ORDENES SON AUTOMATICOS, DE MODO QUE EN ALTA O BAJA DE REGISTROS SE ACTUALIZAN TODOS LOS ORDENES RELACIONADOS. LOS SUBORDENES PUEDEN SER MANUALES O AUTOMATICOS. SI EL SUBORDEN ES MANUAL, EL PROGRAMADOR DE APLICACION ES RESPONSABLE DE LAS ALTAS Y BAJAS CORRESPONDIENTES. SI EL SUBORDEN ES AUTOMATICO, SE EXAMINA CADA REGISTRO A SER DADO DE ALTA, Y DE CUMPLIR CON EL CRITERIO TAMBIEN EN EL SUBORDEN ES DADO DE ALTA. LOS SUBORDENES TAMBIEN PUEDEN SER INCLUIDOS EN UN CONJUNTO DE DATOS, CON LO QUE SON MANTENIDOS COMO AUTOMATICOS. SE MANTIENE UN INDICE SEPARADO DE LLAVES PARA LOS SUBORDENES DEFINIDOS, Y SE PUEDE RELACIONAR LOGICAMENTE A UN MIEMBRO DEL CONJUNTO CON MIEMBROS DE OTRO CONJUNTO DE DATOS EN UNA BASE JERARQUICA.

EN LA ANTERIOR DESCRIPCION, "LIBROS-CIENCIAS" ES UN SUBORDEN MANUAL, QUE HA DE MANTENERSE, INDICANDO AL SISTEMA QUE SE DE DE ALTA EN EL CUANDO CORRESPONDA ANTES DE DAR DE ALTA UN REGISTRO EN LIBROS. EN CASO DE HABERSE DESCRITO UNA CONDICION, EL SUBORDEN SERIA AUTOMATICO, Y EL SISTEMA LO MANTENDRIA. "REGISTRO-EMPLEADO" REPRESENTA UN ORDEN PARA "PERSONAL-UNIV".

SE PUEDEN ENCADENAR MIEMBROS DE CONJUNTOS DE DATOS, PARA LO QUE SE CUENTA CON CINCO TIPOS DE LIGAS : SIMBOLICA, CONTADA, VERIFICADA, AUTO- CORREGIBLE, Y NO PROTEGIDA. EN EL REGISTRO PROPIETARIO SE TIENE UNA DIRECCION RELATIVA DE DISCO APUNTANDO AL MIEMBRO RELACIONADO, CON INFORMACION QUE PERMITE LA RECUPERACION DIRECTA DE LOS DATOS, GENERALMENTE EN UNO O DOS ACCESOS ADICIONALES A DISCO.

DETALLE DE DEFINICION DE DATOS Y ESTRUCTURAS.

EN LA DECLARACION DE BASE DE DATOS SE ESPECIFICAN ITEMS, CONJUNTOS DE DATOS Y ORDENES DE PRIMER NIVEL. LLEVA PARAMETROS DE TIEMPO DE EJECUCION, COMO MAXIMO DE MEMORIA A SER USADA PARA AREAS DE ENTRADA/ SALIDA ANTES DE QUE PUEDAN SER MANDADAS A DISCO, EL NUMERO DE TRANSACCIONES ENTRE PUNTOS DE SINCRONIA, Y EL NUMERO DE PUNTOS DE SINCRONIA ENTRE PUNTOS DE CONTROL. TAMBIEN LLEVA OPCIONES, QUE DETERMINAN CARACTERISTICAS DE LAS RUTINAS DE ACCESO, A SABER, QUE SE GENERE CODIGO QUE DE INFORMACION DEL DESEMPENO DE LA BASE A TIEMPO DE EJECUCION, Y DE TRAZA DE MODIFICACIONES. PARA LA TRAZA DE MODIFICACIONES SE PUEDEN ESPECIFICAR LOS ATRIBUTOS DE LOS ARCHIVOS, COMO TAMANO DE LOS BLOQUES, TAMANO Y NUMERO DE AREAS EN DISCO, ETC.

EN LA DECLARACION DE CONJUNTO DE DATOS SE ESPECIFICAN SU TIPO, OPCIONES FISICAS, EL QUE SEAN REQUERIDOS TODOS LOS ITEMS AL DAR DE ALTA UN REGISTRO, Y LAS DESCRIPCIONES DE REGISTRO, CON LAS CONDICIONES A VERIFICARSE SOBRE LOS ITEMS.

LA DESCRIPCION DE REGISTRO ESPECIFICA EL FORMATO DE UN REGISTRO GENERICO EN UN CONJUNTO DE DATOS. LOS REGISTROS ASI ESPECIFICADOS SERAN ALMACENADOS EN UN ARCHIVO FISICO. ES UNA SERIE DE ITEMS, QUE EN EL CASO DE REGISTROS DE FORMATO VARIABLE ES PRECEDIDA POR UN NUMERO DANDO EL TIPO DE REGISTRO.

UN ITEM PUEDE SER OTRO CONJUNTO DE DATOS, UN ORDEN, UNA LIGA, UN ITEM DE CONTROL, UN ITEM DE DATOS O UN ITEM DE GRUPO. UN ITEM DE DATOS CONLLEVA TIPO, SEA BOOLEANO, ALFABETICO, NUMERICO, REAL, O CAMPO, Y OPCIONES, COMO EL QUE SEA REQUERIDO, QUE LAS LLAVES SE PRESENTEN SOLO UNA VEZ, EL CUANTAS VECES APARECE EN UN GRUPO, LOS VALORES DE DEFAULT PARA NULO Y VALOR INICIAL, Y QUE EMPIECE A FRONTERA DE PALABRA.

UN ITEM DE CONTROL PUEDE SER PARA REINICIO, COMO TIPO Y LA CUENTA DE TRANSACCIONES EN LOS CONJUNTOS DE DATOS DE REINICIO, DE CUENTA, PARA EL NUMERO DE REFERENCIAS APUNTANDO A ESTE REGISTRO, Y DE POBLACION, DE UN CONJUNTO DE DATOS U ORDEN INCLUIDO.

UN ITEM DE GRUPO CONTIENE A SU VEZ ITEMS, PUEDE SER REQUERIDO Y OCURRIR UN NUMERO VARIABLE DE VECES.

LAS CONDICIONES QUE SE HAN DE CUMPLIR PARA EL ALMACENAMIENTO DE UN REGISTRO SON EXPRESIONES BOOLEANAS COMBINANDO ITEMS DE DATOS, CONSTANTES Y OPERACIONES, DE UNA MANERA LOGICA Y ESPECIFICA.

LOS ITEMS DE LIGA DETERMINAN RELACIONES ENTRE MIEMBROS DE CONJUNTOS DE DATOS, EN ALGUNOS CASOS MEDIANTE ORDENES.

LA DECLARACION DE UN ORDEN ESPECIFICA A QUE CONJUNTO DE DATOS ESTA ASOCIADO, SU ESTRUCTURA DE LLAVE Y OPCIONES. UN SUBORDEN SE ASOCIA A UN CONJUNTO DE DATOS O A OTRO ORDEN, Y PUEDE SER AUTOMATICO ESPECIFICANDO UNA CONDICION.

LA ESTRUCTURA DE LLAVE DETERMINA LA LLAVE, DANDO EL O LOS ITEMS QUE LA CONSTITUYEN, PARA LAS ESTRUCTURAS SECUENCIAL CON INDICE, AL AZAR CON INDICE, LISTA ORDENADA, SI NO SE PERMITEN LLAVES DUPLICADAS Y SI SE PERMITEN EN QUE ORDEN, AL PRINCIPIO O AL FINAL SE ALMACENAN LOS REGISTROS CON LLAVES IGUALES, Y LOS DATOS A GUARDARSE EN LAS TABLAS DE LLAVE. TAMBIEN SE PUEDEN ESPECIFICAR LISTA DESORDENADA Y SUS DATOS EN TABLA, Y ESTRUCTURA DE VECTOR DE BITS.

LAS OPCIONES DE LOS ORDENES ESPECIFICAN LOS ATRIBUTOS FISICOS DE LAS TABLAS DE INDICES, TALES COMO NUMERO DE AREAS DE ENTRADA/SALIDA, TAMANOS DE TABLAS, NUMERO Y TAMANO DE AREAS EN DISCO, POBLACION, FACTOR DE CARGA, QUE DETERMINARA LA GENERACION DE CADA NUEVA TABLA, Y EL MODULO A USARSE EN LA TECNICA DE ACCESO AL AZAR CON INDICE.

IV.2.2 INTERFASE CON EL USUARIO : HOST.

EN DMS II SE PREVEE EL USO DE LAS BASES DE DATOS BAJO DOS LENGUAJES HUESPEDES : COBOL Y ALGOL.

LAS INTERFASES PARA AMBOS LENGUAJES SON MUY SIMILARES ENTRE SI SINTACTICAMENTE ; LOS POSTULADOS PARA ACCESAR UNA BASE SON CASI IDENTICOS, APRECIANDOSE LAS MAYORES DIFERENCIAS EN LA INVOCACION DE LA BASE Y EN EL PROCESO DE EXCEPCIONES.

SIN EMBARGO, Y DADO QUE ALGOL NO CUENTA CON UNA ESTRUCTURA DE DATOS DE REGISTRO SIMILAR A LA DE COBOL, EL ACCESO A ITEMS INDIVIDUALES EN UNA BASE DIFIERE SUSTANCIALMENTE ENTRE AMBOS LENGUAJES. EN UN PROGRAMA DE COBOL, UN REGISTRO DE DATOS SE ACCESA DIRECTAMENTE. EN ALGOL SE USA UN AREA DE REGISTRO, PERO LA INFORMACION NO SE PUEDE ACCESAR SINO MEDIANTE UN MAPEO EXPLICITO DE SUS ITEMS A VARIABLES DE ALGOL.

INVOCACIONES EN COBOL.

SE AMPLIO COBOL CON UNA SECCION A REFERIRSE A UNA BASE DE DATOS, EN LA DIVISION DE DATOS, ENTRE LAS SECCIONES DE ARCHIVO Y DE TRABAJO Y ALMACENAMIENTO. EN ESTA, SE USA EL INDICADOR DE NIVEL DB PARA SELECCIONAR LA BASE. LAS REFERENCIAS A LA BASE EN LA DIVISION DE PROCEDIMIENTO PUEDEN SER CON UN SINONIMO QUE SE ESPECIFICA AQUI. ESTA OPCION, DE NOMBRE INTERNO, ES APLICABLE TAMBIEN A CONJUNTOS DE DATOS Y ORDENES, LO QUE PERMITE ESTABLECER MULTIPLES AREAS DE REGISTRO PARA CADA CONJUNTO DE DATOS, Y MULTIPLES RUTAS PARA CADA ORDEN. SE PUEDEN INVOCAR YA SEA LA TOTALIDAD DE LAS ESTRUCTURAS DE LA BASE, O SELECCIONAR CON NIVEL 01, SOLO ALGUNOS CONJUNTOS DE DATOS Y ORDENES. EL COMPILADOR SELECCIONA IMPLICITAMENTE LAS ESTRUCTURAS INCLUIDAS, Y PRESENTA EN EL LISTADO, EN FORMATO DE COBOL, TODOS LOS FORMATOS DE REGISTRO Y DESCRIPCIONES DE ITEMS, ORDENES Y LLAVES.

DADOS UN CONJUNTO DE DATOS Y UNO O MAS ORDENES RELACIONADOS CON EL, SE PUEDE ESTABLECER CUALQUIERA DE LAS SIGUIENTES SITUACIONES :

1) UN AREA DE REGISTRO CORRIENTE PARA EL CONJUNTO DE DATOS (D), UNA RUTA PARA EL ORDEN (S).
   01 D. (S IMPLICITO)
   LA INSTRUCCION FIND S CARGARIA LOS DATOS AL AREA DE REGISTRO DE D.

2) VARIAS AREAS DE REGISTRO CORRIENTE (D, X), Y UNA RUTA (S).
   01 D. (S IMPLICITO)
   01 X = D USING NONE. (NO SE RELACIONA RUTA)
   SI SE EJECUTA FIND S, SE CARGARIA EL AREA DE REGISTRO DE D.
   PARA CARGAR X, SE HA DE EJECUTAR FIND X VIA S.

3) VARIAS AREAS DE REGISTRO CORRIENTE (D, X) CADA UNA CON UNA RUTA (S DE D, S DE X).
   01 D. (S IMPLICITO)
   01 X = D. (S IMPLICITO)
   CON FIND S OF D SE CARGA EL AREA DE REGISTRO DE D, Y CON FIND S OF X LA DE X. EN CUALQUIER CASO, NO SE INTERFIERE CON LA RUTA NO USADA.

4) MAS AREAS DE REGISTRO CORRIENTE (D, X, Y) QUE RUTAS (S DE D, S DE X).
   01 D. (S IMPLICITO)
   01 X = D. (S IMPLICITO)
   01 Y = D USING NONE. (NO SE RELACIONA RUTA)
   PARA CARGAR EL AREA DE Y, SE HA DE EJECUTAR SEA FIND Y VIA S OF D, O FIND Y VIA S OF X.

5) UNA REFERENCIA A UN ORDEN USANDO UN SINONIMO, O ASOCIAR IMPLICITAMENTE UN ORDEN CON UN AREA DE REGISTRO DADA.
   01 X = D USING S.
   01 Y = D USING T.
   FIND S CARGARIA EL AREA DE X, Y FIND T EL AREA DE Y.

TAMBIEN SE PUEDE ESTABLECER UN ORDEN SIN ASOCIARLO IMPLICITAMENTE CON NINGUNA AREA PARTICULAR DE REGISTRO.
   01 SY = S. (S ORDEN DE ALGUN CONJUNTO DE DATOS D)
   PARA USARLO, SE TIENE LA FRASE VIA, COMO EN FIND D VIA SY.

DEBEN DE USARSE TODOS LOS ORDENES ASOCIADOS A CADA CONJUNTO DE DATOS. TAMBIEN, SE DEBEN INVOCAR LOS CONJUNTOS DE DATOS ASOCIADOS CON ORDENES INCLUIDOS O CON LIGAS, CUANDO A TRAVES DE ELLOS SE VAYAN A ESTABLECER RUTAS.

DECLARACION DE BASE DE DATOS EN ALGOL.

COMO TODO IDENTIFICADOR, AQUEL QUE NOMBRE O REPRESENTE INTERNAMENTE EN EL PROGRAMA A UNA BASE DE DATOS HA DE SER DECLARADO CON ANTERIORIDAD A SU USO. EN ESTE CASO, SE TRATA DE UNA INVOCACION, YA SEA DE LA TOTALIDAD DE LAS ESTRUCTURAS PREVISTAS EN EL ARCHIVO DESCRIPTOR, O DE SOLO ALGUNOS DE LOS CONJUNTOS DE DATOS Y ORDENES, CONSTITUYENDO LA BASE DE DATOS INTERNA. ESTA CONTIENE POR CADA CONJUNTO DE DATOS INVOCADO UN AREA DE REGISTRO, Y POR CADA ORDEN INVOCADO, UN APUNTADOR DE REGISTRO CORRIENTE.

SE PUEDEN MANIPULAR VARIOS REGISTROS DE UN MISMO CONJUNTO DE DATOS SIMULTANEAMENTE, SI SE INVOCA LA MISMA ESTRUCTURA VARIAS VECES, ESTABLECIENDO TANTAS AREAS DE REGISTRO, APUNTADORES AL REGISTRO Y RUTAS COMO SE NECESITE. UNA OPCION DEL COMPILADOR PRESENTA EN EL LISTADO TODAS LAS ESTRUCTURAS INVOCADAS, CON FORMATOS DE REGISTROS, DESCRIPCIONES DE ITEMS Y DE LLAVES Y DEMAS INFORMACION PERTINENTE.

DIVISION DE PROCEDIMIENTO EN COBOL.

EL AREA DE REGISTRO PARA UN CONJUNTO DE DATOS SE ESTABLECE EN DOS PARTES, UNA PARA ITEMS DE CONTROL Y OTRA PARA ITEMS DE DATOS. LA DE DATOS ES SIMILAR A UNA ENTRADA 01 DE TRABAJO-ALMACENAMIENTO, CON LO QUE SE PUEDE USAR EN LAS INSTRUCCIONES COMUNES DE COBOL.

PROCESO DE EXCEPCIONES. PARA CADA PROGRAMA EXISTE UN REGISTRO ESPECIAL, LLAMADO DMSTATUS, QUE CADA VEZ QUE TERMINA UNA INSTRUCCION DE MANEJO DE LA BASE RECIBE UN VALOR, DETERMINADO POR LA NATURALEZA DE LAS EXCEPCIONES POSIBLES EN CADA INTRUCCION. DMSTATUS CONLLEVA UNA SERIE DE ATRIBUTOS PARA AISLAR LAS DISTINTAS EXCEPCIONES. UNO DE ELLOS DA UN VALOR DE CIERTO SI OCURRE UNA EXCEPCION DENTRO DE UNA CATEGORIA PARTICULAR, COMO NO ENCONTRADO, DUPLICADO, ETC. OTROS DOS SON ATRIBUTOS NUMERICOS, IDENTIFICANDO EL ERROR Y SU ORIGEN, A SABER, SUBCATEGORIA Y NUMERO DE ESTRUCTURA INVOLUCRADA.

LA SECCION DE DECLARATIVAS DE LA DIVISION DE PROCEDIMIENTO EN COBOL SE EXTENDIO PARA PERMITIR ESPECIFICAR UN PROCEDIMIENTO A USARSE EN CASO DE ERROR EN MANEJO DE LA BASE. ESTE PROCEDIMIENTO ES LLAMADO CUANDO ES CAUSADA UNA EXCEPCION QUE NO TIENE ASOCIADA UNA CLAUSULA DE ON EXCEPTION. DE SUCEDER UNA EXCEPCION SIN HABER NINGUNA DE ESTAS DOS PREVISIONES, EL PROGRAMA ES TERMINADO, REPORTANDOSE LA CATEGORIA Y SUBCATEGORIA DEL ERROR, Y EL NUMERO DE LA ESTRUCTURA INVOLUCRADA.

PROCESO DE EXCEPCIONES EN ALGOL.

AL TERMINO DE CADA INSTRUCCION DE MANEJO DE LA BASE, SE REGRESA AL PROGRAMA DE ALGOL UNA PALABRA DE STATUS, CON SUBCAMPOS IDENTIFICANDO SI OCURRIO UN ERROR, LA CATEGORIA Y SUBCATEGORIA DE ESTE, Y EL NUMERO DE LA ESTRUCTURA INVOLUCRADA. SI OCURRE UN ERROR Y NO SE DIO VARIABLE PARA EL STATUS, SE TERMINA EL PROGRAMA. SI SE DA VARIABLE PERO NO SE TOMAN ACCIONES EN CASO DE ERROR, PUEDE LLEGARSE A RESULTADOS IMPREDECIBLES.

ATRIBUTOS DE LA BASE.

EN COBOL, EL USO DE LOS ATRIBUTOS ES SIMILAR AL DE LOS ATRIBUTOS DE TAREA Y ARCHIVO. EN ALGOL, ESTOS ITEMS SON USADOS DE MANERA SIMILAR A LOS ITEMS DE DATOS, PERO NO PUEDEN CAMBIARSE SUS VALORES DIRECTAMENTE.

ATRIBUTO DE CUENTA. EL VALOR DE ESTE ATRIBUTO PARA UN REGISTRO DADO ES EL NUMERO DE REFERENCIAS APUNTANDOLE. HA DE CONSULTARSE INMEDIATAMENTE DESPUES DE QUE EL REGISTRO SE HACE CORRIENTE. OCURRE UNA EXCEPCION SI SE TRATA DE DAR DE BAJA AL REGISTRO Y ESTA CUENTA NO ES CERO, PERO ES MAS EFICIENTE CONDICIONAR LA BAJA AL VALOR DEL ATRIBUTO.

ATRIBUTO DE TIPO DE REGISTRO. EN UN REGISTRO CORRIENTE DE FORMATO VARIABLE, ESTE SE REFIERE AL TIPO DE REGISTRO.

ATRIBUTO DE POBLACION. EL VALOR DE ESTE ATRIBUTO ES EL DE LA POBLACION CORRIENTE DE UN CONJUNTO DE DATOS INCLUIDO. HA DE CONSULTARSE INMEDIATAMENTE DESPUES DE QUE EL REGISTRO QUE LO CONTIENE SE HACE CORRIENTE.

INSTRUCCIONES DE DMS II BAJO LENGUAJE HUESPED.

"OPEN" ABRE UNA BASE DE DATOS PARA SU USO. "FIND" LEE UN REGISTRO DE UN CONJUNTO DE DATOS. SI SE DESEA ENCONTRARLO Y BLOQUEAR MODIFICACIONES CONCURRENTES DE OTROS USUARIOS, SE USA "LOCK" ("MODIFY"). PARA ESCRIBIR UN NUEVO REGISTRO EN UN CONJUNTO DE DATOS, O ACTUALIZAR UNO YA EXISTENTE, SE USA "STORE". PARA DAR DE BAJA UN REGISTRO SE USA "DELETE", QUE TAMBIEN LO DA DE BAJA DE TODOS LOS ORDENES AUTOMATICOS. "FREE" DESBLOQUEA UN REGISTRO QUE FUE BLOQUEADO CON ANTERIORIDAD. "GENERATE" CREA UN SUBORDEN A PARTIR DE UNO O DOS SUBORDENES. "GET" Y "PUT" SE USAN PARA TRANSFORMAR VARIABLES DE LA BASE EN VARIABLES DE ALGOL Y VICEVERSA. CON "SET" SE POSICIONA A LOS CONJUNTOS DE DATOS A SU PRINCIPIO O FINAL, O SE DA VALOR DE NULO A ITEMS DE DATOS EN EL REGISTRO CORRIENTE. "CLOSE" CIERRA UNA BASE DE DATOS. "CREATE" INICIALIZA EL AREA DE REGISTRO DE UN CONJUNTO DE DATOS. "RECREATE" INICIALIZA EN EL AREA DE REGISTRO DE UN CONJUNTO DE DATOS SOLO LOS ITEMS DE CONJUNTOS DE DATOS INCLUIDOS, ORDENES, DE LIGAS Y DE CONTROL. "INSERT" INCORPORA UN REGISTRO A UN ORDEN MANUAL. "REMOVE" SACA UN REGISTRO DE UN ORDEN MANUAL. "ASSIGN" ESTABLECE UNA LIGA A UN REGISTRO DE UN CONJUNTO DE DATOS. "BEGIN-TRANSACTION" PONE AL PROGRAMA EN ESTADO DE TRANSACCION, Y "END-TRANSACTION" LO SACA.

LOS VERBOS DE MANIPULACION DE DATOS OPERAN COMO UNA EXTENSION AL COMPILADOR DE APLICACION. INCLUYEN ABRE, CIERRA, ALMACENA, MODIFICA (BLOQUEA), DA DE BAJA, LIBERA, ENCUENTRA, CREA, RECREA, INSERTA Y REMUEVE. LA CLAUSULA DE INVOCACION PERMITE AL PROGRAMADOR CONSIDERAR A LA BASE DE DATOS EN SU TOTALIDAD, O SELECCIONAR CONJUNTOS DE DATOS Y ORDENES.

LAS INSTRUCCIONES DE ENCUENTRA, MODIFICA Y DA DE BAJA SE PUEDEN APLICAR A REGISTROS PARTICULARES, COMO AL PRIMERO, ULTIMO, SIGUIENTE Y ANTERIOR. CUANDO UN CONJUNTO DE DATOS HAA SIDO INVOCADO MAS DE UNA VEZ, S PUEDE USAR UNA FRASE DE "POR MEDIO DE", PARA IDENTIFICAR EL AREA DE REGISTRO DESEADA. LA INSTRUCCION DE BLOQUEA PROTEGE A UN REGISTRO CONTRA MODIFICACION CONCURRENTE POR ALGUN OTRO USUARIO. EL SISTEMA RESUELVE AUTOMATICAMENTE LOS CONFICTOS ENTRE ACCESOS, SEA SEGUN LAS PRIORIDADES O SEA SEGUN EL MOMENTO DE SOLICITUD, Y REGRESA UNA EXCEPCION AL LIBERAR LOS REGISTROS DE UN PROGRAMA QUE BLOQUEE AL SISTEMA.

EL COMANDO DE LIBERA ES OPCIONAL EN MUCHOS CASOS, DADO QUE SE EJECUTA UNO IMPLICITAMENTE ANTES DE CUALQUIER OPERACION QUE ESTABLEZCA UN NUEVO APUNTADOR A REGISTRO CORRIENTE. TAMBIEN SE CIERRAN AUTOMATICAMENTE LAS BASES AL TERMINO DE UN PROGRAMA DE USUARIO. INSERTA Y REMUEVE SE USAN CON SUBORDENES MANUALES. CREA Y RECREA INICIALIZAN LOS ITEMS DE DATOS EN UN AREA DE CONTROL. SI SE HAN DEFINIDO LLAVES MULTIPLES Y NO TODAS SON CONOCIDAS, SE PUEDEN ESPECIFICAR EXPRESIONES PARCIALES DE SELECCION.

SELECCION DE REGISTRO.

SE USA UNA EXPRESION DE SELECCION PARA IDENTIFICAR UN REGISTRO PARTICULAR EN UN CONJUNTO DE DATOS, PARA LAS INSTRUCCIONES DE ENCUENTRA, MODIFICA Y DA DE BAJA. SE PUEDEN SENALAR EL AREA DE REGISTRO Y APUNTADOR A SER USADOS, CUANDO SE HA INVOCADO EL CONJUNTO DE DATOS MAS DE UNA VEZ. SE TIENEN LAS SIGUENTES FORMAS :

CON SOLO EL NOMBRE DEL ORDEN O DEL CONJUNTO DE DATOS, SE REFIERE REGISTRO INDICADO POR LA RUTA O POR EL APUNTADOR DE REGISTRO CORRIENTE.

CON PRIMERO O ULTIMO, SIGUIENTE O ANTERIOR, SE REFIERE AL CORRESPONDIENTE REGISTRO EN EL ORDEN O CONJUNTO DE DATOS.

SI SE ESPECIFICAN NOMBRE DE ORDEN Y CONDICION DE LLAVE, SE PUEDE SELECCIONAR EL PRIMER REGISTRO, O EL SIGUIENTE, DE LOS QUE CUMPLEN ESA CONDICION DE LLAVE, QUE PUEDE SER COMPLEJA, CON OPERACIONES LOGICAS SOBRE LOS ITEMS DE LA LLAVE.

CON UN NOMBRE DE REFERENCIA, SE SELECCIONA EL REGISTRO APUNTADO POR LA LIGA.

IV.3 ESTRUCTURAS FISICAS EN DMS II.

IV.3.1 CARACTERISTICAS COMUNES.

BLOQUEO.

LOS ARCHIVOS DE DATOS DE LAS BASES DE DATOS TIENEN COMO TAMANO DE REGISTRO Y DE BLOQUE 30 PALABRAS, CORRESPONDIENDO CON LA UNIDAD BASICA EN DISCO.

LAS RUTINAS DE ACCESO LEEN Y ESCRIBEN MEDIANTE DMIO, QUE ES UNA VARIANTE DE ENTRADA/SALIDA DIRECTA, A CUALQUIER SEGMENTO DE LOS ARCHIVOS DE LOS DE LA BASE DE DATOS.

LAS RUTINAS DE ACCESO GENERALMENTE MANEJAN UN TAMANO DE BLOQUE CORRESPONDIENTE A CADA ARCHIVO, CON CADA BLOQUE ALINEADO A FRONTERA DE SEGMENTO. LA DIRECCION DEL BLOQUE LA CONSTITUYE EL NUMERO DE SEGMENTO, O NUMERO DE REGISTRO DEL MCP. CADA RENGLON DEL ARCHIVO ES UN NUMERO ENTERO DE BLOQUES. ALGUNAS PARTES DE ALGUNOS ARCHIVOS NO SON MANEJADAS COMO BLOQUES. LAS PALABRAS EN UN BLOQUE SE NUMERAN DESDE CERO.

CHEQUEO DE DIRECCION Y SUMA DE CHEQUEO.

EN DEFINICION DE DATOS Y ESTRUCTURAS SE PUEDE ESPECIFICAR CHEQUEO DE DIRECCIONES, LO QUE INCLUIRA UNA PALABRA CON LA DIRECCION DE CADA BLOQUE FISICO DE CADA ESTRUCTURA, SIGUIENDO LAS PALABRAS DE DATOS, Y LA SUMA DE CHEQUEO SI ESTA ESTA PRESENTE. EL CHEQUEO DE DIRECCIONES DE DMSREAD DARA UN ERROR SI EL VALOR NO CORRESPONDE.

TAMBIEN SE PUEDE ESPECIFICAR PARA UN ORDEN O PARA UN CONJUNTO DE DATOS LA SUMA DE CHEQUEO, LO QUE INCLUIRA UNA PALABRA DE CHEQUEO A CADA BLOQUE, COMO RESULTADO DE UNA APLICACION A TODAS LAS PALABRAS DEL BLOQUE, EXCLUYENDO LA DE CHEQUEO DE DIRECCION, Y QUE SE CALCULA CUANDO SE HACEN LECTURAS O ESCRITURAS FISICAS.

NO SE PONEN CHEQUEO DE DIRECCION NI SUMA DE CHEQUEO EN LOS BLOQUES QUE NO SON LEIDOS POR DMSREAD.

PALABRAS DE DIRECCION ABSOLUTA, DE CONTROL DE TABLA
Y DE NUMERO DE SERIE DE TABLA.

AA = DIRECCION ABSOLUTA

ESQUEMA DE PALABRA DE DIRECCION ABSOLUTA

```
A A A A A A  A * B B B B
A A A A A A  A * B B B B
A A A A A A  A * B B B B
A A A A A A  A * B B B B
```

A	47:28	CAMPO DE DIRECCION DE BLOQUE
*	19:04	TIENE CEROS CUANDO SE REFIERE A DISCO
B	15:16	CAMPO DE DIRECCION DE PALABRA

INCLUYE UN CAMPO DE DIRECCION DE BLOQUE Y UNO DE DIRECCION DE PALABRA

TCW : PALABRA DE CONTROL DE TABLA, PRIMERA PALABRA DE BLOQUE PARA LAS ESTRUCTURAS SECUENCIAL CON INDICE, LISTA ORDENADA, LISTA DESORDENADA Y AL AZAR CON INDICE. EN UNA TABLA EN USO INCLUYE PRIMERA ENTRADA EN USO, CUENTA DE ENTRADAS EN USO, Y PARA SECUENCIAL CON INDICE UN DIGITO BINARIO DE TABLA FINA. EN UNA TABLA DE DISPONIBLES INCLUYE LA DIRECCION DE BLOQUE DE LA SIGUIENTE TABLA DISPONIBLE.

ESQUEMA DE PALABRA DE CONTROL DE TABLA (TABLA EN USO)

```
A A A A B B  B B B B B B
A A A A B B  B B B B B B
A A A A B B  B B B B B B
A A A A B B  B B B B B C
```

A	47:16	COMIENZO DE TABLA
B	31:31	CUENTA DE TABLA
C	00:01	INDICADOR DE TABLA FINA

ESQUEMA DE PALABRA DE TABLA DE CONTROL (TABLA DISPONIBLE)

```
A A A A A A  A A A A A A
A A A A A A  A A A A A A
A A A A A A  A A A A A A
A A A A A A  A A A A A A
```

A	47:48	DIRECCION DE BLOQUE DE SIGUIENTE TABLA DISPONIBLE

TSN : NUMERO DE SERIE DE TABLA,, SEGUNDA PALABRA DE BLOQUE PARA LAS ESTRUCTURAS SECUENCIAL CON INDICE, LISTA ORDENADA, LISTA DESORDENADA, AL AZAR CON INDICE Y CONJUNTOS ORDENADOS DE DATOS. SE USA FUNDAMENTALMENTE PARA SINCRONIZAR EL ESTADO DE LA TABLA CON LA TRAZA DE MODIFICACIONES.

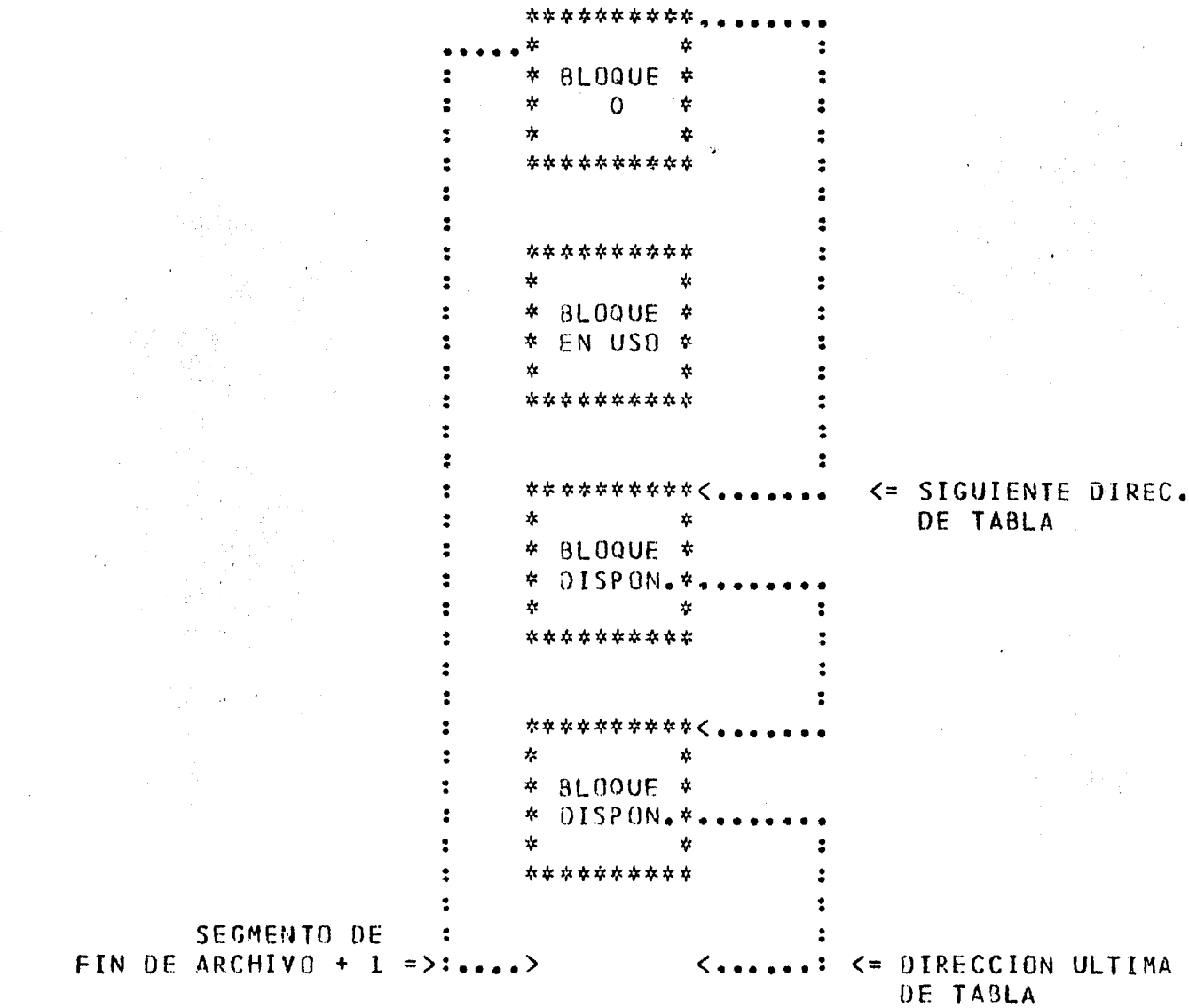
ESQUEMA DE NUMERO DE SERIE DE TABLA

```
A A A A A B  B B B B B B
A A A A A B  B B B B B B
A A A A A B  B B B B B B
A A A A A B  B B B B B B
```

A	47:18	>0 CUANDO SE HAN REGRESADO PARA RECUPERACION ALGUNOS CAMBIOS EN LA TABLA CORRIENTE, SIN HABERSE CAMBIADO LA TABLA OTRA VEZ
B	29:30	SINCRONIA CON AUDITORIA

CADENA DE BLOQUES DISPONIBLES.

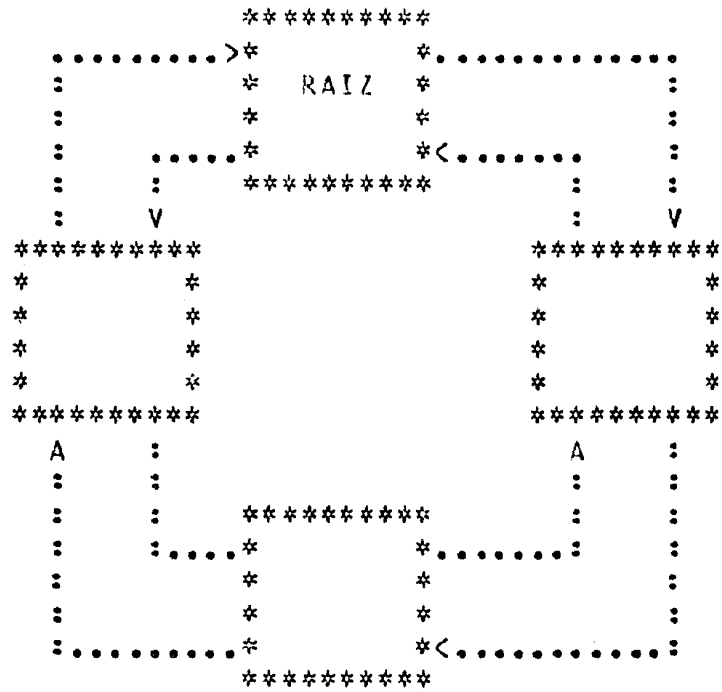
EN LAS ESTRUCTURAS SECUENCIAL CON INDICE, LISTA DESORDENADA, AL AZAR CON INDICE, AL AZAR, CONJUNTOS DE DATOS ORDENADOS Y DESORDENADOS, SE CUENTA CON UNA LISTA LIGADA EN UN SENTIDO, LA CUAL CONTIENE BLOQUES DISPONIBLES, ASIGNANDOSE Y DESASIGNANDOSE SEGUN SE REQUIERE. EL INICIO DE LA CADENA RESIDE EN EL BLOQUE CERO, Y SE PASA A TRAVES DE LA PALABRA CERO DE CADA BLOQUE DISPONIBLE. LA PRIMERA PALABRA DEL BLOQUE CERO TIENE LA DIRECCION DE BLOQUE DEL SIGUIENTE DISPONIBLE, Y LA SEGUNDA LA DEL PRIMER BLOQUE QUE NO HA DE USARSE. CUANDO YA NO HAY BLOQUES EN LA CADENA, LOS VALORES DE ESTAS DOS PALABRAS SON IGUALES.



PARA MANEJAR ERRORES DE LIMITES, LAS RUTINAS DE ACCESO ACOMODAN PARA CADA ARCHIVO UNA FILA EXTRA EN DISCO CUANDO MENOS.

CADENAS DOBLEMENTE LIGADAS, DE BLOQUES EN USO.

PARA LOS BLOQUES EN USO, LAS ESTRUCTURAS DE LISTA ORDENADA, LISTA DESORDENADA, AL AZAR CON INDICE, CONJUNTOS DESORDENADOS DE DATOS, Y AL AZAR, UTILIZAN UNA LISTA DOBLEMENTE LIGADA PARA LOS BLOQUES EN USO. CADA BLOQUE EN USO TIENE UNA LIGA AL ANTERIOR Y OTRA AL SIGUIENTE, LO QUE POSIBILTA SEGUIR LA CADENA EN CUALQUIER SENTIDO.



ENTRADAS DE LLAVE.

EN LAS TABLAS DE INDICES ENCONTRAMOS ITEMS DE LLAVE, DE RELLENO Y DE SOLUCION DE DUPLICIDAD, DIRECCION ABSOLUTA Y DATOS. LAS ENTRADAS ESTAN SIEMPRE ALINEADAS A FRONTERA DE PALABRA, COMO LO ESTAN LA DIRECCION ABSOLUTA Y EL ELEMENTO DE SOLUCION DE DUPLICIDAD.

```
I<.............. PARTE DE LLAVE ..............>I
<ITEM DE LLAVE>...<ITEM DE LLAVE><RELL><SOL.DUP.><AA><DATOS>

A                 A               A     A        A   A
:                 :               :     :        :   :
A FRONTERA DE :
PALABRA           DIGITO          DIG.  PALABRA  PAL.PALABRA
```

LOS ITEMS DE LLAVE, CONCATENADOS JUNTOS Y CON EL ELEMENTO DE SOLUCION DE DUPLICIDAD CONSTITUYEN LA PARTE DE LLAVE, QUE MANTIENE ORDENADAS LAS ENTRADAS. SI PARA ALGUNOS ITEMS SE ESPECIFICA QUE SE USARAN COMO LLAVE EN ORDEN INVERSO AL NORMAL, AQUI LOS VALORES CORRESPONDIENTES APARECEN COMPLEMENTADOS. EL ELEMENTO DE SOLUCION DE DUPLICIDAD ES LA ULTIMA DIRECCION EN EL CONJUNTO DE DATOS. ESTA PRESENTE CUANDO SE PERMITEN REGISTROS CON LLAVE DUPLICADA, SIN ESPECIFICAR QUE SE ALMACENEN AL PRINCIPIO O AL FINAL.

LA PALABRA DE DIRECCION ABSOLUTA APUNTA A LA TABLA DE INDICES O AL REGISTRO DE DATOS AL QUE CORRESPONDE LA LLAVE.

SI SE TIENEN DATOS EN LA LLAVE, SE ENCUENTRAN AL FINAL, EXCEPTO EN LA ESTRUCTURA SECUENCIAL CON INDICE, EN LA QUE SON SEGUIDOS POR LA DIRECCION ABSOLUTA.

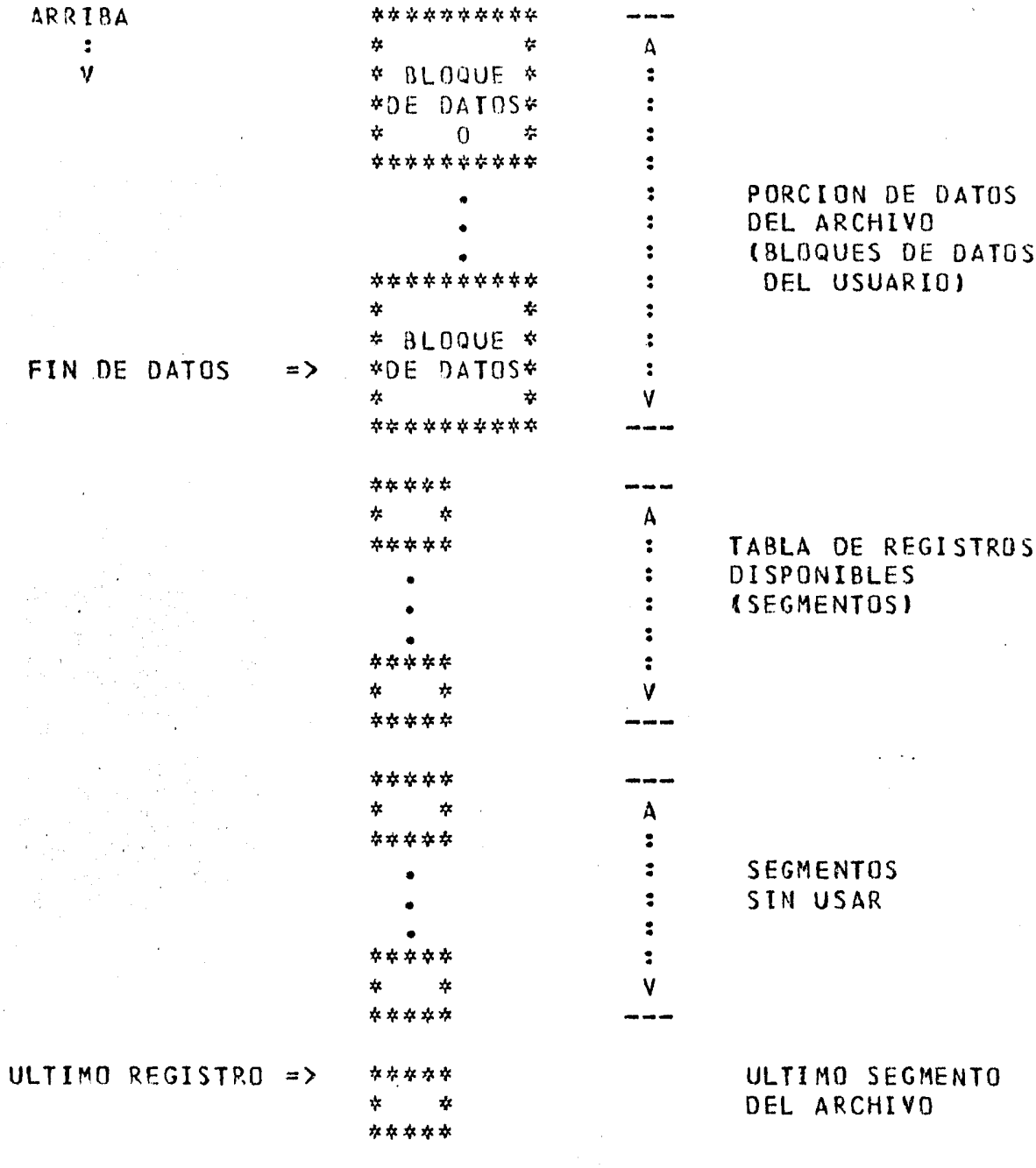
IV.3.2 CONJUNTOS DE DATOS.

EN LAS ESTRUCTURAS DE CONJUNTO DE DATOS TENDREMOS LA INFORMACION QUE TRADICIONALMENTE TENDRIAMOS EN ARCHIVOS. SE OFRECEN DISTINTOS TIPOS, CON PARTICULARIDADES QUE SENALAN SU APLICABILIDAD. POR EJEMPLO, SE DISTINGUEN Y MANEJAN DIFERENTE SI TIENEN O NO FORMATOS VARIABLES ; LOS CONJUNTOS DE DATOS AL AZAR SON PARA RAPIDA RECUPERACION MEDIANTE LLAVE ; LOS CONJUNTOS DE DATOS DIRECTOS SON APLICABLES CUANDO SIRVE DE LLAVE EL NUMERO DE REGISTRO ; Y, LOS CONJUNTOS DE DATOS COMPACTOS SE USAN PARA APROVECHAR AL MAXIMO EL ESPACIO, A COSTA DE TIEMPO

EN LOS CONJUNTOS ESTANDARES DE DATOS, CON O SIN FORMATOS VARIABLES, LOS REGISTROS NO ESTAN ORDENADOS EN LOS BLOQUES. SI EL CONJUNT ES INCLUIDO, LOS REGISTROS EN UN BLOQUE PUEDEN PERTENECER A DIFERENTES PROPIETARIOS, POR LO QUE ENTONCES SE DEBE DE TENER ASOCIADO UN ORDEN AUTOMATICO, Y NO SE PUEDE BUSCAR POR PRIMERO O FINAL, SIGUIENTE O ANTERIOR. LOS BLOQUES NO ESTAN LIGADOS ENTRE SI, Y EL ULTIMO ES DE CONTROL.

SE TIENE UNA LISTA DE ESPACIO DISPONIBLE EN BLOQUES DENTRO DEL CONJUNTO DE DATOS.

CONJUNTOS ESTANDARES DE DATOS, SIN FORMATOS VARIABLES.



ENCONTRAMOS UN GRUPO DE BLOQUES DE DATOS, COMENZANDO CON UN BLOQUE CERO, SEGUIDOS POR UNA TABLA DE REGISTROS DISPONIBLES, SEGMENTOS SIN USAR Y UN BLOQUE ULTIMO.

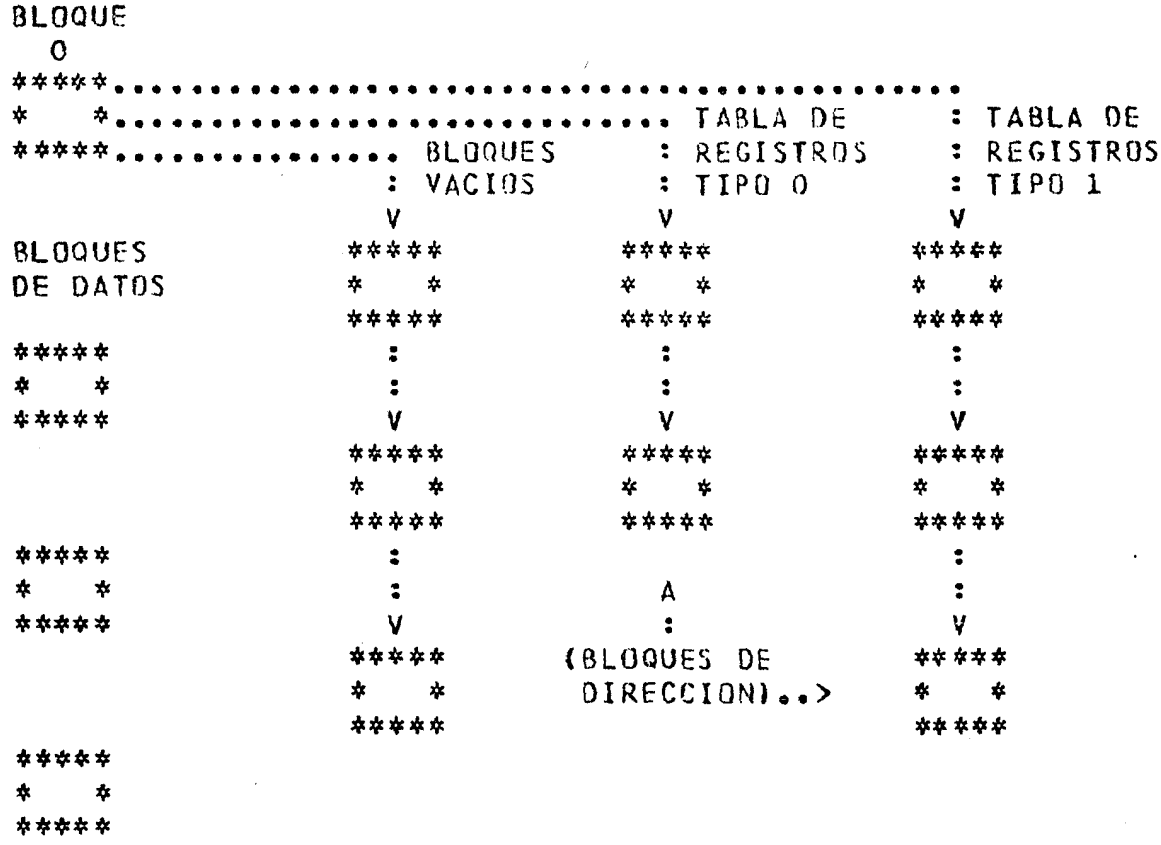
LOS BLOQUES DE DATOS SE COMPONEN DE REGISTROS DE DATOS EN USO Y DADOS DE BAJA. NO SE USA EL REGISTRO CERO EXCEPTO EN EL CASO DE CONJUNTOS DE DATOS DE REINICIO. CADA BLOQUE EMPIEZA EN FRONTERA DE SEGMENTO. LOS REGISTROS EMPIEZAN A FRONTERA DE PALABRA EN LOS BLOQUES.

LA TABLA DE REGISTROS DISPONIBLES CONSTA DE SEGMENTOS, DE LOS CUALES EL PRIMERO, A FRONTERA DE PALABRA, NO CONTIENE INFORMACION, Y LOS SIGUIENTES LLEVAN LA CUENTA DE DIRECCIONES, DE 29, O 29 O MENOS EN EL ULTIMO, Y LAS PROPIAS DIRECCIONES ABSOLUTAS, DE LOS REGISTROS DISPONIBLES.

EN EL ULTIMO REGISTRO SE LLEVAN EL NUMERO DE SEGMENTOS, MENUS DOS, DE LA TABLA DE DISPONIBLES, LA DIRECCION DEL PRIMER REGISTRO A NO USARSE, LA DIRECCION DEL PRIMER SEGMENTO DE LA TABLA DE DISPONIBLES, Y UN INDICADOR DE NINGUN DISPONIBLE.

NO HAY SUMA DE CHEQUEO NI CHEQUEO DE DIRECCION PARA LA TABLA DE DISPONIBLES NI PARA EL SEGMENTO DEL ULTIMO REGISTRO.

CONJUNTOS ESTANDARES DE DATOS CON FORMATOS VARIABLES.



EN ESTA ESTRUCTURA SE TIENEN BLOQUES CERO, DE CONTROL, DE ESPACIO DISPONIBLE EN TABLAS, UNA PARA CADA TIPO DE REGISTRO, BLOQUES VACIOS Y BLOQUES DE DATOS. EL BLOQUE CERO NUNCA CONTIENE PALABRAS DE CHEQUEO DE DIRECCION NI DE SUMA DE CHEQUEO. APUNTA A LOS INICIOS DE LAS TABLAS DE DISPONIBLES Y CONTIENE LA DIRECCION DEL BLOQUE CORRIENTE. LOS BLOQUES DE LOS DEMAS TIPOS SON TODOS DEL MISMO TAMANO.

LOS BLOQUES VACIOS FUERON BLOQUES DE DISPONIBLES QUE FUERON VACIADOS, Y SE TIENEN EN UNA CADENA.

UNA VEZ QUE UN BLOQUE SE USA PARA DATOS, SEGUIRA SIENDO PARA DATOS AUNQUE TODOS LOS REGISTROS SEAN DADOS DE BAJA.

A TIEMPO DE CREACION DE UN REGISTRO, SE BUSCA ESPACIO EN LA TABLA CORRESPONDIENTE AL TIPO. EN CASO DE NO HABER UN ESPACIO DISPONIBLE, SE INTENTA ACOMODARLO AL FIN DEL BLOQUE CORRIENTE. DE NO CABER, SE ENCUENTRA UN BLOQUE VACIO, DE LA CADENA DE BLOQUES VACIOS O DEL FINAL DEL ARCHIVO, Y EN EL SE ALMACENA, PASANDO A SER ESE BLOQUE EL BLOQUE CORRIENTE.

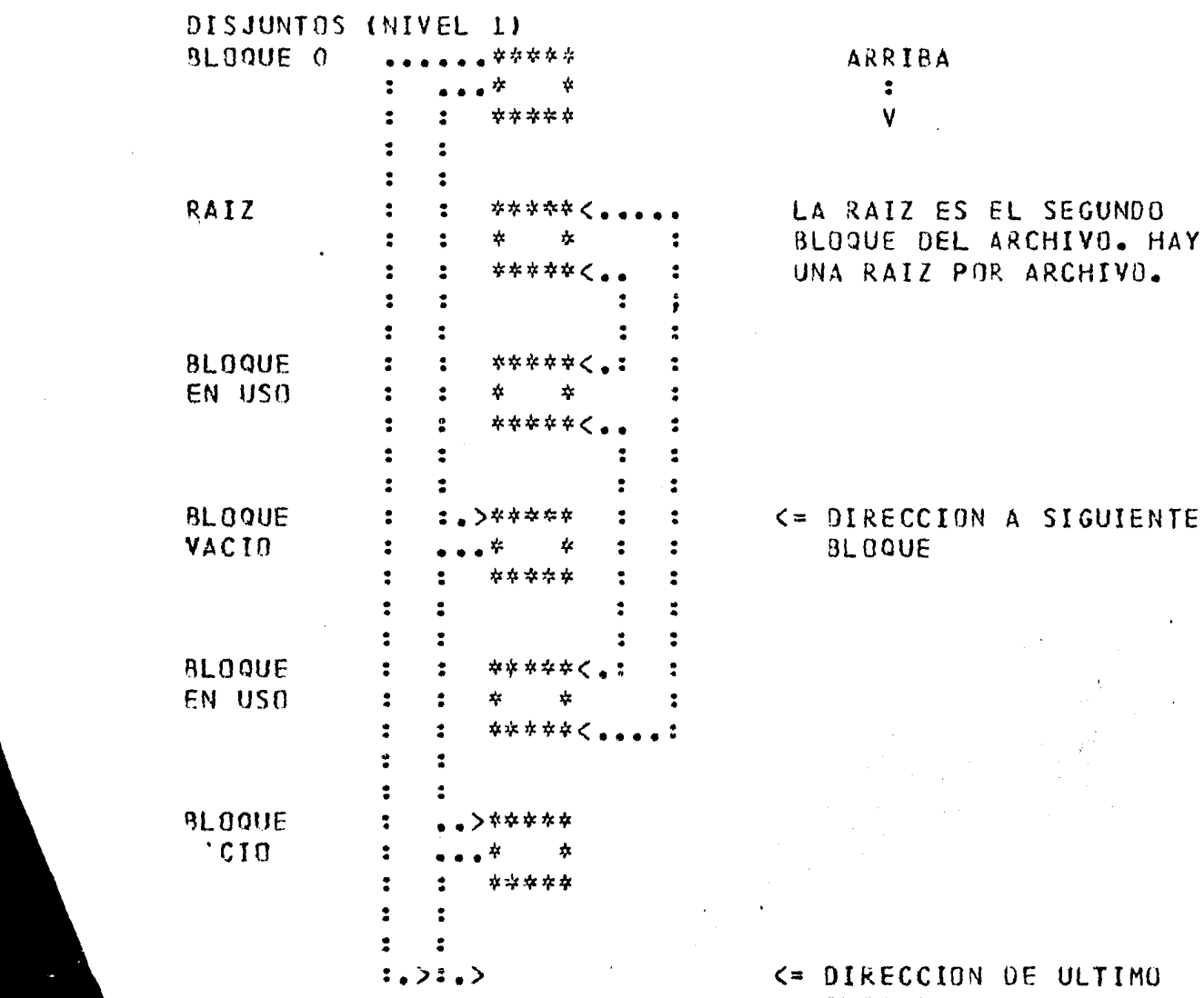
LA PALABRA CERO DE CADA BLOQUE ES DE CONTROL, Y CON ELLA SE DISTINGUEN LOS BLOQUES DE DATOS, LO QUE SE USA EN LAS BUSQUEDAS POR SIGUIENTE. EN BLOQUES DE DATOS APUNTA A LA PRIMERA PALABRA DISPONIBLE. EN BLOQUES DE ESPACIO SENALA EL NUMERO DE DIRECCIONES CONTENIDAS EN EL EL BLOQUE, Y APUNTA AL SIGUIENTE BLOQUE EN LA CADENA.

CONJUNTOS DESORDENADOS DE DATOS.

EN LOS CONJUNTOS DESORDENADOS DE DATOS LOS REGISTROS NO ESTAN ORDENADOS EN LOS BLOQUES ; LOS BLOQUES ESTAN LIGADOS ENTRE SI, Y SE PUEDE TENER ASOCIADO ALGUN ORDEN. EN REGISTROS DE DATOS SE TIENE UNA LISTA DE ESPACIO DISPONIBLE. LAS DOS PRIMERAS PALABRAS DE CADA BLOQUE SON DE CONTROL.

SI EL CONJUNTO DE DATOS ES INCLUIDO, TODOS LOS REGISTROS EN UN BLOQUE PERTENECER A UN MISMO PROPIETARIO.

ESTA ESTRUCTURA ES RECOMENDABLE SOLO CON POCOS REGISTROS.

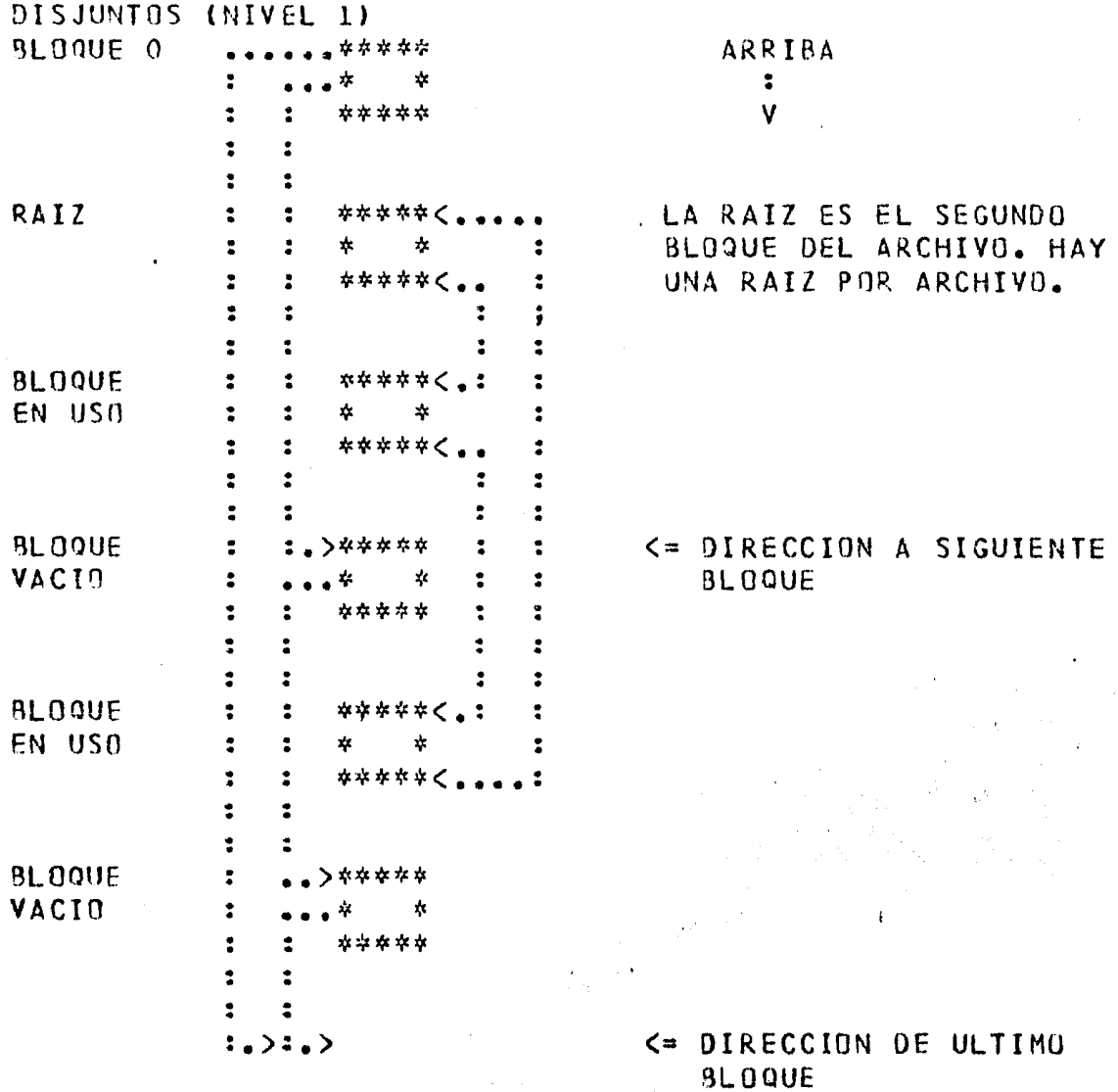


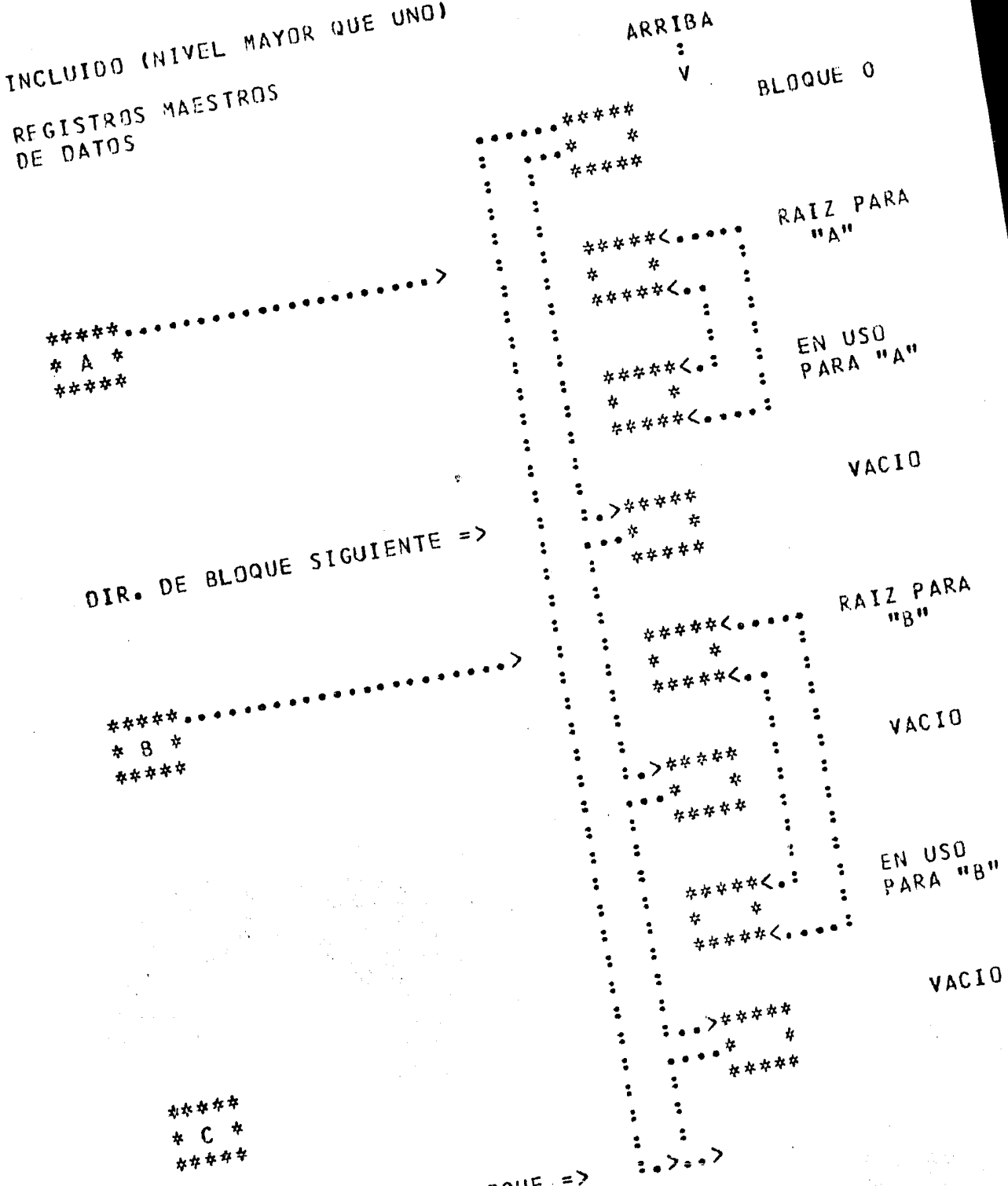
CONJUNTOS DESORDENADOS DE DATOS.

EN LOS CONJUNTOS DESORDENADOS DE DATOS LOS REGISTROS NO ESTAN ORDENADOS EN LOS BLOQUES ; LOS BLOQUES ESTAN LIGADOS ENTRE SI, Y SE PUEDE TENER ASOCIADO ALGUN ORDEN. EN REGISTROS DE DATOS SE TIENE UNA LISTA DE ESPACIO DISPONIBLE. LAS DOS PRIMERAS PALABRAS DE CADA BLOQUE SON DE CONTROL.

SI EL CONJUNTO DE DATOS ES INCLUIDO, TODOS LOS REGISTROS EN UN BLOQUE PERTENECER A UN MISMO PROPIETARIO.

ESTA ESTRUCTURA ES RECOMENDABLE SOLO CON POCOS REGISTROS.

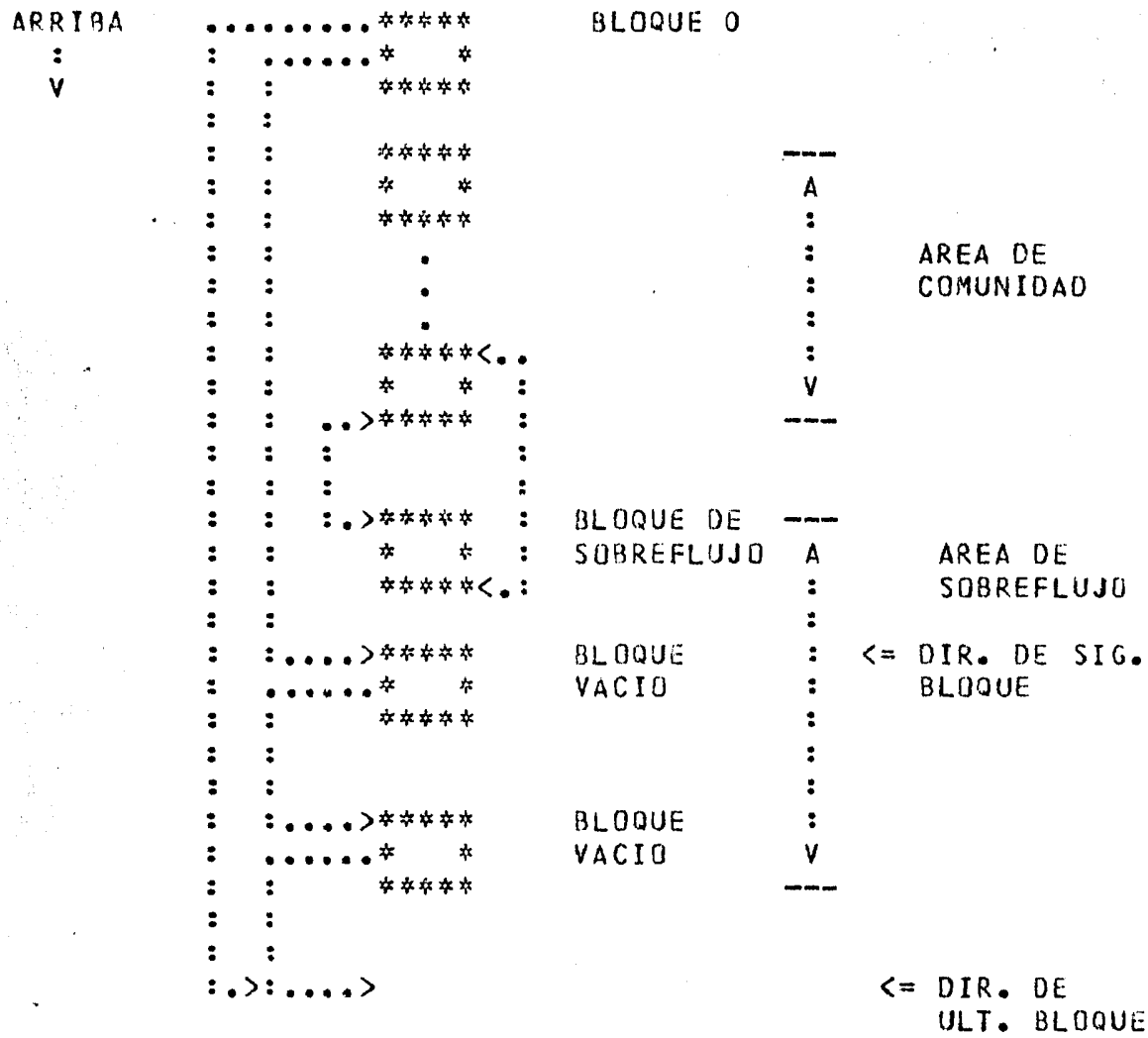



INCLUIDO (NIVEL MAYOR QUE UNO)
REGISTROS MAESTROS
DE DATOS
ARRIBA
V
BLOQUE 0
RAIZ PARA "A"
EN USO PARA "A"
VACIO
A
DIR. DE BLOQUE SIGUIENTE =>
RAIZ PARA "B"
VACIO
B
EN USO PARA "B"
VACIO
C
DIR. DEL ULTIMO BLOQUE =>

EN LOS ARCHIVOS PARA ESTAS ESTRUCTURAS SE TIENEN BLOQUES DE DATOS EN USO, Y AREA DISPONIBLE. LOS BLOQUES EN USO ESTAN FORMADOS EN UNA LISTA DOBLEMENTE LIGADA, CUYA RAIZ EN CONJUNTOS DISJUNTOS ES EL SEGUNDO BLOQUE DEL ARCHIVO, Y A LA CUAL APUNTA EL REGISTRO PROPIETARIO EN CONJUNTOS INCLUIDOS. LOS BLOQUES VACIOS ESTAN LIGADOS ENTRE SI EN UNA LISTA ENCABEZADA POR EL BLOQUE CERO, Y MANTENIDA EN ORDEN DE PRINCIPIO A FIN EN LOS BLOQUES. LAS AREAS DISPONIBLES CONJUNTAS SE JUNTAN.

EN EL CASO DE REGISTROS DE FORMATO VARIABLE, LA LONGITUD DE CADA REGISTRO SE DETERMINA EXAMINANDO SU TIPO, QUE SE ENCUENTRA EN LA PRIMERA PALABRA.

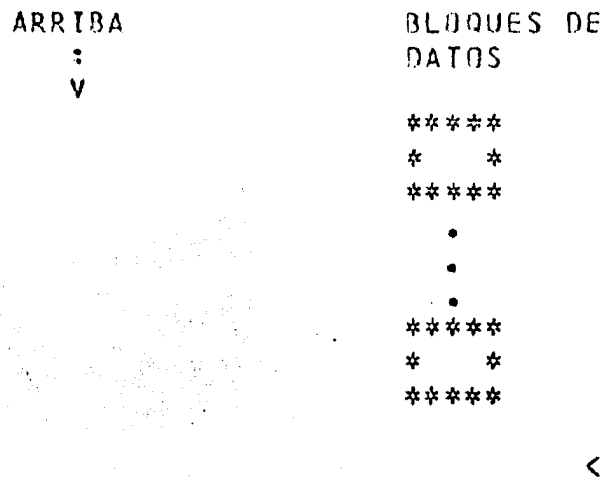
CONJUNTOS DE DATOS AL AZAR.



ESTA ESTRUCTURA SOLO ES PERMITIDA A NIVEL UNO, Y NO SE PUEDEN DECLARAR REGISTROS DE FORMATO VARIABLE. SE TIENEN EN EL ARCHIVO UN AREA DE COMUNIDAD, Y BLOQUES VACIOS Y DE SOBREFLUJO. EL MODULO DE COMUNIDAD ES EL NUMERO DE BLOQUES EN EL AREA DE COMUNIDAD. DADA UNA LLAVE, SE LE APLICA UN ALGORITMO DE DISPERSION, QUE DA UN NUMERO ENTRE UNO Y EL MODULO, INDICANDO EN QUE BLOQUE EN EL AREA BASICA SE LOCALIZARA EL REGISTRO. SI EL BLOQUE EN EL AREA BASICA ESTA LLENO, EL REGISTRO SE ALMACENARA EN UNO DE SUS BLOQUES DE SOBREFLUJO. LOS BLOQUES DE SOBREFLUJO ESTAN ENCADENADOS EN UNA LISTA DOBLEMENTE LIGADA, ENCABEZADA POR EL BLOQUE EN EL AREA BASICA. EL BLOQUE CERO ENCABEZA UNA LISTA LIGADA, A LA QUE ES INCORPORADO CADA BLOQUE DE SOBREFLUJO DEL CUAL SE DEN DE BAJA TODOS LOS REGISTROS.

SI SE BUSCA POR EL SIGUIENTE CON EL ACCESO AL AZAR, ANTES DEL SIGUIENTE BLOQUE EN EL AREA BASICA, LOS BLOQUES DE SOBREFLUJO SON REVISADOS, Y SI ES A TRAVES DEL CONJUNTO DE DATOS, SE TOMAN LOS BLOQUES EN ORDEN FISICO.

CONJUNTOS DE DATOS DIRECTOS.



ESTA ESTRUCTURA SOLO ES PERMITIDA A NIVEL UNO, Y NO SE PUEDEN DECLARAR REGISTROS DE TAMANO VARIABLE. NO SE USA EL REGISTRO CERO, NI SE ALMACENA INFORMACION DE CONTROL. LA LLAVE DE ACCESO Y ALMACENAMIENTO ES EL NUMERO RELATIVO DE REGISTRO. SI SE TRATA DE ALMACENAR UN REGISTRO ADELANTE DEL FIN DE ARCHIVO, SE CAUSA QUE EL ARCHIVO SE LLENE DE BLOQUES MARCADOS COMO DADOS DE BAJA.

CONJUNTO DE DATOS DE REINICIO.

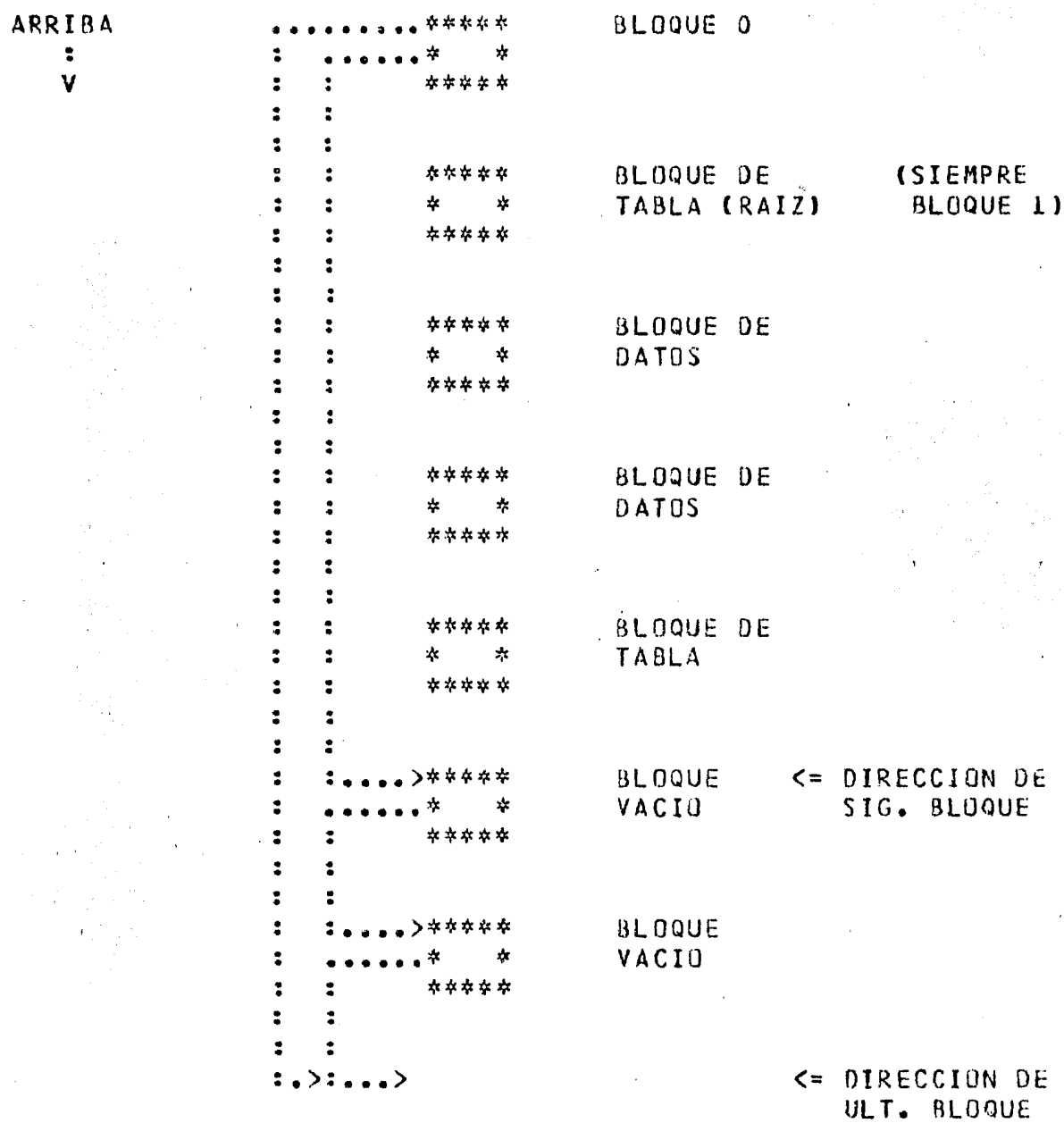
ESTA ESTRUCTURA ES CASI IDENTICA A LA DE CONJUNTOS DE DATOS ESTANDARES SIN FORMATO VARIABLE. EN ELLA SE TIENE INFORMACION DE TRAZA DE MODIFICACIONES. EL ABORTO DE TRANSACCIONES Y LA RECUPERACIUN DE PARADO- ARRANCADO PUEDEN ALMACENARLE REGISTROS AUTOMATICAMENTE, Y EL USUARIO PUEDE TAMBIEN ALMACENAR Y DAR DE BAJA REGISTROS EXPLICITAMENTE. SOLO PUEDE HABER UNA DE ESTAS ESTRUCTURAS POR BASE DE DATOS, Y NO SE LE PUEDEN DECLARAR ITEMS DE CONTROL, COMO LIGAS O ORDENES INCLUIDOS.

SE DECLAREN O NO, SIEMPRE SE INCLUIRAN UN ITEM DE TIPO DE REINICIO EN EL PRIMER DIGITO DE LA SEGUNDA PALABRA, Y UNO DE CUENTA DE TRANSACCIONES, EN LA PRIMERA PALABRA. LA CUENTA DE TRANSACCIONES ES UN NUMERO REAL, QUE SE INCREMENTA EN UNO POR LAS RUTINAS DE ACCESO EN LA FASE FINAL DE END-TRANSACTION. EL TIPO DE REINICIO IDENTIFICA SI EL REGISTRO FUE CREADO POR EL USUARIO, SI ES LA ULTIMA AREA BUENA PARA "BEGIN-TRANSACTION" O PARA "END-TRANSACTION". ESTOS REGISTROS SON ENCONTRADOS POR SIGUIENTE, Y PUEDEN SER ALMACENADOS EXPLICITAMENTE POR EL USUARIO. OTROS TIPOS INCLUYEN, EN USO POR EL SISTEMA COMO ALMACENAMIENTO TEMPORAL DE LA ULTIMA AREA BUENA PARA BEGIN-TRANSACTION, PARA END- TRANSACTION, Y ESPACIO DISPONIBLE PARA REGISTRO.

EL TAMANO DE BLOQUE PARA UN CONJUNTO DE DATOS DE REINICIO DEBE DE SER DE AL MENOS 30 PALABRAS.

PARA GARANTIZAR QUE AL MENOS SE DISPONDRA DE UN SEGMENTO PARA INFORMACION DE CONTROL DE TRAZA, LA TOTALIDAD DEL BLOQUE CERO SE RESERVA PARA USO DEL SISTEMA. EN EL SE TIENE UN INDICADOR DE BASE DE DATOS CERRADA NORMALMENTE, EN USO NORMAL O SIENDO RECONSTRUIDA, EL NUMERO CORRIENTE DE ARCHIVO DE TRAZA, EL NUMERO DE SERIE DEL BLOQUE CORRIENTE DE TRAZA, UN INDICADOR DE TRAZA ACTIVA, E INFORMACION DE REINICIO CUANDO SE ESTA RECONSTRUYENDO LA BASE.

CONJUNTOS DE DATOS COMPACTOS.



EN ESTA ESTRUCTURA LOS REGISTROS SE PUEDEN SER EFECTIVAMENTE DE LONGITUD VARIABLE, SEA POR FORMATOS VARIABLES, ITEMS DE TAMANO VARIABLE, ITEMS DE GRUPO CON OCURRENCIA VARIABLE, ITEMS NO SIEMPRE PRESENTES, O POR COMBINACIONES DE ESTAS POSIBILIDADES. SE TIENE UN TAMANO MAXIMO DE REGISTRO, Y EL TAMANO REAL PUEDE SER HASTA DE SOLO UNA PALABRA.

ESTOS CONJUNTOS PUEDEN SER INCLUIDOS, PERO TAMPOCO EN ESTE CASO SE TIENE EN LOS REGISTROS ASOCIACION CON SUS PROPIETARIOS, Y POR LO TANTO NO SE PERMITEN BUSQUEDAS POR PRIMERO, ULTIMO, SIGUIENTE Y ANTERIOR, Y SE OBLIGA A TENER ASOCIADO UN ORDEN AUTOMATICO.

SE CUENTA CON UN BLOQUE CERO, CON UNA LISTA LIGADA DE BLOQUES VACIOS, CON BLOQUES DE TABLAS DE DATOS. LOS BLOQUES DE DATOS CONTIENEN UN DIRECTORIO DE REGISTROS, CON UNA PALABRA POR CADA REGISTRO ACOMODADO EN EL BLOQUE, CONTENIENDO LONGITUD Y LOCALIZACION. MIENTRAS SE TENGA ESE REGISTRO SE TENDRA LA ENTRADA EN EL DIRECTORIO, AUN CUANDO SE MUEVA EN EL BLOQUE AL CRECER. AL CAMBIAR DE BLOQUE POR NO CABER MAS, LA ENTRADA EN EL BLOQUE ORIGINAL SE REFIERE AL NUEVO BLOQUE, Y AL NUEVO NUMERO DE REGISTRO.

AL DARSE DE BAJA UN REGISTRO SU ENTRADA EN EL DIRECTORIO SE MARCA, Y SE PONEN DISPONIBLES LOS ESPACIOS PARA EL REGISTRO Y PARA EL DIRECTORIO. LOS REGISTROS SON APUNTADOS POR DIRECCIONES ABSOLUTAS, CON EL CAMPO DE DIRECCION DE PALABRA SIGNIFICANDO NUMERO DE REGISTRO, Y LOCALIZANDO UNA ENTRADA EN EL DIRECTORIO QUE A SU VEZ LOCALIZA AL REGISTRO O SENALA UN REGISTRO DADO DE BAJA.

EN LOS BLOQUES DE TABLA SE MANTIENEN TABLAS SECUENCIALES CON INDICE, CON PALABRAS SENALANDO LOS ESPACIOS DISPONIBLES CON DIRECCION DE BLOQUE PARA CADA BLOQUE. SI SE NECESITA ESPACIO PARA UN REGISTRO, SE HACE UNA BUSQUEDA BINARIA HASTA ENCONTRAR EL BLOQUE CON EL ESPACIO MAS PEQUENO QUE BASTE, CAMBIANDOSE LAS ENTRADAS CADA VEZ QUE CAMBIEN.

AL PRINCIPIO SE OBTIENEN LOS BLOQUES DE DATOS Y TABLAS AL FINAL DEL ARCHIVO, Y DESPUES SE ASIGNAN DE LA LISTA DE ESPACIO, QUE SE CONSTRUYE A MEDIDA QUE SE VACIAN BLOQUES.

EN EL BLOQUE CERO SE TIENEN LAS DIRECCIONES DE BLOQUE DEL SIGUIENTE DISPONIBLE, HACIENDO DE ESTE BLOQUE LA CABEZA DE LA LISTA. TAMBIÉN APUNTA AL PRIMER BLOQUE A NO SER USADO, PARA CHEQUEO DE LIMITE.

EN UN BLOQUE DE DATOS ENCONTRAMOS UNA PALABRA DE CONTROL, CON EL NUMERO DE ENTRADAS DE DIRECTORIO DISPONIBLES, EL NUMERO DE ENTRADAS DE DIRECTORIO OCUPADAS, Y EL NUMERO DE PALABRAS DE ESPACIO DISPONIBLE PARA DATOS ; UN NUMERO DE SERI DE TABLA, UN INDICADOR DE TIPO DE BLOQUE, LAS ENTRADAS DE DIRECTORIO, ESPACIO DISPONIBLE Y DATOS.

UNA ENTRADA DE DIRECTORIO CONTIENE PARA EL REGISTRO, SU TAMANO, LOCALIZACION, E INDICADORES DE SI ESTE ES EL BLOQUE ORIGINAL, DE SI ESTA PRESENTE EN ESTE BLOQUE, Y DE ENTRADA EN USO O DISPONIBLE.

EL FORMATO DE UN BLOQUE DE TABLA PARA ESTA ESTRUCTURA ES CASI IDENTICO AL DE LAS TABLAS DE SECUENCIAL CON INDICE. LA PALABRA DOS ES EL TIPO DE BLOQUE, Y LAS ENTRADAS DE LLAVE PUEDEN EMPEZAR EN LA PALABRA 3.

UN BLOQUE DE ESPACIO DISPONIBLE SOLO LLEVA UNA LIGA AL SIGUIENTE BLOQUE DISPONIBLE, Y UN INDICADOR DE TIPO DE BLOQUE.

CONJUNTO DE DATOS ORDENADO.

EN ESTA ESTRUCTURA, EL PRIMER BLOQUE ES LA CABEZA DE LISTA DE ESPACIO Y SUBBLOQUES DISPONIBLES, EL SEGUNDO ES EL BLOQUE RAIZ SI ES DE NIVEL UNO, Y LOS SIGUIENTES SON BLOQUES EN USO, DISPONIBLES Y DE TABLAS. LA INFORMACION EN LOS BLOQUES SE ACOMODA ASI : EN EL PRIMER BLOQUE, DE CABEZA DE LISTA DE SUBBLOQUES DISPONIBLES, Y DE ESPACIO DISPONIBLE, LAS DOS PRIMERAS PALABRAS SON CEROS, LA SIGUIENTE LLEVA LA DIRECCION DEL ULTIMO BLOQUE DESASIGNADO, LA NUMERO TRES LA DIRECCION DEL PRIMER BLOQUE DESPUES DEL FIN DE ARCHIVO.

HAY UN TRATAMIENTO APARTE PARA INCLUIDOS SI SE ESPECIFICA TAMANO DE SUBBLOQUE. BLOQUE DEJADO DE LADO, DEFINICION DE EMPIEZO DE SB=5, Y PALABRAS 6-N, DIRECCION DE TABLA O VACIO. EN LOS BLOQUES DE DATOS, LA PALABRA CERO LLEVA LA LOCALIDAD DEL SIGUIENTE BLOQUE. SI IGUAL A SU PROPIA DIRECCION ES QUE ES ULTIMO BLOQUE DE CADENA DE SOBREFLUJO, SI NO, ES EL NUMERO DE BLOQUE DEL SIGUIENTE EN LA CADENA. TAMBIEN LLEVA TIPO DE BLOQUE, 1. LA PALABRA UNO LLEVA EL NUMERO DE SERIE DE TABLA. SI EL VALOR DE LA PALABRA DOS ES IGUAL A SU PROPIA DIRECCION, ES QUE ES PRIMER BLOQUE DE CADENA DE SOBREFLUJO, SI NO, ES EL NUMERO DEL BLOQUE PREVIO EN LA CADENA. TAMBIEN LLEVA EL NUMERO DE SUBBLOQUES EN ESTE BLOQUE, SIEMPRE UNO PARA DISJUNTOS O PARA INCLUIDOS CUANDO EL PROPIETARIO CONTIENE MAS, ENTONCES TAMANO DE BLOQUE DE REGISTROS DE DETALLE.

LA PALABRA TRES LLEVA PALABRAS DE CONTROL DE BLOQUE, CON CORRIMIENTO DEL PRIMER REGISTRO EN EL SUBBLOQUE, CORRIMIENTO DEL ULTIMO REGISTRO EN EL SUBBLOQUE, O CERO SI EL BLOQUE ESTA VACIO, CORRIMIENTO DEL ESPACIO DEL ULTIMO REGISTRO EN EL SUBBLOQUE, O CERO SI EL ESPACIO ESTA COMBINADO CON EL SIGUIENTE BLOQUE ADYACENTE.

LOS REGISTROS EN LOS BLOQUES Y SUBBLOQUES ENTRAN EN ORDEN INVERSO, ESTO ES, SECUENCIALMENTE DEL FIN DEL BLOQUE O SUBBLOQUE HACIA EL PRINCIPIO.

EL FORMATO DEL REGISTRO ES EN DOS PARTES, AREA DE DATOS Y AREA DE LLAVES. EL AREA DE LLAVES NO EXISTE SI TODOS LOS ITEMS DE LLAVE SON ASCENDENTES, LAS LLAVES SON NUMEROS SIN SIGNO O VARIABLES ALFABETICAS, Y SI TODOS LOS ITEMS DE LLAVE ESTAN JUNTOS EN EL AREA DE REGISTRO. ESTO ES, EXISTE SI CUALQUIERA LLAVE ES DESCENDENTE, TIENE SIGNO, ES REAL O LAS LLAVES NO ESTAN JUNTAS EN EL AREA DE REGISTRO.

NO SE PERMITEN GRUPOS COMO LLAVES.

ESQUEMA DE BLOQUES DE DATOS. SI EL CONJUNTO ES DISJUNTO, O INCLUIDO CON TAMANO DE SUBBLOQUE NO ESPECIFICADO, O ESTE ES MAYOR QUE EL TAMANO DEL BLOQUE, SE TIENE ESTE FORMATO, QUE TAMBIEN EXISTE PARA BLOQUES DE CONJUNTOS DE DATOS ORDENADOS INCLUIDOS CUANDO EL NUMERO DE REGISTROS DE DETALLE PARA UN PROPIETARIO DADO NO ES MENOR QUE EL TAMANO DEL BLOQUE. PARA ESTE CASO EL REGISTRO PROPIETARIO CONTIENE LA DIRECCION DEL PRIMER BLOQUE DE DETALLE EN LA LISTA DE BLOQUES LIGADOS. PARA EL DISJUNTO, SE TIENE EL APUNTADOR EN EL BLOQUE NUMERO UNO.

SI UN REGISTRO ES ANADIDO A UN BLOQUE Y ESTE ESTA LLENO, ENTONCES : SE INSERTA UN NUEVO BLOQUE ENTRE LOS BLOQUES N Y N+1, AJUSTANDOSE LAS LIGAS A SIGUIENTE Y ANTERIOR, Y LOS REGISTROS EN EL BLOQUE N SON PARTIDOS ASI : SI EL REGISTRO A SER ANADIDO ESTA EN EL ULTIMO 10 % DEL BLOQUE, ENTONCES ESTE 10 % DE REGISTROS SON MOVIDOS AL BLOQUE N+1. SI ESTA EN EL PRIMER 10 % DEL BLOQUE, SE MUEVE EL 90 % ULTIMO AL BLOQUE N+1. PARA LOS DEMAS CASOS SON MOVIDOS AQUELLOS REGISTROS ADELANTE EN LA SECUENCIA DEL REGISTRO SIENDO ANADIDO, COLOCANDOSE ENTONCES AL NUEVO REGISTRO EN EL BLOQUE N O EN EL N+1, DEPENDIENDO DE LA SEPARACION.

SI EN REGISTRO ES DADO DE BAJA DE UN BLOQUE, TODOS LOS REGISTROS SUBSECUENTES SON MOVIDOS A LLENAR EL HUECO. NO SE CONSOLIDAN BLOQUES AL DAR DE BAJA, AUNQUE SE PUEDE CORRER REORGANIZACION PERIODICAMENTE, PARA RECUPERAR ESPACIO NO USADO.

EN INCLUIDOS, CON TAMANO DE SUBBLOQUE ESPECIFICADO, EL ESQUEMA PARA BLOQUES ES COMO SIGUE. SE TIENEN APUNTADOR A SIGUIENTE, TIPO DE BLOQUE, NUMERO DE SERIE DE TABLA, APUNTADOR A ANTERIOR Y NUMERO DE SUBBLOQUES, Y PALABRAS DE CONTROL DE BLOQUE PARA CADA SUBBLOQUE. LOS SUBBLOQUES SON APUNTADOS POR LOS REGISTROS PROPIETARIOS.

LA IMPLEMENTACION ES COMO SIGUE : CADA BLOQUE ESTA DIVIDIDO EN UNO O MAS SUBBLOQUES. UNA PALABRA DE CONTROL DE BLOQUE CONTIENE INFORMACION DE LA LOCALIZACION Y ESPACIO USADO EN UN SUBBLOQUE. EL NUMERO DE REGISTROS EN CADA SUBBLOQUE SERA UN MULTIPLO DEL TAMANO DE SUBBLOQUE. JUSTO DESPUES DE CARGA INICIAL DE LA BASE, O DE REORGANIZACION, SUBBLOQUES ADYACENTES PERTENECERAN A REGISTROS MAESTROS ADYACENTES. UNA OPERACION DE ENTRADA/SALIDA QUE CARGUE DETALLES PARA UN PROPIETARIO PUEDE CARGAR DETALLES DE PROPIETARIOS ADYACENTES, AHORRANDO OPERACIONES DE ENTRADA/SALIDA. SI SON ANADIDOS NUEVOS REGISTROS DE DETALLE DE FORMA TAL QUE NO HAYA ESPACIO SUFICIENTE EN EL SUBBLOQUE CORRIENTE, ENTONCES TODOS LOS REGISTROS EN EL SUBBLOQUE SON MOVIDOS AL SUBBLOQUE SIGUIENTE MAYOR EN EL BLOQUE CORRIENTE O EN ALGUN OTRO. SI EL NUMERO DE REGISTROS DE DETALLE EXCEDE AL NUMERO ESPECIFICADO PARA TAMANO DEL BLOQUE ENTONCES UN SUBLOQUE SE HACE AL TAMANO DE UN BLOQUE CON BLOQUES DE SOBREFLUJO SIGUIENTE Y ANTERIOR LIGADOS.

BLOQUE DE TABLA. LA PALABRA CERO LLEVA LIGA A OTRO BLOQUE DE TABLA, Y TIPO DE BLOQUE, 2. LA PALABRA UNO ES EL NUMERO DE SERIE DE TABLA. LA DOS ES EL INDICE DE LA ULTIMA ENTRADA VALIDA EN LA TABLA, Y LAS SIGUIENTES LLEVAN NUMERO DE BLOQUE DE ALGUNO CON SUBBLOQUE VACIO, E INDICE DE PALABRA DE CONTROL DE BLOQUE DE SUBBLOQUE VACIO.

LO SIGUIENTE EXISTE SOLO CUANDO EL CONJUNTO DE DATOS ORDENADO ES INCLUIDO Y SE ESPECIFICA TAMANO DE BLOQUE.

LOS SUBBLOQUES PUEDEN SER DE DIFERENTES TAMANOS EN INCREMENTOS DEL TAMANO ESTABLECIDO. UN TIPO CONCEPTUAL DE SUBBLOQUE SE CALCULA ASI : SBTYPE = (BCW.IDFIRSTREC - BCW.ODLASTAVAIL) DIV RECORDSIZE, O SEA LA DIVISION ENTERA DE LA DIFERENCIA ENTRE EL PRIMER REGISTRO Y EL ULTIMO DISPONIBLE, ENTRE EL TAMANO DE REGISTRO.

CUANDO SE DA SOBREFLUJO DE UN SUBBLOQUE DE UN TAMANO DADO, TODOS LOS REGISTROS SON MOVIDOS A OTRO SUBBLOQUE DEL SIGUIENTE TAMANO MAYOR, PONIENDOSE LA LOCALIDAD DEL SUBBLOQUE AHORA VACIO EN LAS TABLAS DE SUBBLOQUES DISPONIBLES.

DADO UN SUBBLOQUE DE TIPO, LA PALABRA EN EL BLOQUE CERO APUNTADA POR LA SUMA DEL EMPIEZO Y DEL TIPO DEL SUBBLOQUE TIENE CERO SI NO EXISTEN SUBBLOQUES DE ESTE TIPO, O LLEVA LA DIRECCION DE BLOQUE DE UNA CADENA DE BLOQUES DE TABLA QUE A SU VEZ LOCALIZAN SUBBLOQUES DISPONIBLES DE ESTE TIPO.

EL EMPIEZO SE DEFINE COMO 5.

SE HACEN ENTRADAS A LAS TABLAS CUANDO TODOS LOS REGISTROS EN UN SUBBLOQUE HAN SIDO DADOS DE BAJA, O CUANDO HAN SIDO MOVIDOS A UN SUBBLOQUE MAYOR. SE REMUEVEN ENTRADAS DE UNA TABLA CUANDO SE CREA UN NUEVO REGISTRO PROPIETARIO Y ENTONCES SE ALMACENA UN PRIMER REGISTRO DE DETALLE, O CUANDO SE NECESITA UN SUBBLOQUE MAYOR POR SOBREFLUJO.

UN BLOQUE DISPONIBLE LLEVA EN LA PALABRA CERO UNA LIGA A OTRO DISPONIBLE, Y TIPO DE BLOQUE, 3. EN LA PALABRA UNO LLEVA NUMERO DE SERIE DE TABLA. SE LIGAN LOS BLOQUES DISPONIBLES.

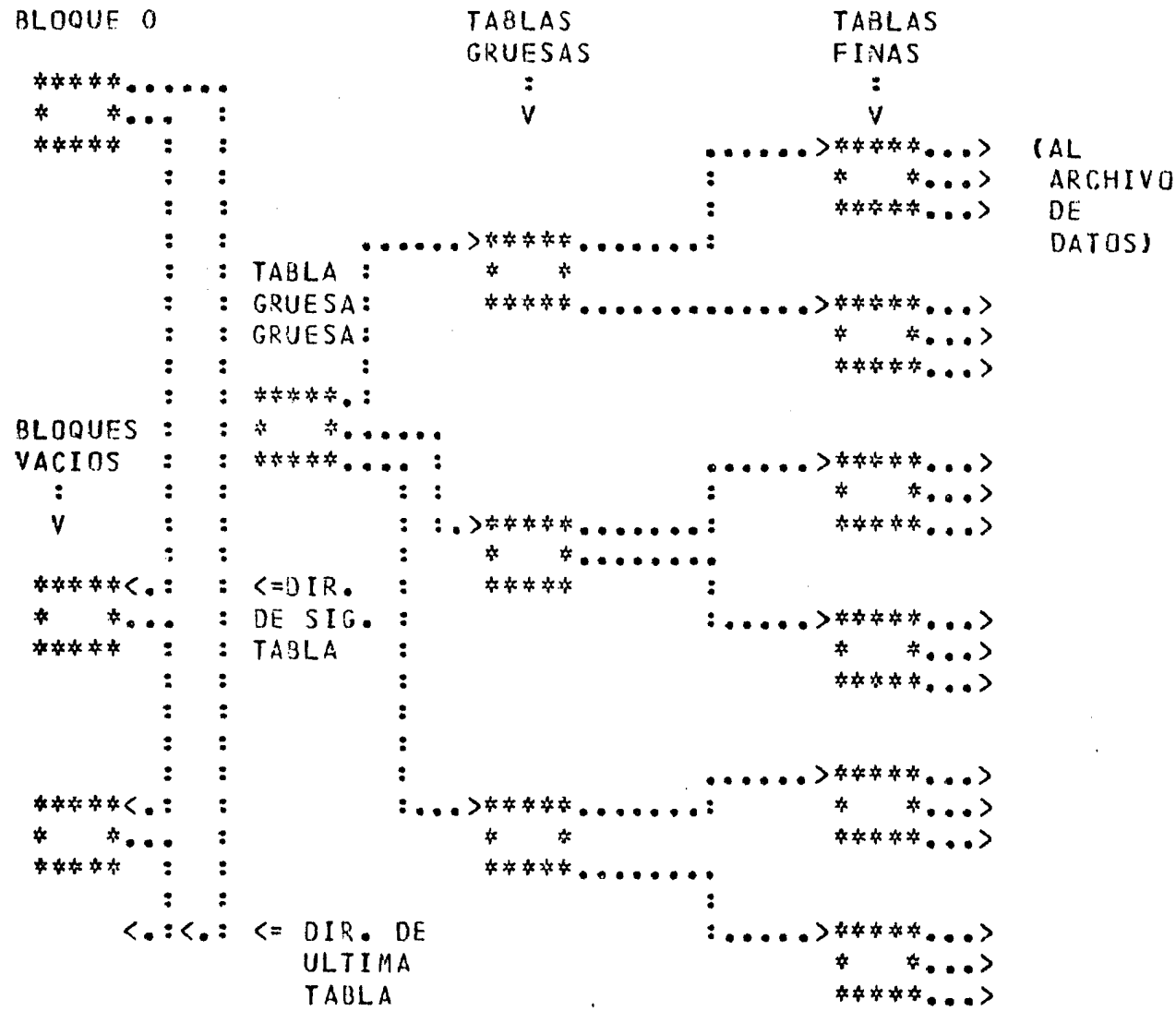
IV.3.3 ORDENES.

LOS ORDENES SON ESTRUCTURAS PARA RECUPERACION DE INFORMACION DE LOS CONJUNTOS DE DATOS. SE TIENEN SECUENCIAL CON INDICE, CON TABLAS GRUESAS Y FINAS ; LISTA ORDENADA, QUE ES MANTENIDA ORDENADA POR EL SISTEMA ; LISTA DESORDENADA, PARA RECUPERAR DE UN CONJUNTO DE DATOS EN ORDEN FISICO HACIA ADELANTE DESDE EL PRINCIPIO, O HACIA ATRAS DESDE EL FINAL ; AL AZAR CON INDICE, DONDE SE TIENEN LAS LLAVES COMPRIMIDAS Y DISPERSAS EN BLOQUES BASICOS Y DE SOBREFLUJO, Y VECTORES DE DIGITOS BINARIOS, PARA INDICAR PERTENENCIA A UNA CATEGORIA.

SECUENCIAL CON INDICE.

ESTA ESTRUCTURA ES UNA COLECCION DE TABLAS FINAS, GRUESAS Y MAS GRUESAS, EN DONDE LAS ENTRADAS DE LAS GRUESAS APUNTAN A LAS FINAS, HASTA APUNTAR LAS MAS FINAS AL CONJUNTO DE DATOS. LAS ENTRADAS SE MANTIENEN EN ORDEN ASCENDENTE.

CUANDO SE LLENA UNA TABLA SE CONSTRUYE OTRA, A LA QUE SE LLEVA UNA PROPORCION DE ENTRADAS DE ACUERDO CON EL FACTOR DE CARGA.



EN ESTA ESTRUCTURA, EL BLOQUE CERO EMPIEZA UNA LISTA LIGADA DE BLOQUES DISPONIBLES, DE TABLAS CON TODAS LAS ENTRADAS DADAS DE BAJA, Y APUNTA AL FIN DEL ARCHIVO. INICIALMENTE SE TIENE SOLO UNA TABLA FINA, Y CRECE LA ESTRUCTURA A TABLAS GRUESAS Y MAS GRUESAS, HASTA UN NIVEL TEORICO DE 22, QUE EN LA PRACTICA NO PASA DE CUATRO.

PARA MANTENER EN ORDEN ASCENDENTE LAS TABLAS, SON COMPLEMENTADOS AQUELLOS ITEMS DECLARADOS EN ORDEN DESCENDENTE. LA RECUPERACION MAS EFICIENTE ES SECUENCIAL, Y LA RECUPERACION SE HACE EFICIENTE MEDIANTE BUSQUEDAS BINARIAS. EN LAS TABLAS SE USAN ENTRADAS FALSAS, CON EL VALOR MAXIMO REPRESENTABLE, QUE NO SON DADAS DE BAJA, LO QUE HACE QUE EL NIVEL DE TABLAS NUNCA SE REDUZCA.

PARA ORDENES DISJUNTOS, EL SEGUNDO BLOQUE DEL ARCHIVO ES LA TABLA DE MAYOR NIVEL. CADA PROPIETARIO DE UN ORDEN NO VACIO APUNTA A SU TABLA DEL MAYOR NIVEL, QUE NUNCA SE MUEVE.

EL OPERADOR DE REEMPLAZO DA INDICES SOLO EN UN SENTIDO, POR LO QUE LAS ENTRADAS DE LLAVE "FLOTAN" SEGUN PERMITE LA PALABRA DE CONTROL DE TABLA. UNA VEZ QUE LAS ENTRADAS ALCANZAN EL PRINCIPIO, SON MOVIDAS HACIA ABAJO, HACIA EL FIN DE LA TABLA. LA MANERA MAS RAPIDA DE CARGAR EL ARCHIVO, DADO QUE LAS ENTRADAS SON MANTENIDAS ORDENADAS, ES EN ORDEN DE MAYOR A MENOR LLAVE.

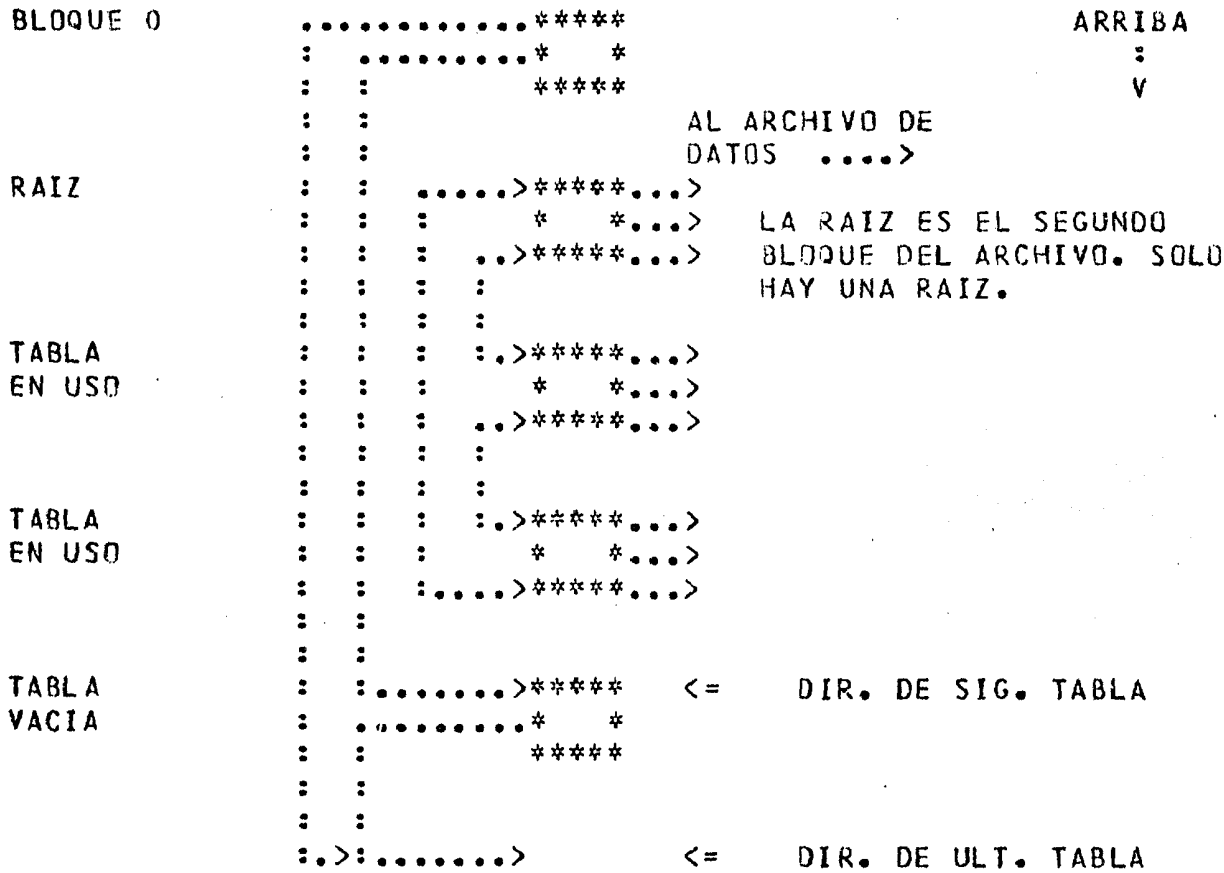
UNA TABLA LLEVA PALABRA DE CONTROL, NUMERO DE SERIE, ENTRADAS DE LLAVE NO USADAS, ENTRADAS DE LLAVE EN USO EN ORDEN ASCENDENTE, Y ENTRADAS NO USADAS. LA PALABRA DE CONTROL APUNTA AL PRINCIPIO Y DA EL NUMERO DE LAS ENTRADAS EN USO, E INDICA SI ES UNA TABLA GRUESA O FINA, ESTO ES SI LA DIRECCION ABSOLUTA ES A OTRA TABLA O AL ARCHIVO DE DATOS. LAS ENTRADAS DE LLAVE ESTAN ALINEADAS A FRONTERA DE PALABRA. EL NUMERO DE SERIE DE LA TABLA SE USA PARA CORDINAR EL ESTADO CORRIENTE DE LA TABLA CON LA TRAZA DE MODIFICACIONES.

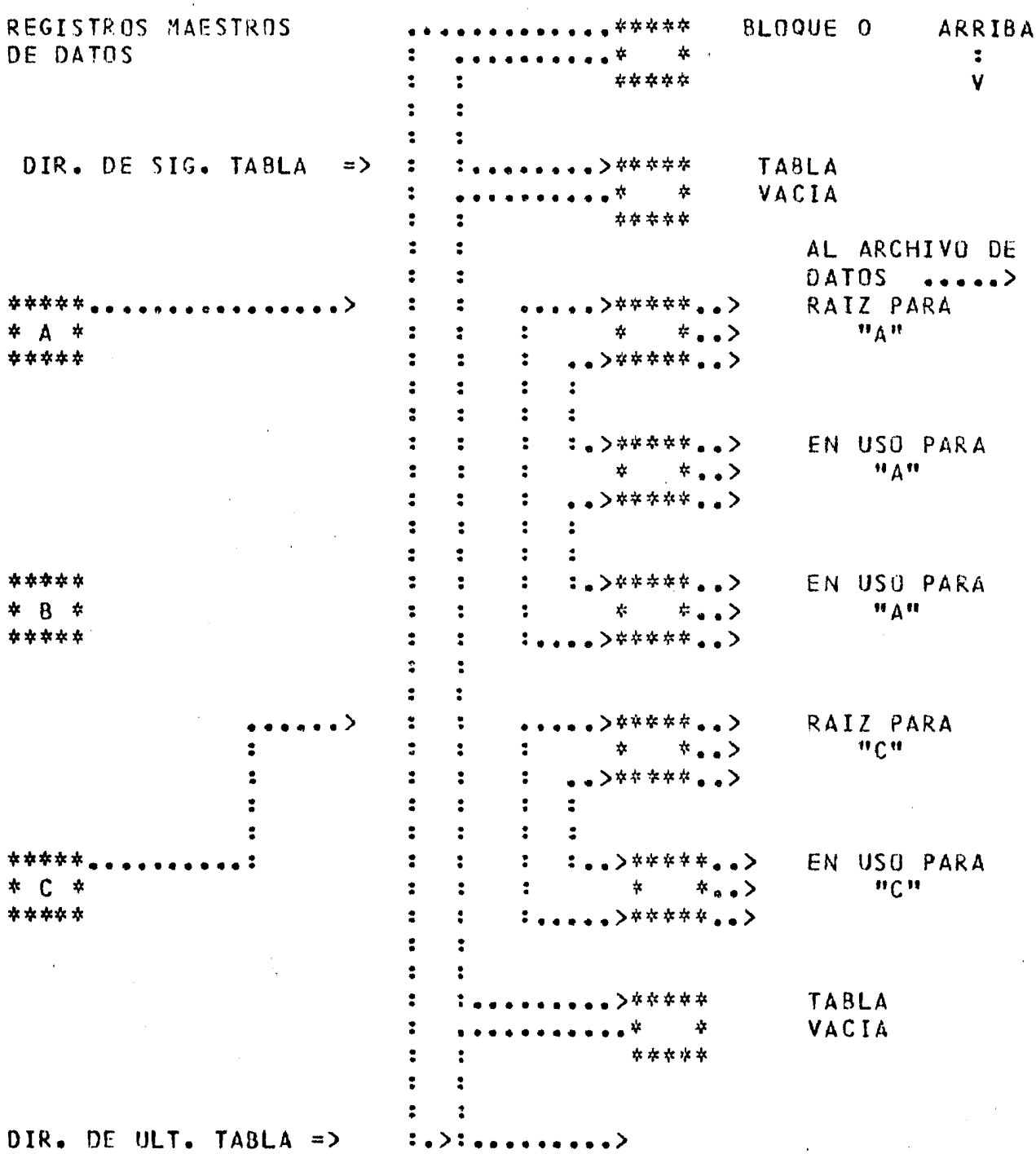
LISTA ORDENADA.

LAS ENTRADAS EN UNA TABLA SE MANTIENEN ORDENADAS SEGUN LLAVE, Y SI LA LISTA ESTA ASOCIADA CON UN CONJUNTO DE DATOS INCLUIDO, TODAS PERTENECEN A UN MISMO PROPIETARIO.

CUANDO SE LLENA UNA TABLA SE CONSTUYE OTRA, A LA QUE SE LLEVA UNA PROPORCION DE ENTRADAS DE ACUERDO AL FACTOR DE CARGA.

DISJUNTA (NIVEL 1)




INCLUIDA (NIVEL MAYOR QUE UNO)
REGISTROS MAESTROS DE DATOS
BLOQUE 0
ARRIBA
DIR. DE SIG. TABLA =>
TABLA VACIA
AL ARCHIVO DE DATOS
RAIZ PARA "A"
EN USO PARA "A"
EN USO PARA "A"
RAIZ PARA "C"
EN USO PARA "C"
TABLA VACIA
DIR. DE ULT. TABLA =>
A
B
C

EN ESTA ESTRUCTURA, LAS TABLAS EN USO CONSTITUYEN UNA LISTA DOBLEMENTE LIGADA, ENCABEZADA POR EL SEGUNDO BLOQUE SI ES DISJUNTO, O POR UN BLOQUE APUNTADO POR CADA REGISTRO PROPIETARIO, CUYA DIRECCION NO CAMBIA UNA VEZ ASIGNADA. LA TABLA RAIZ SIEMPRE CONTIENE LA PRIMERA ENTRADA DE LLAVE EN LA SECUENCIA. EN CADA TABLA, LAS ENTRADAS SON MANTENIDAS EN ORDEN ASCENDENTE. LAS TABLAS DISPONIBLES SE MANTIENEN EN UNA LISTA LIGADA ENCABEZADA POR EL BLOQUE CERO.

LAS ENTRADAS DE LLAVE "FLOTAN" BAJO CONTROL DE LA PALABRA DE CONTROL DE TABLA, QUE APUNTA AL COMIENZO DE LA PRIMERA ENTRADA DE LLAVE EN USO, Y EL NUMERO DE ENTRADAS EN USO. EL NUMERO DE SERIE DE TABLA SE USA PARA COORDINAR EL ESTADO DE LA TABLA CON LA TRAZA DE MODIFICACIONES.

LISTA DESORDENADA.

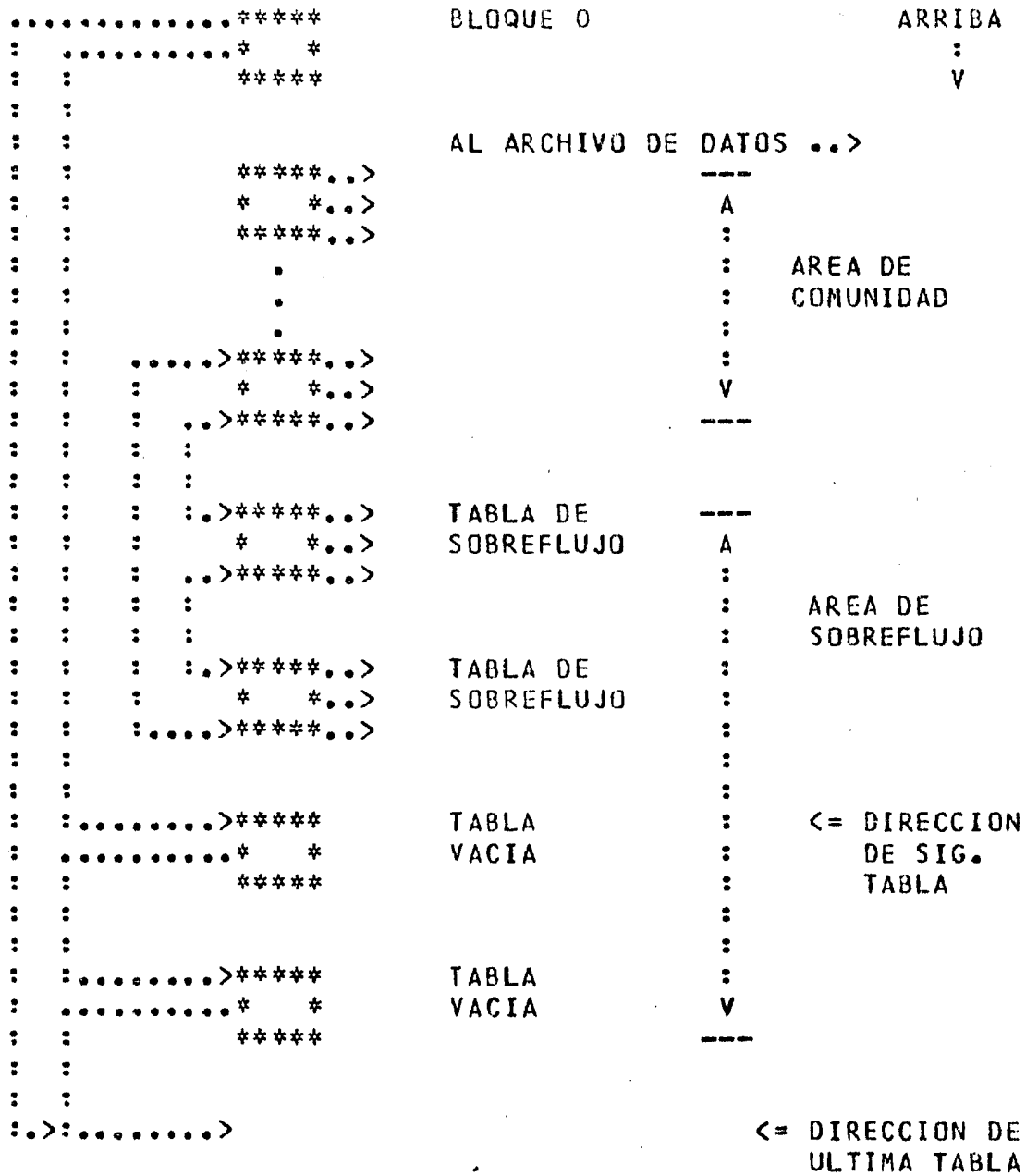
ESTA ESTRUCTURA ES UNA COLECCION DE TABLAS LIGADAS ENTRE SI MEDIANTE PALABRAS DE CONTROL, MUY SIMILAR A LA LISTA ORDENADA, PERO SIN LLAVES. ASI, LAS ENTRADAS CONSTAN SOLO DE DIRECCIONES Y NO ESTAN ORDENADAS. NO SE PUEDE BUSCAR POR UN VALOR DE LLAVE, SINO SOLO POR PRIMERO, ULTIMO, SIGUIENTE O ANTERIOR, SEGUN EL ORDEN FISICO, CON LA PALABRA DE DIRECCION ABSOLUTA TOMADA COMO LLAVE.

SI EL CONJUNTO AL QUE SE ASOCIA ES INCLUIDO, TODAS LAS ENTRADAS EN CADA TABLA PERTENECEN A UN MISMO PROPIETARIO.

CUANDO SE LLENA UNA TABLA, SE CONSTRUYE OTRA, A LA QUE SE LLEVA UNA PROPORCION DE ENTRADAS DE ACUERDO AL FACTOR DE CARGA.

AL AZAR CON INDICE.

LOS BLOQUES EN USO EN ESTA ESTRUCTURA CONSTITUYEN TABLAS DE DOS PARTES. EN LA PRIMERA PARTE DE UNA TABLA SE MANTIENEN EN ORDEN LAS LLAVES DESPUES DE APLICADO UN ALGORITMO DE DOBLADO. EN LA SEGUNDA PARTE SE TIENE LAS LLAVES SIMBOLICAS Y LAS DIRECCIONES DE LOS REGISTROS DEL CONJUNTO DE DATOS ASOCIADO.



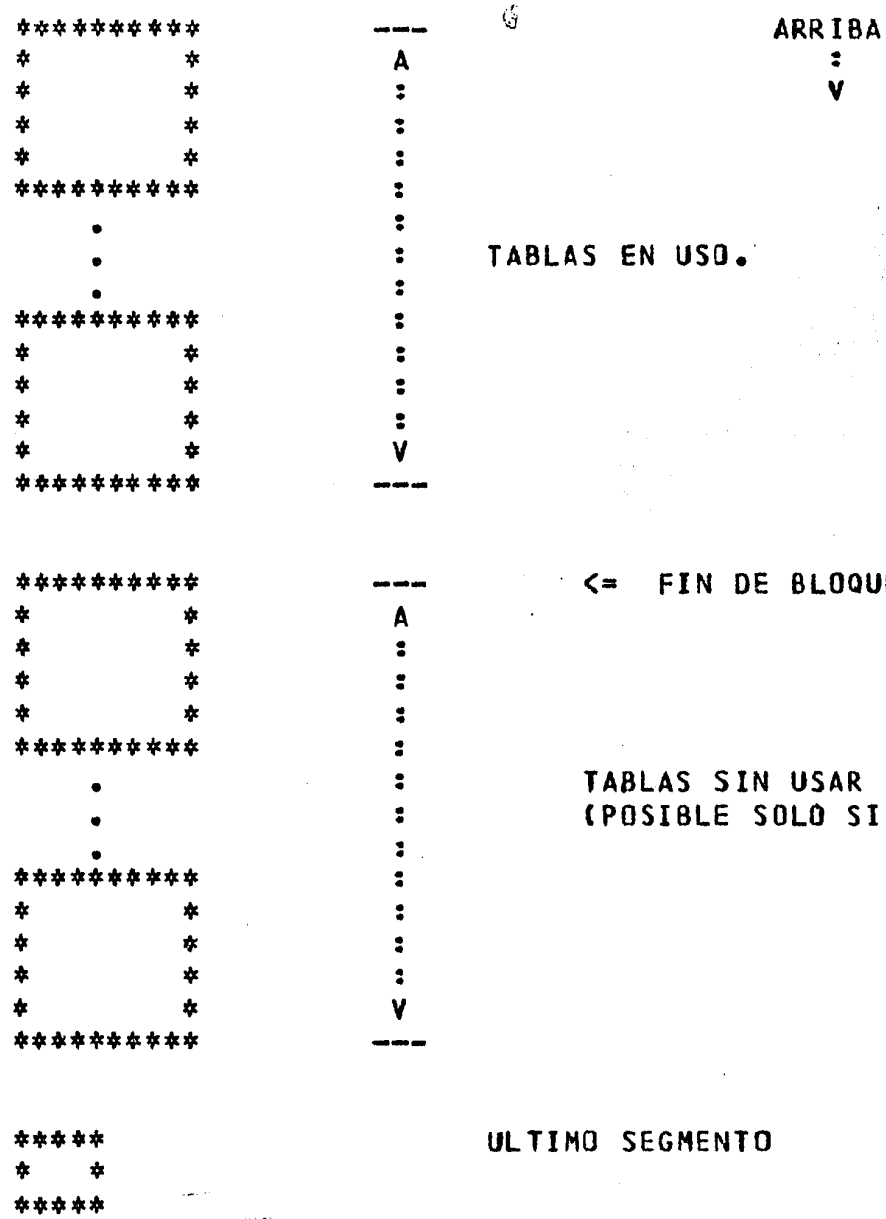
ESTA ESTRUCTURA SE PERMITE SOLO A NIVEL UNO. SE TIENEN UN AREA BASICA Y UNA DE SOBREFLUJO. A LAS LLAVES SE LES APLICA UN ALGORITMO DE DISPERSION, SENALANDO UNA TABLA EN EL AREA BASICA, EN LA QUE EMPIEZA LA BUSQUEDA, Y UN ALGORITMO DE DOBLADO, CUYO RESULTADO SE COLOCA EN LA PRIMERA MITAD DE CADA TABLA. ESTAS LLAVES DOBLADAS SON SUJETAS A UNA BUSQUEDA CON MASCARA, PARA DETERMINAR ENTRADAS EN USO Y DISPONIBLES. CUANDO SE LLENA UNA TABLA EN EL AREA BASICA, SE USA UNA EN EL AREA DE SOBREFLUJO. LAS TABLAS DE SOBREFLUJO SIENDO ASIGNADAS CONSTITUYEN UNA LISTA DOBLEMENTE LIGADA ENCABEZADA POR LA TABLA EN EL AREA BASICA. CUANDO SE DAN DE BAJA TODAS LAS ENTRADAS EN UNA TABLA DE SOBREFLUJO, SE REGRESA A LISTA LIGADA DE TABLAS VACIAS ENCABEZADA POR EL BLOQUE CERO.

LAS LLAVES DOBLADAS EN USO "FLOTAN" BAJO CONTROL DE LA PALABRA DE CONTROL DE TABLA, QUE DA EL CORRIMIENTO DE LA PRIMERA LLAVE DOBLADA EN USO Y EL NUMERO DE ELLAS EN USO. LAS LLAVES DOBLADAS SE CORRESPONDEN CON LAS POSICIONES DE LAS ENTRADAS DE LLAVE, Y LA PARTE DE APUNTADOR DE CADA LLAVE DOBLADA ES EL CORRIMIENTO A LA PRIMERA PALABRA DE LA CORRESPONDIENTE ENTRADA DE LLAVE. LAS LLAVES DOBLADAS EN USO SE MANTIENEN EN ORDEN DESCENDENTE.

LAS BUSQUEDAS POR SIGUIENTE EMPIEZAN EN LAS LLAVES DOBLADAS DE LA PRIMERA TABLA, EN ORDEN DESCENDENTE DE CORRIMIENTO, Y LUEGO EN LAS TABLAS DE SOBREFLUJO, SI LAS HAY, ANTES DE LA SIGUIENTE TABLA BASICA.

VECTOR DE DIGITOS BINARIOS.

EN ESTA ESTRUCTURA, DADO UN CONJUNTO DE DATOS, LOS DIGITOS BINARIOS CON VALOR DE UNO EN UNA CADENA DETERMINAN SI UN REGISTRO CAE EN UNA DETERMINADA CATEGORIA.



LOS DIGITOS BINARIOS LLEVAN UNA RELACION POSICIONAL IMPLICITA CON LOS REGISTROS DE DATOS. SE TIENEN TABLAS SIN USAR SOLO SI NO ES AUTOMATICO. SOLO SON PERMITIDAS A NIVEL UNO, Y CON RESPECTO A UN CONJUNTO DE DATOS ESTANDAR SIN REGISTROS DE FORMATO VARIABLE.

EL BIT 47 DE LA PALABRA CERO DE LA TABLA CERO CORRESPONDE AL REGISTRO EN EL BLOQUE CERO, CORRIMIENTO CERO DEL CONJUNTO DE DATOS, ETC. (ESTE BIT NUNCA ESTARA PRENDIDO, YA QUE PARA CONJUNTOS ESTANDARES DE DATOS NUNCA SE ASIGNA EL REGISTRO CERO) CLARAMENTE, ENCONTRAR POR LLAVE NO ES POSIBLE CON ESTA ESTRUCTURA, LOCALIZANDOSE SOLO POR SIGUIENTE, ANTERIOR, ETC.

IV.3.4 LIGAS.

LAS LIGAS SON ITEMS DEFINIDOS EN LOS CONJUNTOS DE DATOS, PARA RELACIONAR SUS MIEMBROS CON LOS DE OTROS CONJUNTOS. ALGUNOS TIPOS DE LIGAS SON ENTRE ORDENES.

SE TIENEN CINCO TIPOS DE LIGAS : SIMBOLICA, CONTADA, VERIFICADA, AUTO-CORREGIBLE Y NO PROTEGIDA.

A TIEMPO DE EJECUCION, DADOS LOS REGISTROS CORRIENTES DE DOS CONJUNTOS DE DATOS ENTRE LOS QUE SE DEFINIO UNA LIGA CONTADA, LA ASIGNACION DE LA LIGA CAUSA EL ALMACENAMIENTO EN EL REGISTRO AL QUE SE ASIGNA DE LA DIRECCION EN DISCO DEL REGISTRO ASIGNADO, Y EN ESTE EL INCREMENTO EN UNO DE LA CUENTA DE APUNTADORES. NO PUEDEN DARSE DE BAJA LOS REGISTROS CON UNA CUENTA DE ESTAS POSITIVA.

EN EL CASO DE UNA LIGA SIMBOLICA, A TIEMPO DE ASIGNACION SOLO SE ALMACENA LA LLAVE DEL REGISTRO SIENDO ASIGNADO EN EL CAMPO DE LIGA DEL REGISTRO CORRIENTE AL QUE SE ASIGNA. SE PUEDEN DAR DE BAJA O REACOMODARSE LOS REGISTROS EN EL CONJUNTO DE DATOS EN EL QUE ASIGNA, SIN CUIDAR QUE LOS REGISTROS ESTEN ASIGNADOS. CUANDO SE TRATA DE ENCONTRAR UN REGISTRO MEDIANTE LA LIGA, Y ESTA DADO DE BAJA, EL SISTEMA REPORTA UNA EXCEPCION.

LAS LIGAS VERIFICADAS CONSTAN DE UNA DIRECCION DE DISCO Y DE UN VALOR A VERIFICARSE, QUE SON ALMACENADOS EN EL REGISTRO CORRIENTE SIENDO ASIGNADO. CUANDO SE PRETENDE ENCONTRAR UN REGISTRO MEDIANTE LA LIGA, SE REPORTA UNA EXCEPCION SI EL VALOR A VERIFICARSE EN LA DIRECCION APUNTADA NO CORRESPONDE AL DEL CAMPO DE VERIFICACION.

EN LA ASIGNACION DE UNA LIGA AUTOCORREGIBLE SE ALMACENAN EN EL REGISTRO AL QUE SE ASIGNA AMBAS LA DIRECCION EN DISCO Y LA LLAVE DEL REGISTRO SIENDO ASIGNADO. CUANDO SE BUSCA EN EL CONJUNTO DE DATOS ASIGNADO USANDO LA LIGA, SI EL REGISTRO EN LA DIRECCION APUNTADA NO CORESPONDE A LA LLAVE, ESTA SE USA PARA ENCONTRAR A DONDE FUE MOVIDO EL REGISTRO BUSCADO, Y DE ENCONTRARSE SE CORRIGE LA DIRECCION. LA LLAVE HA DE DECLARARSE UNICA.

UNA LIGA NO PROTEGIDA CONSISTE SOLO DE LA DIRECCION EN DISCO. SI SE REORGANIZA EL CONJUNTO DE DATOS ASIGNADO, EL PROCESO A TRAVES DE ESAS LIGAS PUEDE LLEVAR A RESULTADOS IMPREDECIBLES.

V DISENO DE BASES DE DATOS EN DMS II.

V.1 CONSIDERACIONES BASICAS.

ENFOQUE DESCENDENTE.

EL ENFOQUE ASCENDENTE EN DISENO DE BASES DE DATOS ARRANCA CON UNA COLECCION DE ELEMENTOS DE DATOS. SEGUIDAMENTE SE DETERMINAN LAS DEPENDENCIAS FUNCIONALES ENTRE ELLOS, SEGUN EL CRITERIO DEL DISENADOR. ESTO ES, SI ES QUE LOS VALORES DE ALGUNOS ELEMENTOS DE DATOS DETERMINAN LOS VALORES DE OTROS, SE DICE QUE ESTOS SON FUNCIONALMENTE DEPENDIENTES DE AQUELLOS. (COMO EJEMPLO, EL NUMERO DE EMPLEADO DETERMINA SU NOMBRE, EDAD Y DEMAS DATOS.) ENTONCES SE AGRUPAN LOS DATOS (EN CONJUNTOS DE DATOS DE DMS II) Y SE DESIGNAN LAS LLAVES DE ACCESO, BUSCANDO OPTIMIZAR CIERTAS RUTAS DE ACCESO.

LOS CONJUNTOS DE DATOS RESULTANTES PUEDEN O NO CORRESPONDER A ENTIDADES DEL MUNDO REAL. ESTE ENFOQUE CORRESPONDE AL DE BASES DE DATOS RELACIONALES, CON LOS CONJUNTOS DE DATOS DE DMS II EN EL LUGAR DE LAS RELACIONES, LOS REGISTROS COMO LAS N-ADAS DE LAS RELACIONES, Y LOS ELEMENTOS DE DATOS COMO LOS ATRIBUTOS.

EL ENFOQUE DESCENDENTE ES A MENUDO MEJOR Y MAS INTUITIVO. EN RAZON DE QUE MUCHA GENTE TIENDE A PENSAR EN UN NIVEL MAS ELEVADO QUE AL DE ELEMENTOS DE DATOS. EN ESTE ENFOQUE SE COMIENZA CON GRUPOS DE ELEMENTOS DE DATOS QUE CORRESPONDAN SEGUN EL DISENADOR A LAS ENTIDADES INVOLUCRADAS EN LA SITUACION SIENDO MODELADA. LAS ENTIDADES Y SUS RELACIONES RECIPROCAS SE CLARIFICAN EN UNA SERIE DE REFINAMIENTOS SUCESIVOS, INTRODUCIENDOSE LOS ELEMENTOS DE DATOS A MEDIDA QUE EL DISENO PROGRESA. LOS GRUPOS DE ELEMENTOS DE DATOS SERAN CONJUNTOS DE DATOS DE DMS II, Y CORRESPONDERAN GENERALMENTE A ENTIDADES SIMPLES DEL MUNDO REAL.

SE RECOMIENDA ESTE ULTIMO ENFOQUE DADA LA TENDENCIA DE LAS BASES A SER MAS SIMPLES Y GENERALES, Y MAS FACILES DE EXPANDER, DE ENTENDER Y DE DISENAR.

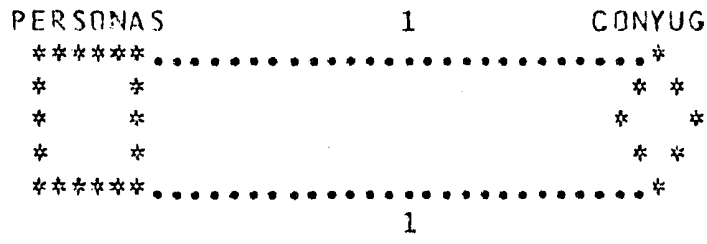
V.2 RELACIONES ENTRE CONJUNTOS DE DATOS.

RELACIONES UNO A UNO.

LAS RELACIONES UNO A UNO SON MAS SIMPLES PERO MENOS COMUNES QUE LAS RELACIONES 1 A N Y N A M. A MENUDO SE PUEDEN CONSEGUIR ANADIENDO UN ELEMENTO DE DATOS A UN REGISTRO, Y USANDO UN ORDEN YA EXISTENTE.

RELACIONES UNO A UNO EN UN SOLO CONJUNTO DE DATOS.

POR EJEMPLO TOMEMOS LA RELACION DE MATRIMONIO COMO DE UNO A UNO. SI TENEMOS UN SOLO CONJUNTO DE DATOS DE PERSONAS, PODEMOS REPRESENTARLA EN UN DIAGRAMA A NIVEL ENTIDAD COMO SIGUE :



EN UN DIAGRAMA COMO EL ANTERIOR, UN RECTANGULO DE ASTERISCOS REPRESENTA UN CONJUNTO DE DATOS DISJUNTO, UN DIAMANTE REPRESENTA UNA RELACION ENTRE LOS REGISTROS DE UNO O DOS CONJUNTOS DISJUNTOS, Y LOS DOS ARCOS CON PUNTOS SE ETIQUETAN PARA INDICAR SI LA RELACION ES UNO A UNO, 1 A N O N A M.

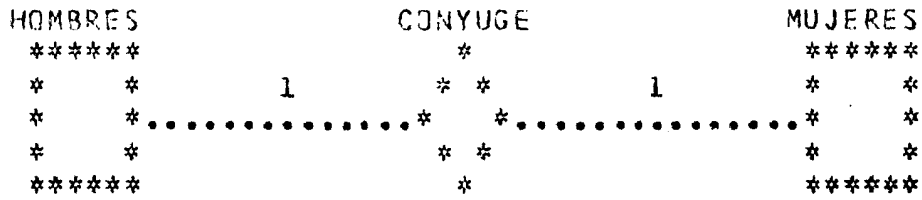
SI TENEMOS QUE EL NOMBRE DE CADA PERSONA ES UNICO, ENTONCES EN DASDL SE REPRESENTARIA :

```
PERSONAS DATA SET
          (
           NOMBRE              ALPHA(30);
           NOMBRECONYUGUE      ALPHA(30); % NULL SI SOLTERO
           .
           .
           .
          );
ORDENNOMBRE SET OF PERSONAS KEY(NOMBRE);
```

DADO UN REGISTRO DEL CONJUNTO DE DATOS DE PERSONAS, EL REGISTRO DEL CONYUGE SE PUEDE ENCONTRAR, SI LO TIENE, USANDO ORDENNOMBRE CON NOMBRE=NOMBRECONYUGE.

RELACIONES UNO A UNO ENTRE DOS CONJUNTOS DE DATOS.

SE PUEDE USAR LA MISMA TECNICA, CON DOS CONJUNTOS DE DATOS. SUPONGAMOS QUE TENEMOS DOS CONJUNTOS DE DATOS, UNO PARA HOMBRES Y OTRO PARA MUJERES. PODEMOS REPRESENTAR LA RELACION MATRIMONIO COMO SIGUE :



REPRESENTANDOSE EN DASDL ASI :

```
HOMBRES DATA SET
          (
           NOMBRE              ALPHA(30);
           NOMBREESPOSA        ALPHA(30); % NULL SI SOLTERO
           .
           .
           .
          );
ORDENHOMBRES SET OF HOMBRES KEY (NOMBRE);

MUJERES DATA SET
          (
           NOMBRE              ALPHA (30);
           NOMBREESPOSO        ALPHA (30); % NULL SI SOLTERA
           .
           .
           .
          );
ORDENMUJERES SET OF MUJERES KEY (NOMBRE);
```

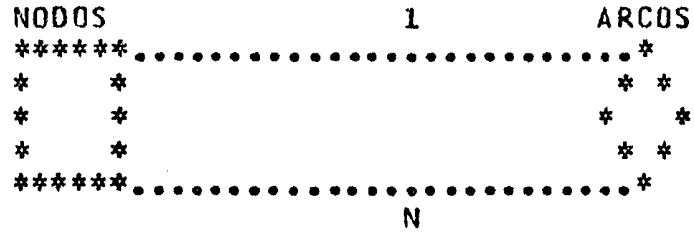
DADO UN REGISTRO DEL CONJUNTO DE DATOS DE HOMBRES, EL REGISTRO DE LA ESPOSA EN EL CONJUNTO DE DATOS DE MUJERES PUEDE ENCONTRARSE USANDO ORDENMUJERES CON NOMBRE=NOMBREESPOSA. SIMILARMENTE PARA ENCONTRAR EL NOMBRE DEL ESPOSO DADO UN REGISTRO DE MUJERES.

RELACIONES 1 A N.

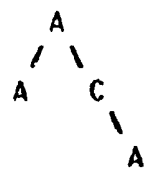
PARA REPRESENTAR RELACIONES 1 A N EN UN CONJUNTO DE DATOS O ENTRE DOS, SE PUEDEN USAR DOS ORDENES AUTOMATICOS DE DMS II, UNO PARA CADA DIRECCION EN LA QUE SE PUEDE VIAJAR EN LA RELACION. USAREMOS EL CONJUNTO DE DATOS DE NOMBRES DE ARCHIVO COMO UN EJEMPLO.

RELACIONES 1 A N EN UN SOLO CONJUNTO DE DATOS.

LA ESTRUCTURA DE NOMENCLATURA DE ARCHIVOS EN BURROUGHS B6700 DA LUGAR A UNA ESTRUCTURA DE ARBOL DONDE CADA NODO TIENE ASOCIADO UN IDENTIFICADOR DE HASTA 17 CARACTERES EBCDIC. SE DESEA ALMACENAR FISICAMENTE LOS IDENTIFICADORES DE MANERA DE AHORRAR ESPACIO. ASI QUE EN LUGAR DE UN CONJUNTO DE DATOS DE NOMBRES DE ARCHIVOS, SE SUGIERE UN CONJUNTO DE DATOS DE NODOS CON LOS IDENTIFICADORES. HAY UNA RELACION ENTRE LOS NODOS DE ESTE CONJUNTO DE DATOS QUE DA LOS ARCOS EN EL ARBOL DE NOMBRES DE ARCHIVOS. LO REPRESENTAMOS COMO SIGUE :



UN IDENTIFICADOR PUEDE APARECER EN VARIOS NODOS DIFERENTES. ESTO SE VE FACILMENTE EN LOS NOMBRES DE LOS ARCHIVOS A/A Y A/C/A. EL IDEENTIFICADOR "A" APARECE EN TRES NODOS DIFERENTES EN LA ESTRUCTURA DE ARBOL PARA LOS DOS NOMBRES :



ASI QUE INTRODUCIMOS UN ELEMENTO DE DATOS, NUMERO UNICO DE NODO LLAMADO NUMNODO.

POR CONVENCION, SEAN DOS NODOS VIRTUALES QUE NO SE ALMACENAN REALMENTE, CON NUMEROS DE NODO 0 Y 1. EL NODO 0 SERA EL ULTIMO PADRE DE TODOS LOS ARCHIVOS DEL SISTEMA, Y EL 1 EL DE TODOS LOS ARCHIVOS DE CLAVE DE USUARIO. UN NOMBRE DE ARCHIVO SERA LA CONCATENACION DE TODOS LOS IDENTIFICADORES DE NODO EN LA RUTA DEL NODO RAIZ AL NODO HOJA DEL ARBOL.

QUISIERAMOS EVITAR ALMACENAR DOS VECES UN MISMO IDENTIFICADOR, EN EL CASO QUE SEA A LA VEZ DIRECTORIO E IDENTIFICADOR FINAL DE UN NOMBRE DE ARCHIVO. POR EJEMPLO, PARA LOS DOS NOMBRES A Y A/B QUISIERAMOS GUARDAR "A" SOLO UNA VEZ. ESTO LO LOGRAMOS SI PONEMOS DOS BANDERAS EN EL REGISTRO DEL NODO. BANDDIR INDICA SI EL IDENTIFICADOR ES PARTE O NO DE UN NOMBRE MAS LARGO, Y BANDARCH INDICA SI EL IDENTIFICADOR ES IDENTIFICADOR ULTIMO DE UN NOMBRE DE ARCHIVO. CUANDO ESTAN PRENDIDAS AMBAS BANDERAS EN UN REGISTRO, HEMOS AHORRADO GUARDAR AL IDENTIFICADOR UNA SEGUNDA VEZ. DE HECHO SOLO ES NECESARIA BANDARCH, PERO ES MAS EFICIENTE USAR BANDDIR, PARA NO BUSCAR INNECESARIAMENTE MAS ALLA DE NODOS HOJA.

ADEMAS DE SU IDENTIFICADOR NUMNODO, CADA NODO TIENE UN NODO PADRE INMEDIATO UNICO. LLAMEMOSLE NUMNODOPADRE. Y REPRESENTAMOS HASTA AQUI EN DASDL COMO SIGUE :

```
NODOS DATA SET
          (
           NUMNODO              FIELD(24);
           NUMNODOPADRE         FIELD(24);
           IDENTIF              ALPHA(17);
           BANDERAS FIELD
                    (
                     BANDDIR;
                     BANDARCH;
                    );
          );
```

POR EJEMPLO, SUPONGAMOS QUE ALMACENAMOS LOS SIGUIENTES NOMBRES DE ARCHIVO, EN ORDEN, Y ASIGNAMOS NUMEROS DE NODO CONSECUTIVAMENTE DESDE EL 2 :

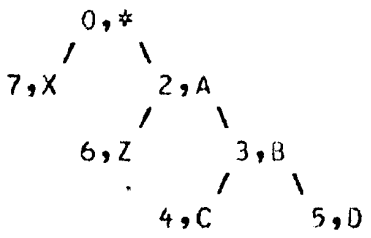
```
*A
*A/B/C
*A/B/D
*A/Z
*X
```

APARECERIAN LOS SIGUIENTES REGISTROS EN EL CONJUNTO DE DATOS DE NODOS:

NODOS

NUMNODO	NUMNODOPADRE	IDENTIF	BANDDIR	BANDARCH
2	0	A	CIERTO	CIERTO
3	2	B	CIERTO	FALSO
4	3	C	FALSO	CIERTO
5	3	D	FALSO	CIERTO
6	2	Z	FALSO	CIERTO
7	0	X	FALSO	CIERTO

ESTA SITUACION REPRESENTA AL ARBOL SIGUIENTE :



LA RELACION DE ARCOS PUEDE REPRESENTARSE CON LOS SIGUIENTES DOS ORDENES :

```
ORDENARCOS SET OF NODOS KEY (NUMNODOPADRE, IDENTIF);
ORDENNODOS SET OF NODOS KEY (NUMNODO);
```

POR DEFAULT LOS DOS ORDENES SON ESTRUCTURAS SECUENCIALES CON INDICE SIN LLAVES DUPLICADAS PERMITIDAS.

DADO UN NODO X, LOS NODOS INMEDIATAMENTE DEBAJO DE EL PUEDEN SER ENCONTRADOS SIMPLE Y EFICIENTEMENTE, EN ORDEN ALFABETICO DE IDENTIFICADOR, COMO SIGUE :

```
SET ORDENARCOS TO BEGINNING;
DO BEGIN
     FIND NEXT ORDENARCOS AR NUMNODOPADRE=X :RSLT;
     IF NOT RSLT THEN
     BEGIN
          [PROCESA REGISTRO DE NODO]
     END;
END UNTIL RSLT;
IF REAL(RSLT).DMERROR NEQ NOTFOUND THEN
     DMTERMINATE(RSLT);
```

Y, DADO UN NODO Y, SU PADRE PUEDE SER ENCONTRADO COMO SIGUE :

```
GET NODOS(NUMNODOPADRE := NUMNODOPADRE);
FIND ORDENNODOS AT NUMNODO=NUMNODOPADRE :RSLT;
```

LOS NUMEROS DE NODO UNICOS SERAN CREADOS POR EL PROGRAMA DE USUARIO PARA ACTUALIZACION. PARA NO EXCEDER EL CAMPO PARA NUMERUS DE NODO HEMOS DE USAR DE NUEVO LOS NUMEROS DE NODO, PARA LO CUAL INTRODUCIMOS OTRA BANDERA, ENUSO. TENEMOS ENTONCES EN DASDL :

```
NODOS DATA SET
          (
           NUMNODO              FIELD(24);
           NUMNODOPADRE         FIELD(24);
           IDENTIF              ALPHA(17);
           BANDERAS FIELD
                    (
                     BANDDIR;
                     BANDARCH;
                    );
           ENUSO                NUMBER(1);
          );

ORDENARCOS SET OF NODOS KEY(ENUSO, NUMNODOPADRE, IDENTIF);
ORDENNODOS SET OF NODOS KEY(ENUSO, NUMNODO);
```

LOS NODOS INMEDIATAMENTE ABAJO DE UN NODO X PUEDEN SER HALLADOS COMO SIGUE :

```
SET ORDENARCOS TO BEGINNING;
DO BEGIN
     FIND NEXT ORDENARCOS AT ENUSO=1 AND
          NUMNODOPADRE=X :RSLT;
     IF NOT RSLT THEN
     BEGIN
          [PROCESA REGISTRO DE ARCOS]
     END;
END UNTIL RSLT;
IF REAL(RSLT).DMERROR NEQ NOTFOUND THEN
     DMTERMINATE(RSLT);
```

EL NODO PADRE DE UN NODO Y PUEDE SER HALLADO COMO SIGUE :

```
GET NODOS(NUMNODOPADRE := NUMNODOPADRE);
FIND ORDENNODOS AT ENUSO=1 AND NUMNODO=NUMNODOPADRE;
```

SE PUEDE HACER UN NUEVO NODO COMO SIGUE :

```
LOCK FIRST ORDENNODOS AT ENUSO=0 :RSLT;
IF RSLT THEN
BEGIN
     IF REAL(RSLT).DMERROR NEQ NOTFOUND THEN
          DMTERMINATE(RSLT);
     LOCK LAST ORDENNODOS :RSLT;
     IF RSLT THEN
     BEGIN
          IF REAL(RSLT).DMERROR NEQ NOTFOUND THEN
               DMTERMINATE(RSLT);
          [CONJUNTO DE DATOS DE NODOS COMPLETAMENTE VACIO.
           ALMACENESE PRIMER NODO CON NUMNODO=1.]
     END ELSE
     BEGIN
          GET NODOS(X := NUMNODO);
          [TODOS LOS NODOS EN USO. CREESE UN NUEVO NODO CON
           NUMNODO = X+1.]
     END;
END ELSE
BEGIN
     [ESTE NODO NO EN USO. REUSESE ESTE NODO.
      CAMBIESE ENUSO DE 0 A 1, PERO MANTENGASE EL MISMO
      VALOR PARA NUMNODO.]
END;
```

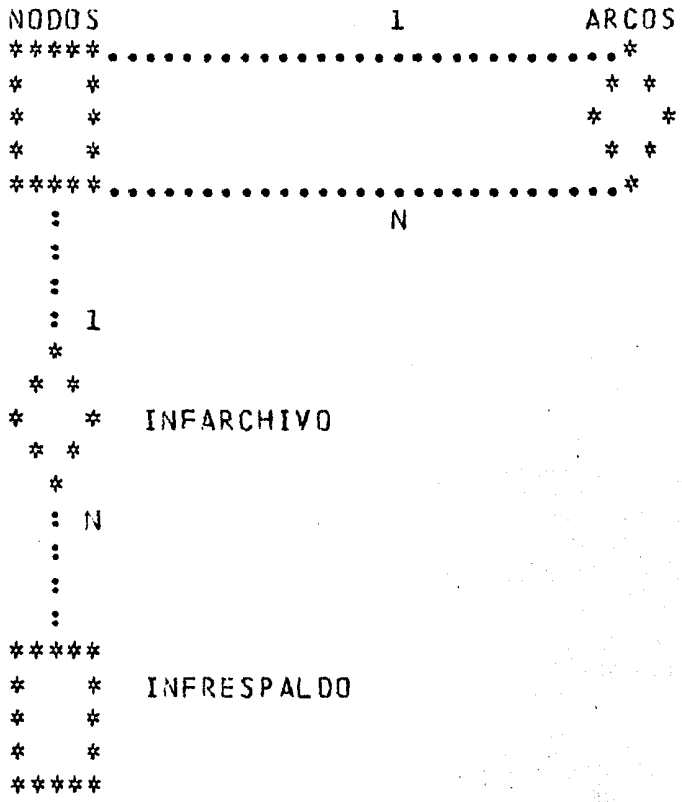
PARA BUSCAR UN ARCHIVO POR SU NOMBRE, SE PUEDE USAR EL SIGUIENTE PROCEDIMIENTO, QUE DA EL ULTIMO NODO CON UN IDENTIFICADOR EN COMUN. ASI, DE NO HALLARSE, SE FACILITARIA ALMACENAR UN NUEVO NOMBRE.

```
EBCDIC ARRAY E[0:13, 0:16]; % E[I,*] GUARDA EL I-ESIMO
                            % IDENTIFICADOR DEL NOMBRE
                            % (I > 0).
PROCEDURE BUSCA (E, N, ULTNODOCOMUN, ULTNIVELCHECADO);
     VALUE N;
     EBCDIC ARRAY E[0,0]; % GUARDA LOS IDENTIFICADORES
     REAL
          ULTNODOCOMUN, % NUM DE NODO DEL ULTIMO EN COMUN
          ULTNIVELCHECADO, % INDICE DEL ULTIMO IDENTIFICADO
          N;         % NUMERO DE IDENTIFICADORES
BEGIN
     REAL I, NUMNODO;
     BOOLEAN BANDDIR, RSLT;

     BANDDIR := TRUE;
     I := 1;
     DO BEGIN
          FIND ORDENNODOS AT ENUSO=1 AND
               NUMNODOPADRE=NUMNODO AND
               IDENTIF=E[I,0] :RSLT;
          IF NOT RSLT THEN
               GET NODOS
                    (NUMNODO := NUMNODO
                    ,BANDDIR := BANDDIR
                    );
     END UNTIL (I := *+1) > N OR RSLT OR NOT BANDDIR;
     ULTNIVELCHECADO := I-1;
     ULTNODOCOMUN := NUMNODO;
     IF RSLT THEN
          IF REAL(RSLT).DMERROR NEQ NOTFOUND THEN
               DMTERMINATE(RSLT);
END;
```

RELACIONES 1 A N ENTRE DOS CONJUNTOS DE DATOS.

SIGUIENDO EL EJEMPLO DE ARCHIVOS, SI ES NECESARIO ALMACENAR INFORMACION ACERCA DE DONDE SE TIENE RESPALDADO CADA ARCHIVO, SE PUEDE ANADIR UN CONJUNTO DE DATOS LLAMADO INFRESPALDO. CADA REGISTRO DE INFRESPALDO CONTIENE INFORMACION ACERCA DE UNA COPIA DE RESPALDO DE UN ARCHIVO. SE PUEDE USAR EL NUMERO DE NODO DEL ULTIMO NODO DEL NOMBRE PARA RELACIONAR UN ARCHIVO CON SU INFORMACION DE RESPALDO, EN UNA MANERA 1 A N. ADEMAS, SERIA DESEABLE PODER OBTENER UNA LISTA DE TODOS LOS ARCHIVOS RESPALDADOS EN UNA CINTA O PAQUETE PARTICULAR. FINALMENTE, HABRIA DE SER FACIL OBTENER LA INFORMACION DE RESPALDO DE TODOS LOS ARCHIVOS DE CUALQUIER FAMILIA DE PAQUETES. ESTO PUEDE SER REPRESENTADO COMO SIGUE :



Y A LA DESCRIPCION DE DASDL, ANADARIAMOS LO QUE SIGUE :

```
INFRESPALDO DATA SET
          (
           NUMNODO              FIELD(24);
           NUMSERIE             ALPHA(6);
           HORAFECHA GROUP
                    (
                     ANO                  NUMBER(2);
                     MES                  NUMBER(2);
                     DIA                  NUMBER(2);
                     HORA                 NUMBER(2);
                     MINUTO               NUMBER(2);
                     SEGUNDO              NUMBER(4,2);
                    );
           TIPOARCH             ALPHA(17);
           CICLO                NUMBER(4);
           VERSION              NUMBER(2);
           IDENTIFFAM           ALPHA(17);
           TIPOMEDIO            ALPHA(6);
           INDICEFAM            NUMBER(3);
           SERIEBASE            ALPHA(6);
           SERIESIG             ALPHA(6);
           FECHAMEDIO GROUP
                    (
                     ANOMEDIO             NUMBER(2);
                     MESMEDIO             NUMBER(2);
                     DIAMEDIO             NUMBER(2);
                    );

INFARCHIVO SET OF INFRESPALDO KEY(NUMNODO,HORAFECHA DESCENDING);
ORDENSERIE SET OF INFRESPALDO KEY(NUMSERIE) DUPLICATES;
ORDENFAM SET OF INFRESPALDO KEY(IDENTIFFAM) DUPLICATES;
```

SE INCLUYO HORAFECHA EN EL ORDEN INFARCHIVO COMO ELEMENTO DE LLAVE DESCENDENTE PARA FACILITAR LA RECUPERACION DE LA INFORMACION DE RESPALDO DE UN ARCHIVO EN ORDEN INVERSO AL TIEMPO.

SE PERMITIERON DUPLICADOS EN ORDENSERIE Y EN ORDENFAM, PERO NO SE ESPECIFICARON COMO PRIMEROS O ULTIMOS, PORQUE ES MAS EFICIENTE EN STORE Y EN DELETE SI EL NUMERO DE DUPLICADOS ES GRANDE. SIN EMBARGO, ESTO CUESTA UNA PALABRA EXTRA POR ENTRADA.

RELACIONES N A M.

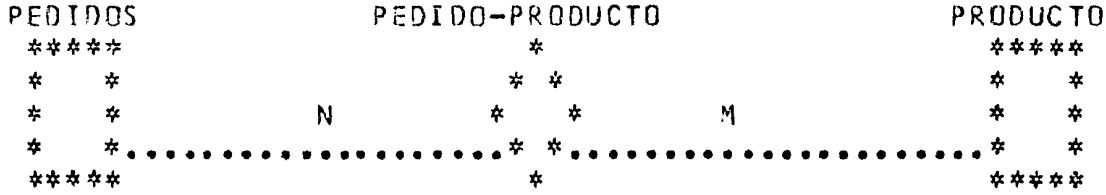
RELACIONES N A M ENTRE DOS CONJUNTOS DE DATOS.

EJEMPLIFICAREMOS CON LA SIGUIENTE DESCRIPCION DE DASDL :

```
PEDIDOS DATA SET
          (
           NUMERODEPEDIDO      NUMBER(6);
           FECHAVENCIMIENTO    NUMBER(6);
           FECHAPEDIDO         NUMBER(6);
           CLIENTE             NUMBER(6);
          );
LLAVEPEDIDO SET OF PEDIDOS KEY(NUMERODEPEDIDO);

PRODUCTOS DATA SET
          (
           NUMERODEPARTE       NUMBER(8);
           DESCRIPCION         ALPHA(24);
           PRECIO              REAL;
          );
LLAVEPRODUCTO SET OF PRODUCTOS KEY(NUMERODEPARTE);
```

SUPONGAMOS QUE PODEMOS TENER UN CIERTO NUMERO DE PRODUCTOS ASOCIADOS CON UN PEDIDO DADO, Y UN CIERTO NUMERO DE PEDIDOS ASOCIADOS CON CADA PRODUCTO. A NIVEL DE ENTIDADES, EL SIGUIENTE DIAGRAMA NOS REPRESENTA LA SITUACION :



LA PREGUNTA ES COMO SE PUEDE REPRESENTAR ESTO EN DMS II : EN DIVERSAS FORMAS. LA PRIMERA, QUE SE LE OCURRE A MUCHA GENTE, ES USAR ORDENES MANUALES INCLUIDOS. SIN EMBARGO, TIPICAMENTE ESTA NO ES LA MEJOR MANERA. GENERALMENTE LA MEJOR ES ANADIR UN CONJUNTO DE DATOS DISJUNTO, AL QUE NOS REFERIREMOS COMO EL CONJUNTO RELACION. SE ALMACENA UN REGISTRO EN EL CONJUNTO RELACION PARA CADA CONEXION ENTRE UN PAR DE REGISTROS, UNO DE CADA CONJUNTO DE DATOS QUE ESTEMOS RELACIONANDO. PARA IR DE UNO A OTRO DE LOS CONJUNTOS DE DATOS ORIGINALES USAMOS DOS ORDENES AUTOMATICOS. LAS LLAVES DE LOS ORDENES SON LAS DE LOS CONJUNTOS ORIGINALES, CONCATENADOS.

PARA EL EJEMPLO ANTERIOR, EL CONJUNTO RELACION SE REPRESENTARIA COMO SIGUE :

```
PEDIDO-PRODUCTO DATA SET
          (
           NUMERODEPARTE       NUMBER(8);
           NUMERODEPEDIDO      NUMBER(6);
          );
LLAVEPEDIDOPRODUCTO SET OF PEDIDO-PRODUCTO
          KEY(NUMERODEPEDIDO, NUMERODEPARTE);
LLAVEPRODUCTOPEDIDO SET OF PEDIDO-PRODUCTO
          KEY(NUMERODEPARTE, NUMERODEPEDIDO);
```

ENTONCES, PODIESEMOS ENCONTRAR TODOS LOS PRODUCTOS ASOCIADOS CON EL PEDIDO NUMERO X COMO SIGUE :

```
REAL X, NUMERODEPARTE;
BOOLEAN RSLT;

SET LLAVEPEDIDOPRODUCTO TO BEGINNING;
DO BEGIN
     FIND KEY OF NEXT LLAVEPEDIDOPRODUCTO
          AT NUMERODEPEDIDO=X :RSLT;
     IF NOT RSLT THEN
     BEGIN
        GET PEDIDO-PRODUCTO(NUMERODEPARTE := NUMERODEPARTE)
        FIND LLAVEPRODUCTO AT NUMERODEPARTE=NUMERODEPARTE;
        [PROCESA REGISTRO DE PRODUCTOS]
     END;
END UNTIL RSLT;
IF REAL(RSLT).DMERROR NEQ NOTFOUND THEN
     DMTERMINATE(RSLT);
```

DE MANERA SIMILAR, PUDIESEMOS ENCONTRAR TODOS LOS PEDIDOS DEL PRODUCTO NUMERO Y COMO SIGUE :

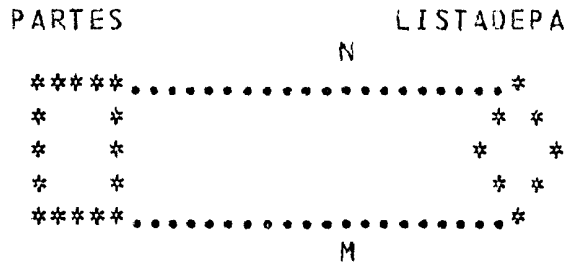
```
REAL Y, NUMERODEPEDIDO;
BOOLEAN RSLT;

SET LLAVEPRODUCTOPEDIDO TO BEGINNING;
DO BEGIN
     FIND KEY OF NEXT LLAVEPRODUCTOPEDIDO
          AT NUMERODEPARTE=Y :RSLT;
     IF NOT RSLT THEN
     BEGIN
       GET PEDIDO-PRODUCTO(NUMERODEPEDIDO:=NUMERODEPEDIDO);
       FIND LLAVEPEDIDO AT NUMERODEPEDIDO=NUMERODEPEDIDO;
       [PROCESA REGISTRO DE PEDIDOS]
     END;
END UNTIL RSLT;
IF REAL(RSLT).DMERROR NEQ NOTFOUND THEN
     DMTERMINATE(RSLT);
```

SE ILUSTRA CON ESTOS FRAGMENTOS DE PROGRAMAS LA CARACTERISTICA DE "FIND KEY OF" DE ORDENES INDEXADOS CON LLAVE. CUANDO SE USA ESTA CONSTRUCCION, SOLO SE ACCESA EL ORDEN, SIN IR AL CONJUNTO DE DATOS. SOLO SE TRANSFIEREN AL AREA DE TRABAJO DEL USUARIO LOS ELEMENTOS DE LLAVE DEL ORDEN, MAS LOS DATOS SI EL ORDEN LOS TUVIESE. DE TAL FORMA, ADEMAS DE SER FACIL DE ENTENDER, LA TECNICA DE CONJUNTO RELACION PUEDE TAMBIEN SER EFICIENTE.

RELACIONES N A M EN UN SOLO CONJUNTO DE DATOS.

LO EJEMPLIFICAREMOS CON UNA ESTRUCTURA DE PRODUCTOS, QUE PUEDEN ESTAR CONSTITUIDOS POR PARTES, QUE A SU VEZ PUEDEN ESTAR CONSTITUIDAS POR PARTES, ETC. LA RELACION INVERSA DE ESTA ES LA DE EN QUE PARTES ES USADA CADA PARTE, Y AQUELLAS EN CUALES OTRAS SON USADAS, ETC. CON UN DIAGRAMA DE NIVEL ENTIDADES, ESTO SE PRESENTA ASI :



LAS SUBPARTES COMUNES SON COMPARTIDAS PARA CONSERVAR ESPACIO Y PARA EVITAR EL PROBLEMA DE COPIAS INCONSISTENTES. ESTO HACE LA RELACION N A M EN LUGAR DE 1 A N.

LA SITUACION, EN LA QUE NUMERODEPARTE IDENTIFICA DE MANERA UNICA CADA PARTE, PUEDE REPRESENTARSE EN DASDL COMO SIGUE :

```
PARTES RANDOM DATA SET
          (
           NUMERODEPARTE       NUMBER(8);
           DESCRIPCION         ALPHA(28);
           PRECIO              REAL;
           .
           .
           .
          );
LLAVEPARTES ACCESS TO PARTES KEY NUMERODEPARTE MODULUS 1500;

LISTADEPARTES DATA SET
          (
           NUMERODEPARTE       NUMBER(8);
           NUMDESUBPARTE       NUMBER(8);
          );

ANALIZA SET OF LISTADEPARTES KEY(NUMERODEPARTE, NUMDESUBPARTE);
SEUSAEN SET OF LISTADEPARTES KEY(NUMDESUBPARTE, NUMERODEPARTE);
```

PARA PROCESAR TODAS LAS PARTES QUE COMPONEN UNA PARTE HASTA SUS PARTES INDIVISIBLES, SE PUEDE USAR EL SIGUIENTE PROCEDIMENTO EN ALGOL, QUE RECORRE EL ARBOL EN FORMA POLACA PREFIJA.

```
PROCEDURE QUEPARTES(NUMERODEPARTE, NIVEL);
          VALUE NUMERODEPARTE, NIVEL;
          REAL NUMERODEPARTE, NIVEL;
BEGIN
          REAL NUMDESUBPARTE;
          BOOLEAN RSLT;
          FIND LLAVEPARTES AT NUMERODEPARTE=NUMERODEPARTE;
          [PROCESA REGISTRO DE PARTES]
          DO BEGIN
               FIND KEY OF FIRST ANALIZA AT
                    NUMERODEPARTE=NUMERODEPARTE AND
                    NUMDESUBPARTE > NUMDESUBPARTE :RSLT;
               IF NOT RSLT THEN
               BEGIN
                    GET LISTADEPARTES
                    (NUMDESUBPARTE:=NUMDESUBPARTE);
                    QUEPARTES(NUMDESUBPARTE, NIVEL+1);
               END;
          END UNTIL RSLT;
          IF REAL(RSLT).DMERROR NEQ NOTFOUND THEN
               DMTERMINATE(RSLT);
END;
```

Y PARA UNA PARTE X DADA, SE LLAMARIA :

```
QUEPARTES(X, 1);
```

EL PROCEDIMIENTO ES RECURSIVO, Y EL VALOR DE NIVEL INDICA LA PROFUNDIDAD RECURSIVA EN EL ARBOL DE LISTA DE PARTES. LA BUSQUEDA EN ANALIZA NO ES AFECTADA POR EL HECHO DE QUE CADA LLAMADA RECURSIVA EN QUEPARTES CAMBIA LA RUTA CORRIENTE DE ANALIZA. AL GUARDAR EL NUMERO DE LA PARTE SIENDO ANALIZADA Y EL NUMERO DE LA ULTIMA PARTE PROCESADA HACIA ABAJO, DE MANERA LOCAL AL PROCEDIMIENTO, SE ELIMINA LA NECESIDAD DE GUARDAR LA RUTA VIEJA. EL PROCEDIMIENTO DEPENDE DE QUE LA LLAVE DE ANALIZA SEA UNICA, YA QUE SE BRINCA CUALQUIER DUPLICADO.

EL PROCEDIMIENTO ES EFICIENTE SI SE ESPECIFICAN UN NUMERO RAZONABLE DE AREAS DE TRABAJO EN DASD PARA ANALIZA, COMO UNAS CUANTAS MAS QUE EL NUMERO MAXIMO DE NIVELES EN LA ESTRUCTURA DE PRODUCTOS.

PARA PROCESAR LAS PARTES EN LAS QUE SE USA UNA DETERMINADA PARTE, Y EN CUALES SON USADAS AQUELLAS, ETC., HASTA DONDE YA NO SON SUBPARTES DE NINGUNA OTRA PARTE, ESTO ES, RECORRER LA LISTA DE DONDE SE USA, SE PUEDE USAR EL SIGUIENTE PROCEDIMIENTO :

```
PROCEDURE ENCUALES(NUMERODEPARTE, NIVEL);
     VALUE NUMERODEPARTE, NIVEL;
     REAL NUMERODEPARTE, NIVEL;
BEGIN
     REAL PARTEARRIBA;
     BOOLEAN RSLT;

     FIND LLAVEPARTES AT NUMERODEPARTE=NUMERODEPARTE;
     [PROCESA REGISTRO DE PARTES]
     DO BEGIN
          FIND KEY OF FIRST SEUSAEN AT
               NUMDESUBPARTE=NUMERODEPARTE AND
               NUMERODEPARTE > PARTEARRIBA :RSLT;
          IF NOT RSLT THEN
          BEGIN
            GET LISTADEPARTES(PARTEARRIBA := NUMERODEPARTE)
            ENCUALES(PARTEARRIBA, NIVEL+1);
          END;
     END UNTIL RSLT;
     IF REAL(RSLT).DMERROR NEQ NOTFOUND THEN
          DMTERMINATE(RSLT);
END;
```

PUEDE LLAMARSE PARA PRODUCIR LA LISTA COMPLETA DE DONDE SE USA LA PARTE CON NUMERODEPARTE Y COMO SIGUE :

```
ENCUALES(Y, 1);
```

SE USA PARTES COMO UN CONJUNTO DE DATOS AL AZAR PORQUE NO IMPORTA QUE TECNICA SE USE PARA DETERMINAR QUE PARTES CONSTITUYEN, O DE QUE PARTES FORMA PARTE UNA DADA, LOS ACCESOS SERAN AL AZAR EN VEZ DE SECUENCIALES. LOS CONJUNTOS DE DATOS AL AZAR SE DISENARON PARA RECUPERAR EL REGISTRO DESEADO EN UN ACCESO DE DISCO EN PROMEDIO. UNA COMBINACION DE ACCESOS AL ORDEN Y AL CONJUNTO DE DATOS TIPICAMENTE TOMARIAN EN PROMEDIO DOS ACCESOS DE DISCO APROXIMADAMENTE.

TAMBIEN SE PUDIESEN ANADIR ORDENES SECUENCIALES CON INDICE, PARA RECUPERAR DEL CONJUNTO DE PARTES, POR EJEMPLO SI SE QUISIERA OBTENER UN REPORTE DE PARTES EN ORDEN DE NUMERO DE PARTE.

V.3 INFORMACION ADICIONAL EN RELACIONES.

LA TECNICA DE CONJUNTO DE DATOS RELACION PERMITE QUE SE REPRESENTE CONVENIENTEMENTE INFORMACION ADICIONAL DE LAS RELACIONES ENTRE DOS CONJUNTOS DE DATOS, LO QUE NO ES POSIBLE USANDO LIGAS NI SUBORDENES MANUALES. LOS ILUSTRAREMOS AMPLIANDO EL EJEMPLO DE LA ESTRUCTURA DE PRODUCTOS.

SE IDENTIFICARAN LAS OPERACIONES DE ENSAMBLADO, CON LAS QUE SE VAN ANADIENDO LAS SUBPARTES A UNA PARTE DADA, CADA UNA CON UN NUMERO, QUE LLAMAREMOS NUMOPER. TAMBIEN IDENTIFICAREMOS LAS DISTINTAS SUBLISTAS DE PARTES QUE HACEN QUE UNA MISMA PARTE SE INTEGRE EN DISTINTAS OPERACIONES, CON NUMITEM. LAS SUBLISTAS DE PARTES SE IMPRIMIRAN EN ORDEN DE NUMITEM.

TENDREMOS UN CAMPO DE CANTIDAD DE SUBPARTE, PARA EVITAR ALMACENAR REGISTROS REPETIDOS PARA PARTES QUE APARECEN VARIAS VECES COMO SUBPARTES, Y PARA MANEJAR CANTIDADES NO DISCRETAS, COMO LIQUIDOS.

LA ESTRUCTURA DE UN PRODUCTO PUEDE CAMBIAR A MEDIDA QUE PASA EL TIEMPO. SIN EMBARGO, SE PROCURA USAR LA MAYORIA DE LAS PARTES EN ALMACEN ANTES DE ENSAMBLAR NUEVAS VERSIONES. POR ESTA Y OTRAS RAZONES SE MANTIENE UNA HISTORIA DE LA ESTRUCTURA DEL PRODUCTO. ESTO PUEDE HACERSE CON UN NIVEL DE ENSAMBLADO. EL NIVEL DE ENSAMBLADO ES UN ENTERO CON VALOR INICIAL DE UNO, QUE SE INCREMENTA EN UNO CADA VEZ QUE CAMBIA UNA PARTE COMPUESTA. DE ESTA MANERA SE EVITA EL CAMBIAR EL NUMERO DE PARTE. ASUMIMOS QUE NO SE LLEGARA AL NIVEL 999. UNA PARTE COMPUESTA SE CAMBIA SI SE LE ANADE UNA PARTE NUEVA, QUE VENDRA DESDE CIERTO NIVEL, O SI SE LE QUITA UNA PARTE DESDE UN DETERMINADO NIVEL. DESIGNEMOS CON NIVELENTRADA EL PRIMER NIVEL EN EL QUE CIERTA SUBPARTE APARECE EN UNA PARTE COMPUESTA, Y CON NIVELSALIDA EL ULTIMO EN EL QUE APARECE. CONVENIMOS QUE UNA PARTE PERMANECERA MIENTRAS NO SE QUITE, POR LO QUE SU VALOR DE NIVELSALIDA ES INICIALMENTE DE 999.

INCORPORAMOS LOS CONCEPTOS ANTERIORES EN LA SIGUIENTE DESCRIPCION :

```
PARTES RANDOM DATA SET
     (
      NUMERODEPARTE       NUMBER(8);
      NIVELENSAMBLE       NUMBER(3); % ULTIMO, CORRIENTE
      DESCRIPCION         ALPHA(24);
     ) VERIFY NUMERODEPARTE > 0;
LLAVEPARTES ACCESS TO PARTES KEY(NUMERODEPARTE)
                                MODULUS 1500;
LISTADEPARTES DATA SET
     (
      NUMCOMPUESTA        NUMBER(8); % NUM. PARTE COMPUESTA
      NUMDESUBPARTE       NUMBER(8);
      NIVENSAMBSUBPARTE   NUMBER(3);;
      NUMITEM              NUMBER(3);
      CANTIDAD             REAL;
      NIVELENTRADA         NUMBER(3);
      NIVELSALIDA          NUMBER(3);
      NUMOPER              NUMBER(3);
     ) VERIFY NIVELENTRADA LEQ NIVELSALIDA AND
              NUMCOMPUESTA*NUMDESUBPARTE > 0;
ITEMSEUSAEN SET OF LISTADEPARTES
     KEY(NUMCOMPUESTA, NUMITEM, NIVELSALIDA DESCENDING)
     DATA(NIVELENTRADA);
SEENSAMBLA SET OF LISTADEPARTES
     KEY(NUMCOMPUESTA, NUMOPER, NUMITEM,
         NIVELSALIDA DESCENDING) DATA(NIVELENTRADA);
SEUSAEN SET OF LISTADEPARTES
     KEY(NUMDESUBPARTE, NUMCOMPUESTA, NUMITEM,
         NIVELSALIDA DESCENDING)
     ;

OPERACIONES DATA SET
     (
      NUMOPER              NUMBER(3);
      NUMCOMPUESTA         NUMBER(8);
     ) VERIFY NUMOPER*NUMCOMPUESTA > 0;
ENSAMBLADO SET OF OPERACIONES KEY(NUMCOMPUESTA, NUMOPER);
```

EN LOS ORDENES DEL CONJUNTO DE DATOS LISTADEPARTES, SE HIZO DE NIVELSALIDA UN ELEMENTO DE LLAVE DESCENDIENTE ANTICIPANDO QUE SE PROCESEN LAS PARTES CONSTITUYENTES PARA LOS NIVELES DE ENSAMBLADO MAS RECIENTES. SE INCLUYO NIVELENTRADA COMO DATOS EN LLAVE PARA CHECAR SIN LLEGAR AL REGISTRO DE LISTADEPARTES.

PARA OBTENER LA EXPANSION DE LA LISTA DE PARTES A TODOS LOS NIVELES PARA UNA PARTE DADA, EN UN NIVEL PARTICULAR DE ENSAMBLADO, SE USA EL SIGUIENTE PROCEDIMIENTO :

```
PROCEDURE QUEPARTESITEM(NUMERODEPARTE, NIVELENSAMBLE, NIVEL);
        VALUE NUMERODEPARTE, NIVELENSAMBLE, NIVEL;
        REAL NUMERODEPARTE, NIVELENSAMBLE, NIVEL;
    BEGIN
        BOOLEAN RSLT;
        REAL NUMITEM, NUMSUBPARTE, NIVENSAMBSUBPARTE,
             NIVELENTRADA, NIVELSALIDA;

        FIND LLAVEPARTES AT NUMERODEPARTE=NUMERODEPARTE;
        [PROCESA REGISTRO DE PARTES.]
        FIND KEY OF FIRST ITEMSEUSAEN AT
             NUMCOMPUESTA=NUMERODEPARTE :RSLT;
        WHILE NOT RSLT DO
        BEGIN
             GET LISTADEPARTES
                  (NUMITEM := NUMITEM
                   ,NIVELSALIDA := NIVELSALIDA
                   ,NIVELENTRADA := NIVELENTRADA
                   );
             IF NIVELSALIDA GEQ NIVELENSAMBLE THEN
             BEGIN
                  FIND ITEMSEUSAEN;
                  GET LISTADEPARTES
                    (NUMDESUBPARTE := NUMDESUBPARTE
                    ,NIVENSAMSUBPARTE := NIVELENSAMBSUBPARTE
                     );
             QUEPARTESITEM(NUMDESUBPARTE, NIVELENSAMBSUBPARTE,
                                NIVEL+1);
             % NOTA : LA RUTA DE ITEMSEUSAEN HA SIDO CAMBIADA
             % POR LA LLAMADA RECURSIVA. SE REESTABLECE LA
             % RUTA Y SE BRINCAN LOS NIVELES MENOS RECIENTES
             % DE ENSAMBLADO DE ESTE ELEMENTO.
                  FIND KEY OF FIRST ITEMSEUSAEN AT
                       NUMCOMPUESTA=NUMERODEPARTE AND
                       NUMITEM>NUMITEM :RSLT;
             END ELSE  % IGNORESE--DEMASIADO RECIENTE
                  FIND KEY OF NEXT ITEMSEUSAEN AT
                       NUMCOMPUESTA=NUMERODEPARTE :RSLT;
        END ELSE
        % SE BRINCAN NIVELES MENOS RECIENTES DE ESTE ELEMENTO
             FIND KEY OF NEXT ITEMSEUSAEN AT
                  NUMCOMPUESTA=NUMERODEPARTE AND
                  NUMITEM>NUMITEM :RSLT;
    END;
    IF REAL(RSLT).DMERROR NEQ NOTFOUND THEN
        DMTERMINATE(RSLT);
END QUEPARTESITEM;
```

ESTE PROCEDIMIENTO ES SOLO PARA LOS ULTIMOS NIVELES DE TODAS LAS PARTES. ESTO ES, SOLO SE INCLUYEN LOS REGISTROS DE LA LISTA DE PARTES CON NIVELSALIDA=999.

EL PROCEDIMIENTO PARA PROCESAR LOS PRODUCTOS EN DONDE SE USA UNA PARTE NO PUEDE EJECUTARSE PARA UN NIVEL DE ENSAMBLADO PARTICULAR, COMO EL ANTERIOR. SERIA COMO SIGUE :

```
PROCEDURE DONDESEUSA(NUMERODEPARTE, NIVEL);
     VALUE NUMERODEPARTE, NIVEL;
     REAL NUMERODEPARTE, NIVEL;
BEGIN
     BOOLEAN RSLT;
     REAL NIVELSALIDA, PARTEARRIBA, NUMITEM;

     FIND LLAVEPARTES AT NUMERODEPARTE=NUMERODEPARTE;
     [PROCESA REGISTRO DE PARTES.]
     FIND KEY OF FIRST SEUSAEN AT
          NUMDESUBPARTE=NUMERODEPARTE :RSLT;
     WHILE NOT RSLT DO
     BEGIN
          GET LISTADEPARTES
               (PARTEARRIBA := NUMCOMPUESTA
               ,NUMITEM := NUMITEM
               ,NIVELSALIDA := NIVELSALIDA
               );
          IF NIVELSALIDA=999 THEN
          BEGIN
               [FIND SEUSAEN SI SE DESEA PROCESAR
                EL REGISTRO DE LISTADEPARTES.]
               DONDESEUSA(PARTEARRIBA, NIVEL+1);
               % NOTA : LA LLAMADA RECURSIVA HA CAMBIADO LA
               % RUTA DE SEUSAEN. REESTABLEZCASE LA RUTA Y
               % BRINQUENSE LOS NIVELES DE ENSAMBLADO MENOS
               % RECIENTES.
               FIND KEY OF FIRST SEUSAEN AT
                    NUMDESUBPARTE=NUMERODEPARTE AND
                    NUMCOMPUESTA=PARTEARRIBA AND
                    NUMITEM>NUMITEM :RSLT;
          END ELSE
               % BRINQUENSE NIVELES DE ENSAMBLADO
               % MENOS RECIENTES
               FIND KEY OF NEXT SEUSAEN AT
                    NUMDESUBPARTE=NUMERODEPARTE AND
                    NUMCOMPUESTA=PARTEARRIBA AND
                    NUMITEM>NUMITEM :RSLT;
     END;
     IF REAL(RSLT).DMERROR NEQ NOTFOUND THEN
          DMTERMINATE(RSLT);
END DONDESEUSA;
```

EL CONJUNTO DE DATOS LISTADEPARTES TIENE CON LO ANTERIOR INFORMACION ADICIONAL ACERCA DE LA RELACION ENTRE DOS REGISTROS. ESO NO PUEDE HACERSE CON LIGAS, NI CON SUBORDENES MANUALES.

REDES.

CON LA TECNICA DE CONJUNTO DE DATOS DE RELACION, SE PUEDE REPRESENTAR CUALQUIER RED, CON UN CONJUNTO DE DATOS CON LOS NODOS DE LA RED, Y CON EL DE RELACION CON LOS ARCOS O CONEXIONES ENTRE LOS NODOS. LOS DOS ORDENES AUTOMATICOS CONTRA EL CONJUNTO RELACION, Y EL ORDEN DE ACCESO CONTRA EL CONJUNTO DE DATOS DE NODOS, PERMITEN RECORRER LA RED SIGUIENDO LOS ARCOS EN CUALQUIER DIRECCION.

V.4 COMPARACION DE ESTRUCTURAS.

HAY VARIAS RAZONES PARA PREFERIR ESTRUCTURAS DISJUNTAS. LOS CONJUNTOS DE DATOS DISJUNTOS PUEDEN SER REPRESENTADOS MEDIANTE TABLAS, QUE SON MAS SENCILLAS QUE LAS JERARQUIAS, QUE SON LO QUE RESULTA DE ESTRUCTURAS INCLUIDAS.

NO SE REQUIEREN ESTRUCTURAS INCLUIDAS PARA REPRESENTAR RELACIONES 1 A 1, 1 A N NI N A M, COMO YA MOSTRAMOS CON UN CONJUNTO DE DATOS RELACION. ESTO TAMBIEN PUEDE SER MEJOR. PARA LA ESTRUCTURA DE PARTES QUE TOMAMOS COMO EJEMPLO, UNA ESTRUCTURA CON SUBORDENES INCLUIDOS SERIA :

```
PARTES RANDOM DATA SET
            (
             NUMERODEPARTE      NUMBER(8);
             .
             .
             .
             SUBPARTES SUBSET OF PARTES KEY(NUMERODEPARTE);
             PARTEARRIBA SUBSET OF PARTES KEY(NUMERODEPARTE);
            );
LLAVEPARTES ACCESS TO PARTES KEY(NUMERODEPARTE)
                          MODULUS 1500;
```

LOS SUBORDENES INCLUIDOS NO PUEDEN REPRESENTAR LA INFORMACION ADICIONAL DE LA RELACION, COMO LA CANTIDAD, NIVEL DE ENSAMBLADO, NUMERO DE OPERACION, ETC., Y TAL INFORMACION ES NECESARIA. PERO AUN SI NO LO FUESE, TOMARIA DOS INSERCIONES, UNA EN CADA SUBORDEN, PARA OBTENER LA LISTA DE PARTES Y LAS CONEXIONES DE DONDE USADO, EN CONTRASTE CON LA CREACION-GUARDADO EN EL CONJUNTO DE DATOS LISTADEPARTES USADO ANTES. ESTO DA MAS POSIBILIDAD A ERROR. LA INSERCION TAMBIEN ES ALGO MAS COMPLICADA, DADO QUE PRIMERO SE HAN DE HACER CORRIENTES DOS REGISTROS. POR ULTIMO, LOS PROCEDIMIENTOS DE QUE PARTES SE USAN Y DONDE SE USA UNA PARTE PUDIESEN SER MENOS EFICIENTES USANDO LOS SUBORDENES, YA QUE AL ENCONTRAR UN REGISTRO DE PARTES POR MEDIO DE UN SUBORDEN CAMBIARIA EL REGISTRO CORRIENTE PARA EL AREA DE TRABAJO DE PARTES, Y EL REGISTRO VIEJO DE PARTES SE HABRIA DE HACER CORRIENTE DESPUES DE CADA RECUPERACION DE UNA SUBPARTE O DE UNA PARTE MAS ARRIBA, PARA PODER ACCESAR SU SUBORDEN OTRA VEZ, PARA PROSEGUIR CON LA SIGUIENTE PARTE HACIA ABAJO O HACIA ARRIBA.

OTRA CONSIDERACION ES DE LAS ESTRUCTURAS DISJUNTAS PUEDEN SER ACCESADAS DIRECTAMENTE, A DIFERENCIA DE LAS INCLUIDAS, PARA LAS CUALES UN REGISTRO MAESTRO HA DE HACERSE CORRIENTE PRIMERO. ASI, LAS ESTRUCTURAS INCLUIDAS IMPONEN OTRAS RESTRICCIONES A LA ESTRUCTURA DE LA BASE DE DATOS.

EN CUANTO A LA UTILIZACION DE ESPACIO EN DISCO, LAS ESTRUCTURAS INCLUIDAS TIENDEN A NO SER TAN EFICIENTES COMO LAS DISJUNTAS. LAS TABLAS DE INDICES NO COMPARTEN BLOQUES ENTRE PROPIETARIOS, Y TODOS LOS BLOQUES EN DMS II EMPIEZAN A FRONTERA DE BLOQUE. SUPONGASE QUE EL CONJUNTO DE DATOS PROPIETARIO TIENE UN MILLON DE REGISTROS, Y QUE CADA REGISTRO PROPIETARIO TIENE AL MENOS UNA ENTRADA DE LLAVE EN SU CONJUNTO INCLUIDO. HABRIA AL MENOS UN MILLON DE BLOQUES PARA EL CONJUNTO INCLUIDO. LOS CONJUNTOS DE DATOS ESTANDARES USAN EL ESPACIO EFICIENTEMENTE, COMPARTIENDO LOS BLOQUES ENTRE PROPIETARIOS, PERO SE REQUIERE AL MENOS UN SUBORDEN INCLUIDO. LOS CONJUNTOS DE DATOS DESORDENADOS INCLUIDOS NO REQUIEREN UN ORDEN CONTRA ELLOS, PERO NO COMPARTEN BLOQUES ENTRE PROPIETARIOS. LOS CONJUNTOS DE DATOS ORDENADOS INCLUIDOS SON LOS MAS EFICIENTES EN CUANTO A UTILIZACION DE ESPACIO EN DISCO, PERO NO PERMITEN LA RECUPERACION MEDIANTE ORDENES O SUBORDENES.

UNA MUY IMPORTANTE CONSIDERACION LA REPRESENTA LA RECUPERACION DE DESASTRES. LOS ORDENES Y SUBORDENES AUTOMATICOS CONTIENEN INFORMACION REDUNDANTE CIENTO POR CIENTO, Y PUEDEN SER RECREADOS DE SU CONJUNTO DE DATOS RAPIDA Y FACILMENTE USANDO EL PAQUETE DE REORGANIZACION DE DMS II. ESTO NO PUEDE HACERSE CUANDO SE TIENEN ESTRUCTURAS INCLUIDAS O LIGAS.

LOS VALORES DE LLAVE NO PUEDEN CAMBIARSE A MENOS DE QUE SE PERMITAN DUPLICADOS EN EL ORDEN O SUBORDEN. SI LO QUE SE QUIERE ES EFECTIVAMENTE NO DUPLICADOS, SE LE PUEDE DAR LA VUELTA NO PERMITIENDO DUPLICADOS, Y EJECUTANDO BAJA/RECREACION/CAMBIO DE VALORES DE LLAVE/ ALMACENAMIENTO. ESTO NO SIEMPRE PUEDE HACERSE PARA REGISTROS MAESTROS, YA QUE NOS PUEDEN DARSE DE BAJA A MENOS QUE SUS ESTRUCTURAS INCLUIDAS ESTEN VACIAS.

A MENUDO PODEMOS CONVERTIR UN CONJUNTO DE DATOS INCLUIDO EN DISJUNTO, ANADIENDOLE ELEMENTOS DE DATOS QUE ESPECIFIQUEN DE MANERA UNICA AL REGISTRO MAESTRO, Y ANTEPONIENDOLE LA LLAVE DE UN ORDEN CONTRA EL CON ESOS ELEMENTOS. POR EJEMPLO :

```
COSAS DATA SET
          (IDENTIFCOSA          NUMBER(3);
           .
           .
           .
           OTRAINFORM DATA SET
                     (
                      NUMERO                NUMBER(3);
                      INFORM                ALPHA(72);;
                     );
           ORDENOTRAINF SET OF OTRAINFORM KEY(NUMERO);
          );
ORDENDECOSAS SET OF COSAS KEY(IDENTIFCOSA);
```

LAS ESTRUCTURAS INCLUIDAS SE TRANSFORMARIAN A COMO SIGUE :

```
COSAS DATA SET
          (IDENTIFCOSA          NUMBER(3);
           .
           .
           .
          );
ORDENDECOSAS SET OF COSAS KEY(IDENTIFCOSA);

OTRAINFORM DATA SET
          (
           IDENTIFCOSA                    NUMBER(3);
           NUMERO                         NUMBER(3);
           INFORM                         ALPHA(72);
          );
ORDENOTRAINF SET OF OTRAINFORM KEY(IDENTIFCOSA, NUMERO);
```

LOS REGISTROS DE OTRA INFORMACION PARA UNA DETERMINADA COSA X, PODRIAN SEGUIRSE COMO SIGUE :

```
SET ORDENOTRAINF TO BEGINNING;
DO FIND NEXT ORDENOTRAINF AT IDENTIFCOSA=X :RSLT
UNTIL RSLT;
```

LIGAS Y SUBORDENES MANUALES.

LOS ORDENES SECUENCIALES CON INDICE Y LOS SUBORDENES AUTOMATICOS HAN DE PREFERIRSE SOBRE LAS LIGAS Y LOS SUBORDENES MANUALES, A LA LUZ DE VARIAS CONSIDERACIONES. PRIMERO, LAS LIGAS Y SUBORDENES MANUALES HAN DE ASIGNARSE POR PROGRAMA. TAMBIEN, EL SECUENCIAL CON INDICE VA ACOMODANDO EL ORDEN, QUE DE OTRO MODO HA DE HACERSE POR PROGRAMA. Y RESULTAN GENERALMENTE MENOS LLAMADAS A OPERACIONES DE MANEJO EN LOS PROGRAMAS DE APLICACION. FINALMENTE, SE PUEDE LLAMAR A LA MAYOR EFICIENCIA DE LAS LIGAS MAS APARENTE QUE REAL.

ILUSTRAREMOS ESTOS PUNTOS COMPARANDO LA ESTRUCTURA DE PRODUCTOS CONSIDERADA ANTES. YA SE DISCUTIO LA IMPLEMENTACION EQUIVALENTE USANDO SUBCONJUNTOS MANUALES, POR LO QUE PASAREMOS A USAR LIGAS. HABRIAN DOS LIGAS EN EL CONJUNTO DE DATOS DE PARTES, PRIMERAPARTE Y PRIMERUSO. HABRIAN CUATRO LIGAS EN EL CONJUNTO DE DATOS DE LISTADEPARTES, COMPUESTA Y COMPONENTE A REGISTROS DE PARTES, Y SIGCOMPUESTA Y SIGCOMPONENTE A REGISTROS DE LISTADEPARTES. ESTO EN DASDL :

```
PARTES RANDOM DATA SET
     (
      NUMERODEPARTE              NUMBER(8);
      PRIMERAPARTE               IS IN LISTADEPARTES COUNTED;
      PRIMERUSO                  IS IN LISTADEPARTES COUNTED;
      CUENTA                     COUNT(1000);
      .
      .
      .
     );
LLAVEPARTES ACCESS TO PARTES KEY NUMERODEPARTE
                                 MODULUS 1500;
LISTADEPARTES DATA SET
     (
      NUMCOMPUESTA               NUMBER(8);
      NUMDESUBPARTE              NUMBER(8);
      COMPUESTA                  IS IN PARTES COUNTED;
      COMPONENTE                 IS IN PARTES COUNTED;
      SIGCOMPUESTA               IS IN PARTES COUNTED;
      SIGCOMPONENTE              IS IN PARTES COUNTED;
      CUENTA                     COUNT(1000);
      .
      .
      .
     );
```

PARA ANADIR UN NUEVO REGISTRO X A LISTA DE PARTES, DEBE DE INSERTARSE UN NUEVO REGISTRO EN LA CADENA HACIA ABAJO, Y OTRO EN LA CADENA HACIA ARRIBA. SI SE DESEA MANTENER LAS LISTAS DE PARTES EN ORDEN DE NUMERO DE PARTE Y DESCENDENTE POR NUMERO DE NIVEL, LAS CADENAS HAN DE RECORRERSE, Y REASIGNARSE LAS LIGAS.

ES FACIL VER QUE ES VENTAJOSO UTILIZAR LOS ORDENES AUTOMATICOS DE DMS II, EN LUGAR DEL CODIGO DE APLICACION DE RELACIONES MEDIANTE LIGAS.

LA TECNICA DE USAR ESTRUCTURAS SECUENCIALES CON INDICE PUEDE CONSIDERARSE COMO LA FACTORIZACION DE LAS LIGAS, Y SU ORDENAMIENTO LOGICO, CON MANTENIMIENTO AUTOMATICO.

VI IMPLEMENTACION.

VI.1 DMALGOL.

DMALGOL ES UN LENGUAJE CREADO PARA DMS II, BASADO EN ALGOL. SE REQUERIA UN COMPILADOR ESPECIAL POR DOS RAZONES : SE NECESITABA USAR LA INFORMACION DEL ARCHIVO DESCRIPTOR PARA CADA BASE DE DATOS DEFINIDA, Y SE NECESITABAN ESTRUCTURAS DE STACK DIFERENTES A LAS DE ALGOL. EL SIMBOLICO DE BASE DE DATOS LO CONSTITUYEN ESQUELETOS DE RUTINAS DE ACCESO, CON COMPILACION CONDICIONAL. TOMANDO DEL ARCHIVO DESCRIPTOR LAS ESTRUCTURAS Y VALORES DE LAS PROPIEDADES, EL CODIGO SE OMITE, INCLUYE Y PARAMETRIZA. EN EL SIMBOLICO SE HACE REFERENCIA A VARIABLES NODO Y A PROPIEDADES DE NODOS, DONDE UN NODO CONSISTE DE UNA LISTA Y DE PROPIEDADES. LOS ELEMENTOS DE UNA LISTA PUEDEN A SU VEZ SER NODOS, LO QUE RESULTA EN GENERACION DE CODIGO A LA MEDIDA DE LAS ESTRUCTURAS.

AMBIENTES.

UN PROGRAMA EN ALGOL CONTIENE CODIGO DE INICIALIZACION QUE RESERVA ESPACIO EN EL STACK PARA SUS VARIABLES LOCALES. COMO LOS AMBIENTES DE LAS RUTINAS DE ACCESO NO ESTARAN CONTENIDAS EN UN STACK EN EJECUCION NO PUEDEN CONSTRUIR SU STACK. EN SU LUGAR, UN PROCEDIMIENTO EN EL SISTEMA OPERATIVO A TIEMPO DE APERTURA DE LA BASE DE DATOS DISPONE LAS VARIABLES DEL STACK BASANDOSE EN UNA IMAGEN DEL BLOQUE DE INFORMACION COLOCADA AL EFECTO EN EL ARCHIVO DE CODIGO DE LAS RUTINAS DE ACCESO. PARA CONSTRUIR ESA IMAGEN SE USA UNA DECLARACION DE AMBIENTE, QUE ESTABLECE LOS LIMITES DE CADA AMBIENTE DE MANERA MUY SIMILAR A COMO LO HACE UN PROCEDIMIENTO. EL NIVEL MAXIMO DE ANIDAMIENTO DE AMBIENTES ES DE TRES.

EL AMBIENTE EXTERNO CONTIENE VARIABLES GLOBALES, COMUNES A LA TOTALIDAD DE LA BASE DE DATOS ; EL INTERMEDIO CONTIENE VARIABLES UNICAS A CADA ESTRUCTURA, Y EL INTERNO CONTIENE VARIABLES UNICAS A UNA EVOCACION DE CADA ESTRUCTURA.

CADA ESTRUCTURA DE LA BASE TIENE UN AMBIENTE D3 Y UNO D4 COMPILADOS PARA ELLA. LAS MISMAS LINEAS DEL SIMBOLICO SON COMPILADAS MUCHAS VECES, AUN CUANDO EN ALGUNOS CASO ALGUNOS PROCEDIMIENTOS SON ESTABLECIDOS COMO DE COMPILACION UNICA, CON UNA PALABRA DE CONTROL PARA CADA ESTRUCTURA, APUNTANDO TODAS AL MISMO DESCRIPTOR DE SEGMENTO, QUE ES COMPARTIDO POR DIFERENTES AMBIENTES, COMPORTANDOSE COMO LOCAL EN CADA CASO.

RUTINAS DE ACCESO.

LA TECNICA BASICA USADA PARA ESCRIBIR LAS RUTINAS DE ACCESO ES LA COMPILACION CONDICIONAL. MEDIANTE ELLA, EL CODIGO SE OMITE, INCLUYE Y PARAMETRIZA SEGUN LOS VALORES Y ESTRUCTURAS EN EL ARCHIVO DESCRIPTOR. EN EL SIMBOLICO SE HACE REFERENCIA A VARIABLES NODO Y A PROPIEDADES DE NODOS. UN NODO ES UNA ESTRUCTURA DE DATOS QUE CONSISTE DE UNA LISTA Y DE UN CONJUNTO DE PROPIEDADES. LOS ELEMENTOS DE LA LISTA PUEDEN A SU VEZ SER NODOS. AL PRINCIPIO DE LA LISTA SE TIENE EL NUMERO DE MIEMBROS DE ELLA, Y SE CUENTA CON UNA CONSTRUCCION PARA ACCESARLOS TODOS UNO A UNO. LAS PROPIEDADES USADAS EN EL SIMBOLICO SE ENCUENTRAN EN EL ARCHIVO DESCRIPTOR, Y SE INCLUYEN LAS NECESARIAS, REFIERIDAS A LOS NODOS DE LA ESTRUCTURA DEFINIDA.

VI.2 PROTECCION CONTRA FALLAS.

DURANTE LA OPERACION DE CUALQUIER SISTEMA DE COMPUTO SE PRESENTAN CIRCUNSTANCIAS QUE PONEN EN PELIGRO LA INTEGRIDAD DE LA INFORMACION. PUEDE RESULTAR QUE UNA TRANSACCION NO LLEGUE A APLICARSE O SE APLIQUE INCORRECTAMENTE, O QUE EL ARCHIVO SE DANE, POR FALLAS DEL EQUIPO O DE LA PROGRAMACION.

LOS ARCHIVOS SON RECUPERABLES HASTA DISTINTOS NIVELES DE CALIDAD, CON DIFERENTES TECNICAS QUE CONLLEVAN CADA UNA UN SOBRECOSTO. LOS NIVELES DE CALIDAD SE REPRESENTAN MEDIANTE ESTADOS. UN ESTADO CORRECTO DE UN ARCHIVO ES CUANDO LA INFORMACION EN EL ES LA MAS RECIENTEMENTE INCORPORADA, Y NO SE ENCUENTRA INFORMACION YA ELIMINADA. UN ESTADO VALIDO ES CUANDO PARTE DE LA INFORMACION SE HA PERDIDO, PERO NO HAY INFORMACION ESPURIA. UN ESTADO CONSISTENTE ES CUANDO UN ESTADO VALIDO CUMPLE CON LOS CRITERIOS DE CONSISTENCIA DICTADOS POR LOS USUARIOS.

EN UN MISMO SISTEMA SE PUEDEN USAR VARIAS TECNICAS DE RECUPERACION PARA DEFERENTES FALLAS, ORDENADAS EN UNA JERARQUIA, DE LA MAS SENCILLA Y BARATA, CAPAZ DE RECUPERAR UN CONJUNTO PEQUENO DE FALLAS, HASTA LA MAS COMPLEJA Y CARA, CAPAZ DE RECUPERAR UN AMPLIO GRUPO DE FALLAS. EN ALGUNAS TECNICAS HAY UN COSTO SOLO SI SE PRESENTA LA FALLA A LA QUE SE APLICAN, Y EN OTRAS SE CONLLEVA SE PRESENTE O NO.

EN ALGUNAS APLICACIONES, EL VALOR DE LA INFORMACION AUMENTA CON EL TIEMPO, HASTA LLEGAR A UN PUNTO EN EL QUE SU INTEGRIDAD ES MAS IMPORTANTE QUE LA EFICIENCIA DE PROCESO.

LAS TECNICAS MAS USADAS PARA LA RECUPERACION DESPUES DE FALLAS, O RESISTENCIA CONTRA ELLAS, SON, EN ORDEN DE MAYOR A MENOR COSTO,

1) COPIAS MULTIPLES
2) TRAZA DE MODIFICACIONES
3) ARCHIVOS DIFERENCIALES
4) REEMPLAZO CUIDADOSO
5) PROTECCION INCREMENTAL
6) RESPALDO/VERSION ACTUAL
7) PROGRAMA DE SALVACION.

PARA FALLAS DE ALTO RIESGO, COMO PERDIDA IRRECUPERABLE DE AREAS EN DISCO, DE PROBABILIDAD SIEMPRE PRESENTE, SE HA DE CONTAR CON UN RESPALDO, QUE SE TOMA GENERALMENTE EN CINTAS. SI SE TOMA AL TERMINAR UN PROCESO, ES CONSIDERADO PROTECCION INCREMENTAL.

EL REEMPLAZO CUIDADOSO SE REFIERE A NO MODIFICAR LOS ARCHIVOS DIRECTAMENTE, SINO SOBRE UNA COPIA QUE EXISTE SOLO DURANTE LA ACTUALIZACION. EL USO DE ARCHIVOS DIFERENCIALES DEBE DE ESTAR CONTEMPLADO EN TODOS LOS PROGRAMAS DE ACTUALIZACION, QUE MODIFICAN EN UN ARCHIVO QUE HA DE INCORPORARSE AL PRINCIPAL DESPUES, Y DE CONSULTA, QUE BUSCAN PRIMERO EN EL ARCHIVO POR LAS MODIFICACIONES QUE AUN NO SE HAN INCORPORADO.

LA TRAZA DE MODIFICACIONES REGISTRA INFORMACION DE TODAS LAS TRANSACCIONES, Y PUEDE SER USADO PARA REGRESAR EL ARCHIVO A UN ESTADO ANTERIOR, O PARA RECUPERAR UN ESTADO CORRECTO APLICANDO LAS TRANSACCIONES NUEVAMENTE A UN ARCHIVO PREVIO.

LA TECNICA DE COPIAS MÚLTIPLES CONSISTE EN TENER VARIAS COPIAS DEL MISMO ARCHIVO ACTIVAS DURANTE EL PROCESO DE TRANSACCIONES, LAS CUALES SON IDENTICAS EXCEPTO DURANTE LA ACTUALIZACION.

DMS II INCORPORA AMPLIAS FACILIDADES DE TRAZA DE MODIFICACIONES Y DE RECUPERACION AUTOMATICA, DESDE PARADO-ARRANCADO DE LA MAQUINA, RECONSTRUYENDO PARTES DE LA BASE, Y REMOVIENDO TRANSACCIONES ABORTADAS. EN LA TRAZA DE MODIFICACIONES SE ESPECIFICAN A INTERVALOS REGULARES PUNTOS DE SINCRONIA Y DE CONTROL. LOS PUNTOS DE SINCRONIA SE USAN PARA RECUPERACION DE CONDICIONES DE PARADO-ARRANCADO Y DE TRANSACCIONES ABORTADAS, Y SON GRABADOS CADA DETERMINADO NUMERO DE TRANSACCIONES. ESTE NUMERO ES DETERMINADO POR EL USUARIO. LOS PUNTOS DE CONTROL INCLUYEN UN NUMERO TAMBIEN ESPECIFICADO POR EL USUARIO DE PUNTOS DE SINCRONIA, SON TOMADOS EN LA CINTA DE TRAZA E INCLUYEN EL VACIADO DE LAS AREAS DE ENTRADA/SALIDA EN MEMORIA. ESTAS CON VACIADAS A LO MENOS CADA DOS PUNTOS DE CONTROL, PARA FACILITAR LA RECUPERACION.

EN CASO DE FALLA DEL SISTEMA, DESPUES DEL PARADO-ARRANCADO LA RECUPERACION ES INICIADA AUTOMATICAMENTE. SE BUSCAN LAS IMAGENES DE ANTES DE MODIFICACION HACIA ATRAS EN LA TRAZA HASTA EL ULTIMO PUNTO DE SINCRONIA PARA RESTAURAR LA BASE DE DATOS A SU CONDICION ORIGINAL, Y SE USAN LAS IMAGENES DE DESPUES PARA ACTUALIZARLA CON LOS RESULTADOS DE LAS TRANSACCIONES COMPLETADAS. PARA LA RECUPERACION DE TRANSACCIONES ABORTADAS SE USA UN CONJUNTO DE DATOS DE REINICIO, QUE CONTIENE AREAS DE REGISTRO PARA REINICIO DE CADA PROGRAMA. CONCURRENTEMENTE CON EL PROCESO DE LA BASE DE DATOS SE OPERAN EL VACIADO EN LINEA Y UNA UTILERIA DE RECUPERACION DE DATOS, QUE CREAN COPIAS DE RESPALDO DE PAGINAS EN LA BASE PARA RECUPERACION DE ERRORES DE DISCO. SE SINCRONIZAN LA TRAZA DE MODIFICACIONES Y EL VACIADO. EN CASO DE UN ERROR IRRECUPERABLE EN DISCO, PUEDE BLOQUEARSE LA PAGINA AFECTADA, CON LO QUE EL PROCESO CONTRA LO DEMAS PUEDE CONTINUAR, O EL OPERADOR PUEDE INVOCAR LA UTILERIA DE RECONSTRUCCION PARA RECARGAR LA PAGINA AFECTADA, RETRASANDOSE O POSPONIENDOSE EL PROCESO DE TRANSACCIONES. CUANDO SE TERMINA LA RECONSTRUCCION DE LA PAGINA, AUTOMATICAMENTE SE HACE DISPONIBLE PARA PROCESO.

TRAZA DE MODIFICACIONES.

LA TRAZA DE MODIFICACIONES ES UNA SECUENCIA DE IMAGENES DE ANTES Y DE DESPUES DE CAMBIOS A LA BASE DE DATOS, ADEMAS DE ALGUNOS REGISTROS DE CONTROL. SE USA PARA RECUPERAR LA BASE DESPUES DE PARADO-ARRANCADO, PARA DAR INFORMACION DE REINICIO A LOS PROGRAMAS, PARA RECONSTRUIR PARTES PERDIDAS DE LA BASE POR ERRORES DE CIRCUITERIA, Y PARA REGRESAR TRANSACCIONES ABORTADAS.

SE PUEDE ESPECIFICAR CUANTAS TRANSACCIONES HABRA ENTRE PUNTOS DE SINCRONIA, Y CUANTOS PUNTOS DE SINCRONIA HABRA ENTRE PUNTOS DE CONTROL. A PUNTO DE SINCRONIA NINGUN PROGRAMA DE APLICACION SE TENDRA A MITAD DE TRANSACCION, Y A PUNTO DE SINCRONIA SE REGRESARA LA BASE EN RECUPERACION DE PARADO-ARRANCADO, O AL ABORTAR UNA TRANSACCION. A PUNTO DE CONTROL SE MARCAN LAS AREAS DE ENTRADA/SALIDA MODIFICADAS EN MEMORIA, Y SI YA ESTAN MARCADAS, SE VACIAN A ALMACENAMIENTO. DE ESTA MANERA, LA RECUPERACION DE PARADO-ARRANCADO NO NECESITA REGRESAR MAS DE DOS PUNTOS DE CONTROL EN LA TRAZA.

LA TRAZA CONSISTE EN UNA SERIE DE ARCHIVOS DE NOMBRE <NOMBRE DE LA BASE>/AUDIT<NUMERO DE ARCHIVO> QUE VAN SIENDO CREADOS NUEVOS A MEDIDA QUE SON NECESARIOS. SE PUEDEN ESPECIFICAR SUS ATRIBUTOS, COMO NUMERO Y TAMANO DE AREAS, ETC. LOS PARAMETROS MAS CRITICOS PARA LA EFICIENCIA SON EL TAMANO DE BLOQUE, Y LA FRECUENCIA DE LOS PUNTOS DE SINCRONIA. SE HA DE LOGRAR UN EQUILIBRIO ENTRE EL SOBRECOSTO DE MANTENER EN MEMORIA LAS AREAS PARA ESTOS ARCHIVOS Y EL TIEMPO PARA ESCRIBIRLOS POR UN LADO, Y EL PROCESO DE RECUPERACION EN PARADO-ARRANCADO O DE ABORTO DE TRANSACCION INVOLUCRANDO DEMASIADAS TRANSACCIONES POR EL OTRO.

LA RECUPERACION DE PARADO-ARRANCADO OCURRE AUTOMATICAMENTE. TODOS LOS TRABAJOS SOBRE LA BASE ESPERARAN EN SU INSTRUCCION DE OPEN HASTA TERMINAR LA RECUPERACION, QUE OCURRE EN CUATRO FASES : SE ENCUENTRA EL FIN DE LATRAZA. SE BUSCA HACIA ATRAS HASTA EL ULTIMO PUNTO DE SINCRONIA, REGRESANDO LA BASE DESDE EL USANDO LAS IMAGENES DE ANTES DE CAMBIO. SE ENCUENTRA EL SEGUNDO PUNTO DE CONTROL DESDE EL FIN DE LA TRAZA, Y SE AVANZA DESDE EL, ASEGURANDO QUE TODAS LAS TRANSACCIONES ANTERIORES AL ULTIMO PUNTO DE SINCRONIA SE COMPLETEN, USANDO LAS IMAGENES DE DESPUES DE CAMBIO. SE GUARDAN EN EL CONJUNTO DE DATOS DE REINICIO LAS AREAS CORRESPONDIENTES A LAS ULTIMAS TRANSACCIONES COMPLETADAS. POR ULTIMO, SE TERMINA EL ARCHIVO DE TRAZA CON UN REGISTRO ESPECIAL, Y SE DESBLOQUEA LA BASE.

EL PROCESO DE TRANSACCIONES BASA LA RECUPERACION Y EL REINICIO, Y ES OBLIGATORIO PARA CAMBIO DE INFORMACION CUANDO A LA BASE DE DATOS SE LE DECLARO TRAZA. EL STACK Y LA BASE DE DATOS INTERNA ENTRAN EN ESTADO DE TRANSACCION AL EJECUTAR BEGIN-TRANSACTION, Y SALEN CON END-TRANSACTION. TAMBIEN AL SALIR DEL ESTADO DE TRANSACCION SE DESBLOQUEAN TODOS LOS REGISTROS BLOQUEADOS.

COMO PARTE DEL PROCESO DE ENTRADA AL ESTADO DE TRANSACCION, EL AREA DE REGISTRO CORRIENTE DEL CONJUNTO DE DATOS DE REINICIO, QUE TIENE LA INFORMACION MINIMA SUFICIENTE PARA EL REINICIO, SE ESCRIBE A LA TRAZA DE MODIFICACIONES.

ABORTO DE TRANSACCIONES.

EN GENERAL, NO SE PUEDE REGRESAR UNA TRANSACCION INCOMPLETA SIN INVALIDAR OTRAS TRANSACCIONES SIENDO PROCESADAS CONCURRENTEMENTE, DADO QUE LOS REGISTROS CAMBIADOS NO SON MANTENIDOS BLOQUEADOS HASTA EL FIN DE LA TRANSACCION. POR LO QUE ABORTAR UNA TRANSACCION PUEDE AFECTAR A TODOS LOS PROGRAMAS CORRIENDO CONTRA LA BASE. LA FUNCION DE ABORTO DE TRANSACCION ES INVOCADA IMPLICITAMENTE CUANDO UN PROGRAMA TERMINA O CIERRA UNA BASE DE DATOS INTERNA QUE ESTE AUN EN ESTADO DE TRANSACCION.

SE USAN LAS IMAGENES DE ANTES DE CAMBIO EN LA TRAZA PARA REGRESAR TODAS LAS TRANSACCIONES DESDE EL ULTIMO PUNTO DE SINCRONIA. ANTES DEL ABORTO SE OBLIGA UN SEUDO-PUNTO DE SINCRONIA. EL CONJUNTO DE DATOS DE REINICIO RECIBE LAS AREAS DE REINICIO PARA LAS ULTIMAS TRANSACCIONES COMPLETADAS, Y LOS PROGRAMAS AFECTADOS CON UNA EXCEPCION SON INFORMADOS QUE HAN DE USAR ESTA INFORMACION PARA REHACER LAS TRANSACCIONES REGRESADAS.

CADA BASE DE DATOS CON TRAZA DE MODIFICACIONES HABRA DE TENER PUES, UN CONJUNTO DE DATOS DE REINICIO. EN EL, SE TENDRAN TIPO DE REINICIO Y CUENTA DE TRANSACCIONES, PROPORCIONADOS POR EL SISTEMA, Y UNO O VARIOS ITEMS DE DATOS CON INFORMACION SUFICIENTE PARA EL REINICIO DE CADA PROGRAMA DE APLICACION.

VI.3 UTILERIAS.

CONTROL DE LA BASE DE DATOS.

TODAS LAS BASES DE DATOS EN DMS II INCLUYEN CADA UNA UN ARCHIVO CON INFORMACION DE CONTROL, DE MANTENIMIENTO AUTOMATICO. ESTE ARCHIVO DEBE DE ESTAR PRESENTE SIEMPRE QUE SE USA LA BASE QUE CONTROLA. ES USADO PARA CONTROL DE INTERBLOQUEO, PARA CHEQUEO DE COMPATIBILIDAD ENTRE LOS ARCHIVOS Y LOS PROGRAMAS, Y PARA CHECAR QUE LOS ARCHIVOS DE DATOS CORRESPONDAN TODOS A UN MISMO NIVEL DE ACTUALIZACION. CONTIENE TAMBIEN OTROS DATOS, ENTRE LOS CUALES SE ENCUENTRAN INFORMACION DE CONTROL DE TRAZA DE MODIFICACIONES, Y PARAMETROS DINAMICOS.

REORGANIZACION.

OCASIONALMENTE CAMBIAN LAS SOLICITUDES A UNA BASE DE DATOS. PARA CUANDO SE REQUIEREN CAMBIOS FISICOS A LOS ARCHIVOS O A LOS REGISTROS, UNA UTILERIA A LA MEDIDA DE LAS ESTRUCTURAS ES GENERADA PARA LA LA REORGANIZACION. LAS SIGUIENTES FACILIDADES ESTAN DISPONIBLES :

1. REORGANIZACION DE CONJUNTOS DE DATOS DE ACUERDO A UN ORDEN DADO, SI A TRAVES DE TAL ORDEN ES COMO SE PRESENTAN EL MAYOR NUMERO DE APLICACIONES.

2. CONSOLIDACION DE ESPACIO DADO DE BAJA O NO USADO, PARA REINTEGRARLO AL SISTEMA.

3. GENERACION DE ORDENES O SUBORDENES AUTOMATICOS, SI SE DESEA UN NUEVO METODO DE ACCESO.

4. BALANCEO DE TABLAS DE INDICES, BUSCANDO UNA DISTRIBUCION UNIFORME EN TABLAS FINAS Y GRUESAS PARA OPTIMIZAR LOS ACCESOS A TRAVES DE ELLAS.

5. CAMBIO EN LOS ATRIBUTOS FISICOS DE UNA ESTRUCTURA, COMO EN AREAS, POBLACION, MODULO Y FACTOR DE CARGA.

6. CAMBIO EN LA DESCRIPCION DE REGISTRO DE UN CONJUNTO DE DATOS DE UN REGISTRO GLOBAL, ANADIENDO O QUITANDO ELEMENTOS.

UTILERIA.

ESTE PROGRAMA PROPORCIONA UN METODO CONVENIENTE DE CONTROLAR LOS ASPECTOS OPERACIONALES DE UNA BASE DE DATOS, AL PROPORCIONAR LAS SIGUIENTES CAPACIDADES :

1. VACIADO Y CARGA DE ARCHIVOS DE LA BASE.

2. IMPRESION DE DATOS Y DE INFORMACION DE CONTROL DE ARCHIVOS DE LA BASE.

3. INICIALIZACION DE ESTRUCTURAS DE LA BASE.

4. ARRANQUE DE FORMAS NO AUTOMATICAS DE RECUPERACION PARA BASES DE DATOS CON AUDITORIA AUTOMATICA.

DIRECTORIO DE CINTAS DE VACIADO.

PARA UNA MAS FACIL RECUPERACION DE LA BASE, DMS II PUEDE MANTENER UN DIRECTORIO CON INFORMACION ACERCA DE LAS CINTAS DE VACIADO, PARA LA DETERMINACION DE EN QUE CINTA SE ENCUENTRA UNA FILA DE INFORMACION.

CONTROL DE PARTICION.

ESTA FUNCION ES NECESARIA A PARTIR DEL USO DE LA CAPACIDAD DE DMS II DE PARTIR LA INFORMACION DE LA BASE EN PORCIONES, DE MODO DE NECESITARSE EN LINEA SOLO AQUELLAS REQUERIDAS POR LA APLICACION CORRIENTE. ESTO ES, SE ESTABLECEN CRITERIOS A NIVEL DE ORDEN, Y EL SISTEMA SOLICITA LA CINTA O PAQUETE CORRESPONDIENTE.

CARGA Y VACIADO.

SE CUENTA CON UNA UTILERIA QUE GENERA PROGRAMAS PARA PASAR DE Y A ARCHIVOS CONVENCIONALES CONJUNTOS DE DATOS DE LA BASE. EL USUARIO PROPORCIONA EL NOMBRE DEL CONJUNTO DE DATOS Y UN NOMBRE DE ARCHIVO DESCRIPTOR.

EL ARCHIVO CONVENCIONAL SE DESCRIBE USANDO UNA DESCRIPCION DE ARCHIVO DE COBOL. EN CARGA, LOS ELEMENTOS EN LA DESCRIPCION SON TRANSFERIDOS A LOS ELEMENTOS CON EL MISMO NOMBRE EN EL CONJUNTO DE DATOS. EL VACIADO ES APLICADO A LA INVERSA. SI LOS NOMBRES DE LOS ELEMENTOS EN EL CONJUNTO DE DATOS Y EN LA DESCRIPCION DEL ARCHIVO NO SON IDENTICOS, PUEDEN SENALARSE COMO IGUALES.

DBS VISIBLE.

ESTA ES UNA CAPACIDAD QUE PERMITE AL USUARIO COMUNICARSE DIRECTAMENTE CON LA BASE DE DATOS, PARA TENER CONTROL DINAMICO SOBRE LOS RECURSOS DEL SISTEMA, EN TERMINOS GLOBALES O ESTRUCTURA POR ESTRUCTURA. SE PUEDE INTERROGAR EL STATUS DE CUALQUIERA ESTRUCTURA FISICA EN LA BASE, Y SE PUEDEN CAMBIAR LOS VALORES DE REBLOQUEO, DE FACTOR DE REBLOQUEO Y LAS ESPECIFICACIONES DE ESPACIO PARA REGISTRO PARA UNA ESTRUCTURA.

COPIA DE AUDITORIA.

SE USA ESTA CAPACIDAD PARA VERIFICAR ARCHIVOS DE AUDITORIA, Y PARA COPIARLOS DE UN MEDIO DE ALMACENAMIENTO A OTRO.

"INQUIRY".

UN METODO MEDIANTE EL CUAL UN USUARIO PUEDE EN UNA TERMINAL EXAMINAR INFORMACION DE UNA BASE DE DATOS EN DMS II LO CONSTITUYE "INQUIRY", QUE SE CONSTRUYE DADO UN ARCHIVO DESCRIPTOR, SELECCIONANDO QUE USUARIOS TIENEN ACCESO A QUE PARTES DE LA BASE. SU FUNCION BASICA ES SELECCIONAR UN REGISTRO EN LA BASE Y DESPLEGAR ITEMS DE EL, AUN CUANDO SE PUEDE GENERAR CON LA CAPACIDAD DE ACTUALIZAR INFORMACION, Y SE PUEDE EXAMINAR LA DESCRIPCION DE LA BASE. CONSTA DE RELATIVAMENTE POCAS INSTRUCCIONES, PERO PUEDEN SER COMBINADAS PARA EJECUTAR OPERACIONES COMPLEJAS. SIEMPRE OBTENDRA LA INFORMACION REQUERIDA, AUN SEA A TRAVES DE BUSQUEDAS LINEALES EN LOS ARCHIVOS, PERO TOMARA VENTAJA DE LOS ORDENES CUANDO SEA POSIBLE.

VI.4 CONTROL DE ACCESO A LAS BASES DE DATOS.

EL PROBLEMA DE ASIGNAR SEGURIDAD A LA BASE DE DATOS APARECE DE LA TENDENCIA A SOBREASEGURAR LOS ARCHIVOS HASTA UN PUNTO TAL DE IMPEDIR LA EJECUCION DE CIERTAS FUNCIONES NECESARIAS. LOS ARCHIVOS DE UNA BASE SON ACCESADOS POR LOS PROGRAMAS DE USUARIO MEDIANTE LAS RUTINAS DE ACCESO, Y TAMBIEN POR PROGRAMAS DEL SISTEMA, COMO LOS DE UTILERIA Y RECUPERACION. DE MODO QUE EL PROBLEMA SE REDUCE A IDENTIFICAR LAS NECESIDADES DE ACCESO DE TODOS LOS PROGRAMAS Y USUARIOS, Y A ASIGNAR LA SEGURIDAD ADECUADA.

HAY TRES MANERAS DE ACCESAR EL CONTENIDO DE UNA BASE DE DATOS, A SABER : A TRAVES DE UNA BASE DE DATOS LOGICA, A TRAVES DE LA BASE DE DATOS FISICA Y A TRAVES DE LOS ARCHIVOS FISICOS.

EL ACCESO A LOS DATOS PRESENTADOS EN UNA BASE DE DATOS LOGICA ES CONTROLADO ESTRICTAMENTE POR LAS RUTINAS DE ACCESO. EL LUGAR DE DEJARLA COMO PUBLICA, SE HA DE RESTINGIR EL ACCESO SEGUN ESPECIFICACIONES EN UN ARCHIVO GUARDIAN. EL ACCESO ESTARA CONTROLADO AUN A LOS USUARIOS CON CLAVE PRIVILEGIADA ; ESTO DARA LA MAYOR SEGURIDAD POSIBLE ADEMAS DE CIERTA INDEPENDENCIA DE DATOS, A SABER, LA PREVISTA EN EL DISENO DE LAS BASES LOGICAS.

EL ACCESO A LA BASE DE DATOS FISICA TAMBIEN ES CONTROLADO POR LAS RUTINAS DE ACCESO. DE DEJARSE PUBLICA, CUALQUIER USUARIO TENDRIA ACCESO A TODAS LAS COMPONENTES DE LA BASE. SE HA DE ESPECIFICAR EN UN ARCHIVO GUARDIAN QUE EL ACCESO ESTE CONDICIONADO EXCLUSIVAMENTE AL ADMINISTRADOR, YA QUE DE NO HACERSE ASI SE PERDERIA POR COMPLETO LA INDEPENDENCIA DE LOS DATOS.

EL MAS IMPORTANTE ACCESO A CONTROLAR ES AQUEL A LOS ARCHIVOS FISICOS QUE CONSTITUYEN LA BASE. NUNCA HABRIA DE PERMITIRSE A LOS USUARIOS EL ESCRIBIR DIRECTAMENTE A ESTOS ARCHIVOS, A RIESGO DE CORROMPER LOS DATOS. PERO NO ES POSIBLE SIMPLEMENTE RESTRINGIR INDISCRIMINADAMENTE EL ACCESO, DADAS LAS NECESIDADES DE RECUPERACION.

ALGUNOS PROGRAMAS DEL SISTEMA ACCESAN DIRECTAMENTE LOS ARCHIVOS FISICOS, A SABER, UTILERIA, RECUPERACION, RECUPERACION DE DATOS, Y ABORTO. SOLO SE PUEDE CONTROLAR COMO Y CUANDO CORREN UTILERIA Y RECUPERACION DE DATOS. LOS OTROS DOS CORREN CUANDO SON NECESARIOS, BAJO LA CLAVE DE LA TAREA DEL USUARIO, ASUMIENDO CONSECUENTEMENTE SUS PRIVILEGIOS DE SEGURIDAD. LA RECUPERACION JAMAS SE COMPLETARA POR UN ERROR DE SEGURIDAD, SI EL SUSODICHO USUARIO NO TIENE ACCESO DE LECTURA/ ESCRITURA A TODOS LOS ARCHIVOS QUE COMPONEN LA BASE.

SE HA DE PERMITIR PUES EL ACCESO LIBERAL PARA LA RECUPERACION, MIENTRAS SE HA DE RESTRINGIR ESTRICTAMENTE PARA MANTENER LA CONFIABILIDAD DE LOS DATOS. EL MEJOR ENFOQUE ES EL DE MANTENER TRES CONJUNTOS DE ARCHIVOS GUARDIANES, UNO PARA CADA MODO DE ACCESO. LOS DE LAS BASES DE DATOS LOGICAS DESIGNARIAN A LOS USUARIOS LEGALES DE CADA UNA. EL DE LA BASE DE DATOS FISICA PERMITIRIA EL ACCESO SOLO AL ADMINISTRADOR ; Y EL DE LOS ARCHIVOS FISICOS CONTEMPLARIA SOLAMENTE A LOS PROGRAMAS DE UTILERIA, RECUPERACION, RECUPERACION DE DATOS, RECONTRUCCION Y ABORTO.

CONSTRUCCION DE UN ARCHIVO GUARDIAN.

LA COMPLICACION QUE APARECE AL CONSTRUIR UN ARCHIVO GUARDIAN SE REFIERE AL MANEJO QUE EL SISTEMA HACE DE LOS CONFLICTOS. SI A UNA CLAVE DE USUARIO SE LE DA UN CONJUNTO DE PRIVILEGIOS DE ACCESO, Y A UN PROGRAMA SE LE DA OTRO DIFERENTE, AL CORRER ESE PROGRAMA BAJO ESA CLAVE DE USUARIO SE PRESENTA UN CONFLICTO, QUE EL SISTEMA RESUELVE ASUMIENDO AQUELLOS PRIVILEGIOS QUE APARECIERON PRIMERO EN EL ARCHIVO, FUESEN PARA EL USUARIO O PARA EL PROGRAMA, E IGNORANDO LAS ESPECIFICACIONES SUBSECUENTES.

EN GENERAL, UN ARCHIVO GUARDIAN SE CONSTRUIRIA ASI :

```
DEFAULT = NO;
PROGRAMA *UTILITY/<NOMBRE DE LA BASE DE DATOS> = RW;
PROGRAMA *RECOVERY/<NOMBRE DE LA BASE> = RW;
PROGRAMA *DATARECOVERY/<NOMBRE DE LA BASE> = RW;
PROGRAMA *RECONSTRUCT/<NOMBRE DE LA BASE> = RW;
PROGRAMA *ABORT/<NOMBRE DE LA BASE> = RW;
TODOS LOS PROGRAMAS DE ACTUALIZACION DE LA BASE = RW;
CLAVES DE ACCESO A TENER ACCESO DE LECTURA/ESCRITURA = RW;
TODOS LOS PROGRAMAS DE CONSULTAS A LA BASE = RO;
CLAVES DE USUARIO A TENER ACCESO DE LECTURA = RO;
```

DE ESTA MANERA, QUEDAN ESPECIFICADAS EXACTAMENTE LAS RESTRICCIONES DE SEGURIDAD QUE SE DESEAN PARA CADA BASE DE DATOS LOGICA, CADA CLAVE DE USUARIO Y CADA PROGRAMA.

VII CONCLUSIONES.

ES FACIL A LA LUZ DE LA INFORMACION PRESENTADA RECONOCERLE A DMS II SU RIQUEZA DE POSIBILIDADES. NO SE PUEDE TAMPOCO NEGAR LA NECESIDAD DE DAR AL DISENO DE LAS BASES EL MAS ESTRICTO CUIDADO, SIEMPRE EVALUANDO EN TERMINOS DE COSTOS LAS DISTINTAS SOLICITUDES DE APLICACION. SE HA DE PROCURAR MANTENER EL DISENO EN EL MINIMO ESENCIAL, Y SIEMPRE CONSIDERAR LAS MULTIPLES ALTERNATIVAS, PREVIENDO DESDE LUEGO LA CONSERVACION O RECUPERABILIDAD DE LA INTEGRIDAD DE LA BASE Y LA SENCILLEZ DEL CODIGO DE APLICACION.

EL PROVEEDOR PARA DMS II TOMO CONTRA EL ENFOQUE TRADICIONAL SIN BASES DE DATOS, ESTO ES, DE DEFINICION DE DATOS EN LOS PROGRAMAS, Y CONTRA EL ENFOQUE TRADICIONAL CON BASES DE DATOS, A SABER, DE INTERPRETACION DE FORMATOS A TIEMPO DE EJECUCION, UN INGENIOSO ENFOQUE CON LAS RUTINAS REENTRANTES. PREPARADAS EN DMALGOL EN EL SIMBOLICO DE BASE DE DATOS, TIENEN TODAS LAS VENTAJAS DE USO DE UN LENGUAJE DE ALTO NIVEL ; Y, AJUNTADAS PRECISAMENTE A LA DESCRIPCION DE LAS ESTRUCTURAS, CONSIGUEN UN NIVEL DE EFICIENCIA DIFICIL DE SUPERAR.

ASI DE CERCA COMO SE SIGUEN LAS CARACTERISTICAS MAS EFICIENTES DEL EQUIPO POR PARTE DE DMS II, ASI EL USUARIO HA DE SEGUIR LAS DE DMS II EN SUS APLICACIONES. ESTA DADA TAMBIEN LA POSIBILIDAD DE IMITAR EL DETALLE DEL MANEJO INTERNO, PARA EL CASO DE LAS ESTRUCTURAS FISICAS.

ESTAMOS VIENDO EL FUTURO DE LOS EQUIPOS DE COMPUTO : LA MEMORIA SE ABARATARA HASTA HACER LOS MEDIOS PRESENTES DE ALMACENAMIENTO INCOSTEABLES EN TERMINOS DE MANTENIMIENTO Y RIESGO, LOS SISTEMAS OPERATIVOS SE INTEGRARAN A LA MEMORIA, LOS MANEJADORES RELACIONALES SE IMPONDRAN SOBRE LOS PRESENTEMENTE OFRECIDOS, Y LA INTELIGENCIA ARTIFICIAL VENDRA A IMPONERSE SOBRE LA MAYORIA DE LOS TRABAJADORES HUMANOS EN ESTE CAMPO. PERO ESO NO SUCEDERA MANANA, Y MIENTRAS, LOS SISTEMAS Y EQUIPOS SEGUIRAN REQUIRIENDO EVALUACION, APROVECHAMIENTO Y OPTIMIZACION DE USO. POR UN LADO, SE SEGUIRA REQUIRIENDO ENTENDER AL USUARIO FINAL, Y POR EL OTRO, SE MANTENDRA LA NECESIDAD DE ENTENDER A LA COMPUTADORA.

VIII BIBLIOGRAFIA.

BURROUGHS CORP. :

AN INTRODUCTION TO BURROUGHS B6700 SYSTEMS.

DATA MANAGEMENT SYSTEM INFORMATION MANUAL.

DMS II DATA AND STRUCTURE DEFINITION LANGUAGE
(DASDL) REFERENCE MANUAL.

DMS II HOST LANGUAGE INTERFACE
REFERENCE MANUAL.

DMS II UTILITIES AND OPERATIONS GUIDE.

DMALGOL IMPLEMENTATION B6000 SERIES
MARK 32 APPENDIX A.

DONALD GREGORY, CUBE SAN FRANCISCO 1980 :

SAFE AND WORKABLE SECURITY FOR DMS II DATA BASES.

DATAPRO RESEARCH CORPORATION, 1977 :

A BUYER'S GUIDE TO DATA BASE MANAGEMENT SYSTEMS.

DMS II BURROUGHS CORP. SOFTWARE.

COMPUTER SURVEYS :

THIRD GENERATION COMPUTER SYSTEMS
VOL. 3 N. 4.

JOHN DONOVAN :

SYSTEMS PROGRAMMING.

PHILIP A. BERNSTEIN, 1980 :

DATABASE SYSTEM TECHNOLOGY.

FUNDACION ARTURO ROSEBLUETH
PARA EL AVANCE DE LA CIENCIA, A. C., 1977 :

SEMINARIO DE ESTRUCTURAS Y BASES DE DATOS.