

MAESTRIA EN INFORMATICA ADMINISTRATIVA

TEMA: LENGUAJES DE CUARTA GENERACION

TM
Z7164
.C8
FCPYA
1988
S2

TM

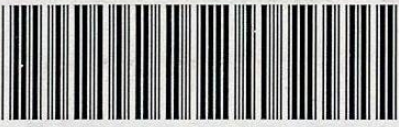
Z7164

.C8

FCPYA

1988

S2



1020073578

UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE CONTADURIA PUBLICA Y ADMINISTRACION
SECRETARIA DE POST-GRADO

MAESTRIA EN INFORMATICA ADMINISTRATIVA
TEMA: LENGUAJES DE CUARTA GENERACION

ASESOR: ING. MARIO ALBERTO VAZQUEZ MURAIRA
AUTOR: LIC. ELISA SANCHEZ CABELLAS

MONTERREY, N. L.

JUNIO DE 1988

TM
Z7164
.C8
FCP4A
1988
S2



FONDO TESIS

62975

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

AGRADEZCO SINCERAMENTE:

Al ING. MARIO ALBERTO VAZQUEZ, asesor del trabajo de investigación presentado como tesis, su apoyo y recomendaciones.

Al ING. MARIANO MACIAS, Coordinador de la Maestría en "INFORMATICA ADMINISTRATIVA" de nuestra Facultad, sus atenciones así como el haber despertado en mí, al impartir su cátedra, el interés por esta rama del conocimiento.

Al LIC. SERGIO H. MARTINEZ FLORES, Director de la Biblioteca central del ITESM por el invaluable apoyo en material de consulta para realizar la investigación.

DEDICATORIA

A MI ESPOSO ING. JESUS VALADEZ BOTELLO
POR SU APOYO MORAL Y COMPRENSION; POR
PERMITIRME REALIZARME COMO PROFESIONISTA.

INDICE

1.	INTRODUCCION	1
2.	LENGUAJES DE CUARTA GENERACION	2
2.1	DEFINICION	2
2.2	CARACTERISTICAS GENERALES	2
2.3	OBJETIVOS	3
2.4	EVOLUCION DE LOS LENGUAJES DE CUARTA GENERACION	3
2.5	CARACTERISTICAS	3
2.6	USO DE LOS LENGUAJES DE CUARTA GENERACION	4
2.7	VENTAJAS DE LOS LENGUAJES DE CUARTA GENERACION	4
3.	ANALISIS DE ALGUNOS DE LOS LENGUAJES CONSIDERADOS COMO DE CUARTA GENERACION	6
3.1	RALLY	6
3.1.1.	DEFINICION	6
3.1.2.	CARACTERISTICAS GENERALES	6
3.1.3.	CARACTERISTICAS ESPECIFICAS	6
3.1.3.1.	ARCHIVOS DE APLICACION RALLY	7
3.1.3.2.	FUENTES DE DATOS PARA LAS APLICACIONES RALLY	8
3.1.3.3.	SISTEMA DE EJECUCION "RUNTIME" DE RALLY	8
3.1.4.	COMO CREAR UNA APLICACION EN RALLY	9
3.1.5.	BASES DE DATOS Rdb/VMS	12
3.1.6.	CONEXIONES A FUENTES DE DATOS	12
3.1.7.	FORMAS Y REPORTEES	13

3.1.8.	MENUS	14
3.1.8.1.	PARA DETERMINAR EL TRABAJO QUE PUEDE HACER SOBRE UNA FORMA O REPORTE EXISTEN CUATRO MODOS	15
3.1.8.2.	LAS FUNCIONES AGREGADAS QUE USTED PUEDE DEFINIR SON:	16
3.1.8.3.	CAMPOS COMPUTADOS	16
3.1.8.4.	VALIDACION	17
3.1.8.5.	MENSAJES DE AYUDA Y ERROR	17
3.1.9.	RALLY Y OTROS PRODUCTOS VAX	18
3.1.10.	VENTANAS Y TAREAS	18
3.1.11.	CARACTERISTICAS DE SEGURIDAD	19
3.1.12.	COMANDOS DE RALLY	19
3.1.13.	MACROS	20
3.1.13.1.	PASOS QUE SE SIGUEN PARA CREAR UNA MACRO	21
3.1.14.	PASOS QUE SE SIGUEN EN LA CREACION DE UNA APLICACION USANDO COMANDOS	21
3.1.15.	CREACION DEL MENU PRINCIPAL Y PRUEBA DE LA APLICACION	27
3.2.	MAPPER	28
3.2.1.	DEFINICION	28
3.2.2.	CARACTERISTICAS GENERALES	28
3.2.3.	CARACTERISTICAS ESPECIFICAS	29
3.2.4.	BASE DE DATOS MAPPER 1100	31
3.2.5.	FUNCIONES QUE DEBE REALIZAR UNA PERSONA ASIGNADA COMO COORDINADOR DE MAPPER	31
3.2.6.	TECLAS ESPECIALES QUE UTILIZA MAPPER	32
3.2.7.	PASOS PARA VISUALIZAR UN REPORTE	33
3.2.8.	LA LINEA DE CONTROL	34

3.2.9.	FORMATOS DE UN REPORTE	35
3.2.10.	FUNCIONES DE MAPPER	35
3.2.10.1	FUNCIONES DE BUSQUEDA	37
3.2.10.2.	FUNCIONES DE COMPARACION Y DE CLASIFICACION	37
3.2.10.3.	OTRAS FUNCIONES	38
3.2.11.	CORRIDA DE DISENOS	38
3.2.12.	EJEMPLOS DE UN SISTEMA MAPPER	38
3.2.13.	ALGUNAS FACILIDADES DE MAPPER	40
3.3.	LINC	42
3.3.1.	DEFINICION	42
3.3.2.	CARACTERISTICAS GENERALES	42
3.3.2.1.	CONCEPTOS DE LINC	43
3.3.2.2.	TOPICOS DE LA METODOLOGIA	44
3.3.2.3.	CADA COMPONENTE O EVENTO REQUIERE DE TABLAS	44
3.3.3.	CARACTERISTICAS ESPECIFICAS	45
3.3.3.1.	COMPONENTES	45
3.3.3.2.	EVENTOS	46
3.3.3.3.	ESPECIFICACIONES OPCIONALES	46
3.3.3.4.	ESPECIFICACIONES DEL USUARIO	47
3.3.3.5.	ESPECIFICACIONES DE RED	47
3.3.3.6.	ESPECIFICACIONES DE BASE DE DATOS LOGICAS	47
3.3.3.7.	ESPECIFICACION GLOBAL	48
3.3.3.8.	ESPECIFICACIONES DE INTERFACE	48
3.3.3.9.	ESPECIFICACIONES DE DATOS	48
3.3.3.10.	ESPECIFICACIONES LOGICAS	49

3.3.3.11.	COMANDOS QUE SE UTILIZAN PARA CREAR LAS ESPECIFICACIONES	49
3.3.3.11.1.	COMANDOS DE LAS ESPECIFICACIONES OPCIONALES	49
3.3.3.11.2.	COMANDOS DE LAS ESPECIFICACIONES DEL USUARIO	50
3.3.3.11.3.	ESPECIFICACIONES DE RED	52
3.3.3.12.	SINTAXIS DE LOS DIAGRAMAS	53
3.3.3.13.	EJEMPLO DE UN DIAGRAMA RAILROAD	54
3.3.3.14.	SEMANTICA	55
3.3.3.15.	COMO SE DEFINE UNA ENTRADA DE DATOS EN LINC	55
3.3.3.16.	IDENTIFICACION Y CLASIFICACION DEL DOMINIO	56
3.3.3.16.1.	CLASIFICACION HORIZONTAL DE DOMINIO	57
3.3.3.16.2.	CUALIFICACION DE DATOS	58
3.3.3.16.3.	REFERENCIAS DE DOMINIO	58
3.3.3.16.4.	CLASIFICACION DEL DOMINIO VERTICAL	59
3.3.3.16.5.	LINC ES AHORA UN LENGUAJE DE PROCEDIMIENTO	60
3.3.3.17.	QUE ES LIRC?	61
3.3.3.17.1.	REPORTES	61
3.3.3.17.2.	OMISIONES FISICAS	61
3.3.3.17.3.	ARCHIVOS EXTRACT Y ARCHIVOS SORT	62
3.3.3.17.4.	COMUNICACION ENTRE DATOS	62
3.3.3.17.5.	INTERFACES DEL SISTEMA NON-LINC	62
3.3.3.18.	COMO SE DEFINE UN REPORTE EN LIRC?	62
3.3.3.18.1.	SALIDA DEL PANEL (FRAME)	63
3.3.3.18.2.	LOGICA PRINCIPAL	63
3.3.3.18.3.	PANEL LOGICO	65

3.4.	POWERHOUSE	66
3.4.1.	DEFINICION	66
3.4.2.	CARACTERISTICAS GENERALES	66
3.4.3.	CARACTERISTICAS ESPECIFICAS	66
3.4.3.1.	DICTIONARY PLUS	67
3.4.3.2.	QUICK	67
3.4.3.2.1.	COMANDOS DEL QUICK	68
3.4.3.3.	QUIZ	70
3.4.3.3.1.	COMANDOS DEL QUIZ	71
3.4.3.4.	QTP	71
3.4.3.4.1.	COMANDOS DEL QTP	71
3.4.3.5.	THE EXPERT	72
3.4.3.6.	GRAPHICS	72
3.4.4.	COMO MANEJAR POWERHOUSE	73
3.4.5.	AYUDA EN LINEA	73
3.4.6.	RUTINAS EN OTROS LENGUAJES	74
3.4.7.	MANEJO DE DIFERENTES TIPOS DE ARCHIVO	74
3.4.8.	USUARIOS A LOS CUALES ESTA ENFOCADO EL POWERHOUSE	75
3.4.9.	UTILERIAS DE APOYO	75
3.4.10.	OTRAS VERSIONES DE POWERHOUSE	76
3.4.10.1.	TIPO DE USUARIO	76
3.4.10.2.	TIPO DE DESARROLLADOR	76
3.4.10.3.	CODIGO DE GENERADO/ INTERPRETE	77
3.4.10.4.	GENERADOR DE APLICACIONES	77
3.4.10.5.	APLICACIONES TIPICAS	77
3.4.10.6.	INTERFACE CON QBM	77
3.4.10.7.	MEDIO AMBIENTE BASE DE DATOS	78

3.4.10.8.	TAMANO DE LA BASE DE DATOS QUE SOPORTA	78
3.4.10.9.	VOLUMEN DE TRANSACCIONES	78
3.4.10.10.	NUMERO DE USUARIOS	79
3.4.10.11.	MEDIO AMBIENTE FISICO	79
3.4.10.12.	BASE INSTALADA ACTUAL	79
3.5.	OMNIS 3	80
3.5.1.	DEFINICION	80
3.5.2.	CARACTERISTICAS GENERALES DEL OMNIS 3	80
3.5.3.	ESPECIFICACIONES TECNICAS	80
3.5.3.1.	ARCHIVOS DE BIBLIOTECA	80
3.5.3.2.	ARCHIVOS	81
3.5.3.3.	PANTALLAS DE ENTRADA	81
3.5.3.4.	REPORTES	81
3.5.3.5.	MENUS	82
3.5.3.6.	FORMATOS DE BUSQUEDA	82
3.5.3.7.	ARCHIVOS DE DATOS	82
3.5.4.	BASES DE DATOS	82
3.5.5.	EL MENU OPTION	84
3.5.6.	PASOS PARA CREAR UNA APLICACION EN OMNIS 3	84
3.5.6.1.	DISENOS DE ARCHIVOS	85
3.5.6.2.	DISENO DE PANTALLA	88
3.5.6.3.	DEFINICION DE CAMPOS	90
3.5.6.4.	DEFINICION DE COMANDOS	91
3.5.6.5.	CAPTURA DE INFORMACION	93
3.5.6.6.	SEGURIDAD	94
3.5.6.7.	REPORTES	95

3.5.6.8.	MENUS	96
3.5.7.	CONFIGURACION MINIMA REQUERIDA	96
4.	METODOLOGIAS PARA MEDIR LA PRODUCTIVIDAD DE SOFTWARE	97
4.1.	CARACTERISTICAS PRINCIPALES	97
4.2.	METODOLOGIAS	97
4.2.1.	LINEAS DE CODIGO	98
4.2.2.	METODO DE HALSTEAD	98
4.2.3.	PUNTOS DE FUNCION	100
4.2.3.1.	CRITERIOS PARA EVALUAR LAS ENTRADAS EXTERNAS	100
4.2.3.2.	CRITERIOS PARA EVALUAR LAS SALIDAS EXTERNAS DE USUARIO	102
4.2.3.3.	CRITERIOS PARA EL CONTEO REFERENTE A LOS ARCHIVOS DEL USUARIO	103
4.2.3.4.	CRITERIOS PARA EL CONTEO DE INTERFASES CON OTRAS APLICACIONES	105
4.2.3.5.	CRITERIOS PARA EL CONTEO DE CONSULTAS	105
4.2.3.6.	FACTORES DE AJUSTE DE VALOR	106
4.2.3.7.	FORMATO PARA EL CALCULO DEL INDICE DE PUNTOS DE FUNCION	109
4.2.3.8.	GUIA PARA SELECCIONAR LOS FACTORES DE COMPLEJIDAD PARA LA ENTRADA	111
4.2.3.9.	GUIA PARA SELECCIONAR LOS FACTORES DE COMPLEJIDAD PARA LA SALIDA	112
4.2.3.10.	GUIAS PARA SELECCIONAR LOS GRADOS DE COMPLEJIDAD EN LOS ARCHIVOS LOGICOS DEL USUARIO	113
4.2.3.11.	GUIA PARA SELECCION DE GRADOS DE COMPLEJIDAD DE LAS INTERFACES CON OTRAS APLICACIONES	114

4.2.3.12. GUIA PARA SELECCIONAR LOS FACTORES
DE COMPLEJIDAD PARA LA CONSULTA 114

5. PROVEEDORES DE DBMS Y LENGUAJES
DE CUARTA GENERACION 116

CONCLUSIONES

ANEXO

BIBLIOGRAFIA

1. INTRODUCCION

El uso de la computadora en todos los campos se ha generalizado a tal grado en los últimos tiempos, que su evolución ha traído como consecuencia el surgimiento de nuevos lenguajes de programación; los que se conocen como **LENGUAJES DE CUARTA GENERACION** y cuyas características pretenden suplir la falta de programadores especializados en todas las áreas; así como aumentar la productividad de las empresas que los usen.

Este escrito pretende presentar las características generales y específicas de algunos de ellos con la finalidad de familiarizar al lector con sus fundamentos internos y su funcionamiento para que los consideren como una solución a sus problemas de falta de personal especializado, grandes centros de cómputo y tiempo de desarrollo entre otros.

2. LENGUAJES DE CUARTA GENERACION

2.1 DEFINICION

Un lenguaje de cuarta generaci3n es una herramienta de un sistema de computo dise1ada a un alto nivel.

- . Pocas instrucciones
- . Mayor trabajo de equipo.
- . Menor trabajo de programadores.

2.2 CARACTERISTICAS GENERALES.

- Los lenguajes de cuarta generaci3n aparecieron a fines de la d3cada de los setentas.
- Se aplican a problemas especificos y emplean sistemas de bases de datos o archivos no estandar.
- Deben utilizar el software estandar del sistema que se emplea en su medio ambiente, especialmente para servicios de comunicaciones de datos o bien deben reemplazar completamente los sistemas previos.
- Requieren m1s capacidad de computo que los sistemas de tercera generaci3n (cobol, PL / I).

Los lenguajes de cuarta generaci3n se dividen en dos categorias.

La primera categoria son los que fueron dise1ados y usados principalmente para profesionales de Inform1tica, para el desarrollo de aplicaciones para la producci3n.

Estas herramientas tienden a ser menos "amigables al usuario" y estan orientadas a la ejecuci3n, a la integridad de los datos, as1 como a la seguridad y actualizaci3n de estos a altas velocidades.

La segunda categoria son los lenguajes de cuarta generaci3n orientados al uso directo de usuarios finales, llenando sus necesidades diarias de informaci3n. Entre sus caracteristicas

se encuentra el énfasis que ponen en la facilidad de uso, los reportes a la medida, la actualización de datos a baja frecuencia y las aplicaciones "THROW-AWAYS".

2.3 OBJETIVOS.

El objetivo principal de los lenguajes de cuarta generación es hacer el proceso de desarrollo de aplicaciones y el de mantenimiento más fácil, ya que el 75% del presupuesto de las compañías, se utiliza en esta área y sólo se dedica el 25% a la productividad.

Otro objetivo es producir la cantidad de experiencias requerida por el programador. Así como reducir el tiempo y el esfuerzo del ciclo de mantenimiento.

2.4 EVOLUCION DE LOS LENGUAJES DE CUARTA GENERACION.

En la primera generación de los lenguajes de este tipo se empleó código en lenguaje maquina.

En la segunda generación, lenguaje ensamblador.

En la tercera, lenguajes de alto nivel como: cobol, RPG, FORTRAN, etc. Complementados con: administradores de bases de datos y comunicaciones.

2.5 CARACTERISTICAS

Idealmente, los lenguajes de 4a generación deben ser evaluados bajo las siguientes características:

- Que no sea un lenguaje de procedimiento.
- Que especifique el que, no el como.
- Que haga selección automática de los datos.
- Que almacene comandos de alto nivel.
- Que pueda ser usado por profesionales del centro de cómputo por usuarios de sistemas o por usuarios casuales.

- Que pueda crecer en capacidad y en uso.
- Que cuente con poderosas facilidades de SOFTWARE para:
 - . LA SALIDA GRAFICA
 - . EL DISEÑO AUTOMÁTICO DE PANTALLAS
 - . LAS FACILIDADES DE AYUDA
 - . QUE SEA FÁCIL DE APRENDER

Estadísticas tales como las funciones trigonométricas, financieras, etc.

- Que sirva como interfase entre los archivos y las múltiples bases de datos.
- Que cuente con visión lógica de los datos para los usuarios utilizando bases de datos relacionales o accesos parecidos.
- Que cuente con diccionario de datos.
- Que la estructura de los datos provea seguridad e integridad de los datos.
- Que la migración hacia otros equipos de las aplicaciones existentes sea transparente.-

2.6 USOS DE LOS LENGUAJES DE CUARTA GENERACION.

El usuario final UF puede hacer sus propios programas, acceder sus datos, hacer gráficas interactivas, hacer series en línea y desarrollar sistemas de información gerencial complejos.

2.7. VENTAJAS DE LOS LENGUAJES DE CUARTA GENERACION.

No se requieren compilaciones de programas (en algunos casos).

Aproximadamente utilizan la décima parte del código usado (necesario) en lenguajes de tercera generación.

Se reduce el trabajo de la prueba de programas.

Cualquier programa de negocios en lenguajes de tercera generación se puede escribir en lenguajes de cuarta generación.

Los más integrados sólo se pueden escribir en lenguajes de cuarta generación.

El 75% de la programación la pueden hacer UF.

Para los altos y medianos ejecutivos, los lenguajes de cuarta generación generan la posibilidad de realizar queries en los bancos de datos de la empresa y acceder datos en segundos.

Los lenguajes de cuarta generación son benéficos porque: Mejoran la productividad en un rango de 5:1 a 300:1 en diferentes actividades.

La mayoría del STAFF de la organización podría utilizar o tratar con computadoras pues los lenguajes están diseñados para ambos: Especialistas en computación y usuarios finales.

- Son herramienta ideal para los centros de información.
- Son ideales como generadores de aplicación.
- Tienen facilidad de uso.
- Propician la expansión de nuevas aplicaciones.
- Verificación de sintaxis y lógica del sistema.
- Son autodocumentables.

3. ANALISIS DE ALGUNOS DE LOS LENGUAJES CONSIDERADOS COMO DE CUARTA GENERACION.

3.1. RALLY

3.1.1. DEFINICION

VAX RALLY es un lenguaje de cuarta generaci3n que se puede usar para crear aplicaciones. Los usuarios de dichas aplicaciones pueden reunir datos fuente de sistemas locales o de sistemas remotos, trabajar con ellos y desplegar luego resultados, todo dentro de RALLY.

Es un lenguaje que corre en todos los equipos VAX de la compa1a DIGITAL, excluyendo Micros. Su sistema operativo es el VMS y el Micro-VMS.

3.1.2 CARACTERISTICAS GENERALES

Rally consta de dos sistemas principales:

- Un sistema para definir y modificar aplicaciones conocido como DIALOGO y
- Un sistema para ejecutar aplicaciones al tiempo que se deseen correr o de que se pide sean corridas, conocido como "RUNTIME".

3.1.3 CARACTERISTICAS ESPECIFICAS.

QUE SE NECESITA PARA CREAR UNA APLICACION EN RALLY?

- Un Archivo de definici3n de la aplicaci3n.
- Una o m1s fuentes de datos almacenados.
- El sistema de ejecuci3n de RALLY.

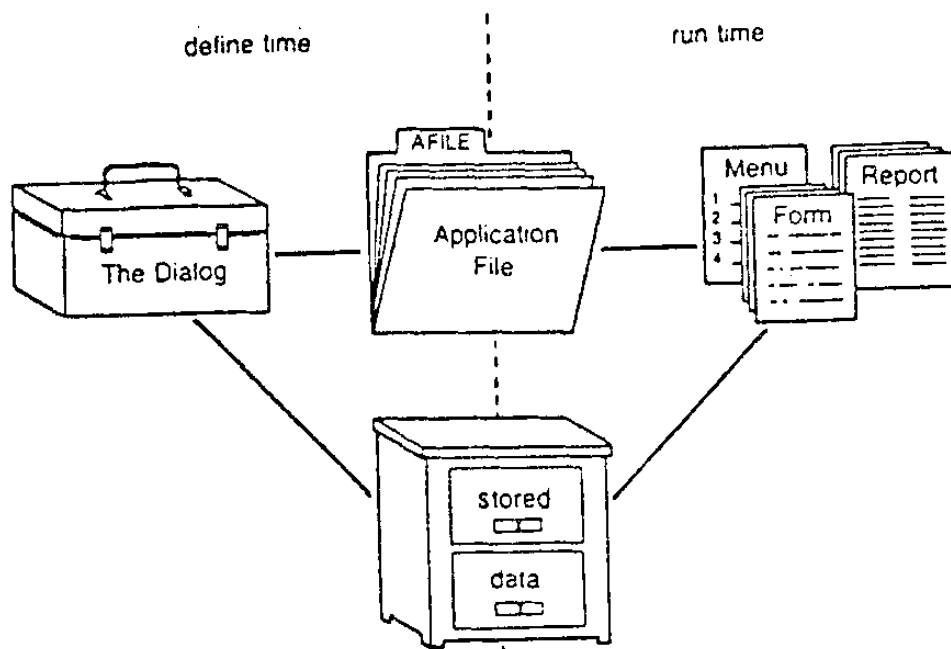
3.1.3.1 ARCHIVOS DE APLICACION RALLY.

Cuando se crea una aplicación con RALLY se define cada objeto de la aplicación. Los principales objetos incluyen:

- MENUS.
- FORMAS Y REPORTES CONECTADOS A LAS FUENTES DE DATOS. ALMACENADOS.
- MENSAJES DE AYUDA Y MENSAJES DE ERROR.

El DIALOGO, sistema de definición de RALLY, genera una serie de estructuras de datos que definen los objetos de la aplicación.

Estas definiciones se guardan en un archivo denominado "AFILE", archivo de la aplicación.



Cuando los usuarios finales trabajan con las aplicaciones creadas, el sistema de ejecución de RALLY interpreta las estructuras de datos en el Archivo de Aplicación.

Mucho del trabajo realizado por los intérpretes tradicionales, en el sistema de ejecución de RALLY lo hace en el tiempo de la definición de la aplicación; de tal manera que éstas son interpretadas rápidamente.

No hay necesidad de que los archivos sean compilados o

enlazados como sucedería con las aplicaciones desarrolladas en COBOL o en algún otro lenguaje procedual.

Esto significa que cuando se modifican las definiciones de los objetos de una aplicación en el archivo de aplicaciones, los cambios se realizan rápidamente y pueden verse inmediatamente los resultados.

3.1.3.2 FUENTES DE DATOS PARA LAS APLICACIONES RALLY

Las aplicaciones de RALLY proporcionan la habilidad de reunir datos de varias fuentes.

Estas pueden ser sistemas locales o remotos así como el sistema donde se pretende desarrollar la aplicación. Inclusive puede reunir datos de fuentes que estén en uso.

- Bases de datos Rdb/VMS.
- Dominios de "DATATRIVE" que usa datos almacenados en:
 - . archivos RMS
 - . bases de datos DBMS

Cuando se definen aplicaciones con RALLY, se puede almacenar, modificar y borrar los datos en bases de datos Rdb/VMS.

Cuando se accesan los datos en bases de datos Rdb/VMS pueden escribirse y leerse mientras que los datos almacenados en dominios "DATATRIVE" sólo se pueden leer.

3.1.3.3 SISTEMA DE EJECUCION "RUNTIME" DE RALLY.

Para ejecutar una aplicación de RALLY, el sistema de ejecución se proporciona con el sistema "TEAMDATA", sistema administrador de la información para usuarios finales; que entre otras posibilidades corren aplicaciones en RALLY.

Existen cuatro modos de correr una aplicación en RALLY.

- Dentro de un sistema de definición de la aplicación RALLY.

- A través de "TEAMDATA".
- De ALL-IN-1, cuando la aplicación ha sido incorporada en el menú ALL-IN-1.
- Desde el nivel DCL usando el comando RALLY run.

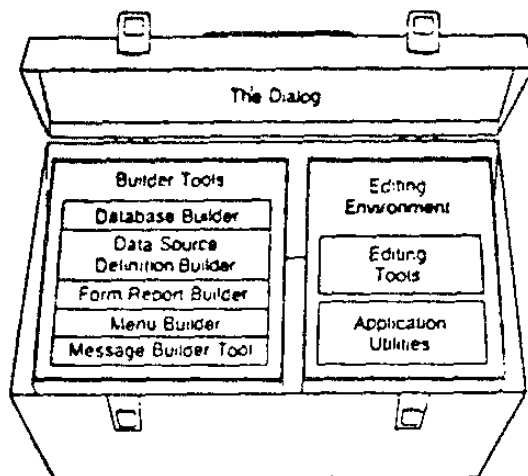
3.1.4. COMO CREAR UNA APLICACION EN RALLY

Para crear aplicaciones en RALLY se usa un sistema de definición EL DIALOGO.

El diálogo consiste de una serie de menus y formas respaldadas por ayuda exhaustiva en línea.

La figura represente el Diálogo, el cual está formado de dos sistemas componentes:

- Las Herramientas del Constructor "THE BUILDER TOOLS".
- El Medio Ambiente de Edición "THE EDITING ENVIRONMENT".



REPRESENTACION SIMPLIFICADA DEL DIALOGO.

Las herramientas del constructor permiten poner aplicaciones juntas rápidamente, estas herramientas poderosas se usan para crear prototipos y aplicaciones simples. Se puede trabajar con estas herramientas para hacer opciones de menús y llenado de campos en las formas.

LAS HERRAMIENTAS DEL CONSTRUCTOR SON:

- . CONSTRUCTOR DE LA BASE DE DATOS.- El cual crea y modifica las bases de datos Rdb/VMS.
- . CONSTRUCTOR DE LA DEFINICION DE LOS DATOS.- El cual crea conexiones de la base de datos a las formas y reportes que trabajan con esa fuente.
- . CONSTRUCTOR DE FORMAS Y REPORTES.- El cual crea la estructura de formas y también crea las salidas a la pantalla.
- . CONSTRUCTOR DEL MENU.- El cual crea mensajes de ayuda para la aplicación.

EL MEDIO AMBIENTE DE EDICION.- Es la parte del DIALOGO que se usa para modificar los objetos creados con las herramientas del constructor.

Este se puede usar también para crear nuevos objetos de aplicaciones con excepción de bases de datos Rdb/VMS, que solamente se pueden editar con el constructor de bases de datos.

También se pueden usar para trabajar con lenguajes de programación como VAX-COBOL o VAX-BASIC.

Además RALLY proporciona un lenguaje de programación de propósito especial que define las características avanzadas de la aplicación. Se llama ADL, lenguaje avanzado de desarrollo.

Se usa para aumentar el poder de las aplicaciones de RALLY a través de sus características.

- Campos en formas o reportes cuyos valores son calculados en el sistema de ejecución.
- Validación avanzada de la entrada de datos.
- Manipulación avanzada de datos.

La mayoría de los procedimientos ADL son cortos; algunos se concretan a una línea.

El proceso de crear una aplicación con RALLY consiste de una

serie de pasos en los que se define cada objeto en la aplicación.

Una característica interesante de RALLY es que el DIALOGO es por sí mismo una aplicación de RALLY.

Ya que una persona que define una aplicación la usa para construir otra. Esto significa que ambos, el que la define así como el usuario final de aplicaciones de RALLY, trabajan en medio ambientes similares y hablan el mismo lenguaje, eliminando así la laguna que frecuentemente separa a los definidores de aplicaciones de los usuarios finales.

Debido a que DIALOGO es una aplicación de RALLY se pueden usar las formas, los reportes y los menus en el DIALOGO para proporcionar ideas para desarrollar sus propias aplicaciones.

PASO 1 DEFINIR LA BASE DE DATOS.

Crear una base de datos relacional con la herramienta del constructor de base de datos.

PASO 2 CREAR LAS DEFINICIONES DE FUENTES DE DATOS.

Definir los objetos RALLY que concretan tu base de datos a las formas y reportes de tu aplicación. Puedes usar una base de datos existente Rdb/VMS o un dominio "datatrieve".

PASO 3 CREAR LAS FORMAS Y REPORTE

Crear formas y reportes con la herramienta para construir formas o reportes. RALLY proporciona salidas por omisión para ambos, formas y reportes, que se pueden modificar posteriormente en el medio ambiente de edición.

PASO 4 CREAR EL MENU PRINCIPAL DE APLICACION.

Crear el menu principal con la herramienta del constructor de menu. En este paso se puede determinar también el modo de uso de las formas y reportes, tales como actualizar, insertar, llegar a ella a través de un menu principal. También se pueden asignar números de ayuda a los mensajes creados por usuario.

PASO 5 CREAR MENSAJES DE AYUDA Y ERROR.

Componer y editar mensajes de ayuda y error con la herramienta constructora de mensajes para el menu, formas y reportes.

PASO 6 CORRER LA APLICACION EN RALLY.

Correr el prototipo y más tarde desarrollar la aplicación. Los usuarios de la aplicación pueden trabajar con ella inmediatamente.

3.1.5. BASES DE DATOS Rdb/VMS.

Algunas aplicaciones trabajan por completo con fuentes de datos existentes, mientras que otras requieren de la creación de nuevas fuentes de datos.

Si se desea crear nuevas fuentes de datos se puede hacer con la herramienta del constructor de bases de datos de RALLY o solamente modificar la existente.

Cuando se crea una aplicación que trabaja con bases de datos existentes, se puede proporcionar a RALLY nombres y lugares de tal manera que la aplicación pueda trabajar con ellos.

3.1.6 CONEXIONES A FUENTES DE DATOS.

Para que las formas y los reportes de una aplicación trabajen con datos, se debe crear una conexión a la fuente de esos datos. El objeto de RALLY que hace esta conexión se llama definición de la fuente.

Una definición de la fuente de datos puede trabajar con:

- Una relación individual en una base de datos Rdb/VMS.
- Un solo dominio "datatrieve"
- Múltiples relaciones en base de datos Rdb/VMS.

La definición en la fuente de datos es el objeto que determina cuales registros son traídos de la fuente de datos. En el caso de las bases de datos Rdb/VMS, los datos pueden ser almacenados así como recuperados.

Una forma o reporte en una aplicación de RALLY puede trabajar con una o más definiciones de fuentes de datos. Dichas definiciones pueden pertenecer a una base de datos o varias, pueden ser locales o remotas o pertenecer a uno o varios dominios "datatrieve".

3.1.7 FORMAS Y REPORTEES

Las formas y los reportes son las herramientas de los usuarios finales para capturar y desplegar datos dentro de las aplicaciones de RALLY y estas pueden ser creadas por el DIALOGO en el estilo más apropiado para la gente que va a usarlos.

Las formas y los reportes pueden ser creados por omisión con la herramienta del constructor de formas y reportes basados en la definición de la fuente de datos, luego puede usarse el medio ambiente de edición para modificarlos o crear nuevos como se deseen. Los reportes o las formas pueden ser:

- INSERTADOS
- MODIFICADOS
- BORRADOS
- DESPLEGADOS
- FORMATEADOS

Una de las características más innovadoras de RALLY es que se siguen los mismos pasos para definir las formas y los reportes, lo que simplifica el proceso de aprendizaje así como el diseño de los dos componentes esenciales de una aplicación.

RALLY no distingue entre formas y reportes así que algunos usuarios pueden usar la misma definición como entrada de datos, mientras que otros la usan como reporte.

Sales Prospect Entry Form

Cust Name: _____

Sales Id: _____

Address Data 1: _____

Address Data 2: _____

Town: _____

State/Province: _____

Postal Code: _____

Phone: _____

Use this form to enter information about sales prospects or to query the Sales Prospect Database. If you need help, press HELP or GOLD-H.

APLICACION DE FORMA/REPORTE EN RALLY

3.1.8 MENUS

Los menus proporcionan el significado básico de acceso a los objetos en una aplicacion RALLY.

Los menus conducen a formas, reportes, a otros menus u otros objetos.

Se construyen con la herramienta del constructor de menus en el DIALOGO.

Se pueden construir en forma horizontal o de banda o en un sinómuero de estilos que RALLY proporciona.

3.1.8.1 PARA DETERMINAR EL TRABAJO QUE PUEDE HACERSE SOBRE UNA FORMA O REPORTE EXISTEN 4 MODOS.

- EL MODO BUD.- En el cual se pueden ver todos los registros, cambiar los valores de los campos en los registros individuales o borrar registros o insertar datos.
- EL MODO QUERY.- Este modo permite a los usuarios traer datos de un almacen, basados en un valor o combinaci3n de valores que ellos suministran en uno o m3s campos de un registro.
- EL MODO INSERT.- Este modo permite a los usuario almacenar nuevos registros.
- EL MODO PRINT.- Permite a los usuarios enviar la salida de un reporte o forma directamente a la impresora o a un archivo.

LOS MENUS CONSTRUIDOS EN EL MEDIO AMBIENTE DE EDICION PUEDEN SER LISTADOS:

- VERTICALMENTE
- HORIZONTALMENTE
- EN COLUMNAS
- EN CUALQUIER PATRON QUE SU TERMINAL PERMITA

PUEDEN SER SELECCIONADOS:

- CON UNA PALABRA
- CON UNA ABREVIACION UNICA
- CON UNA LETRA
- CON UN NUMERO

EL MOVIMIENTO DEL CURSOR PUEDE SER CON LAS FLECHAS.

- EXTENDIENDO LA ILUMINACION AL DESPLAZARSE.
- CREANDO UNA FLECHA HORIZONTAL QUE INDIQUE HACIA LA SELECCION.

El medio ambiente de edición incluye un editor de imagen que modifica la apariencia de los menus. Se puede agregar texto, así como características gráficas tales como límites o video inverso.

Una característica de una aplicación avanzada de RALLY es definir una lista de valores de donde los usuarios deben seleccionar o hacer referencia a ellos para cambiar el contenido del campo por medio de un comando de RALLY o hacer que RALLY copie esos valores en otros campos.

Los agregados son campos que proporciona RALLY para la información resumida. Los agregados se definen en el medio ambiente de la edición.

3.1.8.2. LAS FUNCIONES AGREGADAS QUE USTED PUEDE DEFINIR SON:

- SUMA
- MAXIMO
- CUENTA
- PROMEDIO
- PROMEDIO-ALTO
- PROMEDIO-BAJO

Se pueden usar agregados para crear totales para reportes completos, para páginas individuales dentro de los reportes o para cada registro.

3.1.8.3 CAMPOS CALCULADOS.

Rally proporciona la habilidad para crear campos computados

basados en computaciones que usted defina.

Puede definir los campos computados con la herramienta del constructor de bases de datos o en el medio ambiente de edición para las formas o reportes, o en el ADL cuyo procedimiento puede ser una sola línea. Ejemp:

FORM1.CUSTOMER-COST:=FORM1.BASE-COST*FORM1.MAKEUP:

3.1.8.4. VALIDACION

Una de las más importantes consideraciones en cualquier aplicación de procesamiento es la entrada de datos validados en la base de datos.

Los criterios de validación en el medio ambiente de edición son:

- Validación específica.
 - . Desde una lista de valores.
- Definir un campo que debe ser llenado con una entrada desde la lista de valores.
 - . Desde un rango de valores.
- Máximo o mínimo.
 - . Con procedimientos ADL.
- Especificar validaciones más complejas.

3.1.8.5. MENSAJES DE AYUDA Y ERROR.

RALLY proporciona ayuda en línea a los usuarios de las aplicaciones.

Se pueden crear mensajes de ayuda para describir un menú completo, una selección de un menú, una forma completa o un campo de una forma.

Tambièn se pueden crear mensajes de error para su aplicaciòn. Crear texto con la herramienta del constructor de mensajes y usar un procedimiento del ADL para conectarlo con el objeto apropiado.

3.1.9. RALLY Y OTROS PRODUCTOS VAX.

Las aplicaciones de RALLY trabajan con muchos otros productos de Software VAX incluyendo y dependiendo de la definiciòn de la aplicaciòn.

- DCL
- TEAMDATA.
- ALL-IN-1.
- Rdb/VMS DATABASES.
- DATATRIEVES DOMAINS.
- RMS FILES.
- DBMS DATABASES.

3.1.10 VENTANAS Y TAREAS

Siempre que los usuarios estàn trabajando con RALLY pueden trabajar con ventanas.

Las ventanas pueden ser tan grandes como la pantalla o tan pequeñas como un solo caracter.

Trabajar en RALLY implica trabajar dentro de una tarea. Una aplicaciòn de Rally consta de una o mäs tareas, cada una desplegada en una ventana.

La tarea consiste en una serie de acciones de Rally conectadas.

Una acción de Rally es la invocaciòn de un objeto tal como un menu, una forma o un reporte o un mensaje.

La tarea es un grupo de acciones relacionadas que aparecen

en una ventana en la aplicación del usuario. Por Ejemplo:

Si un usuario está trabajando con una entrada de una aplicación de una orden de ventas y empieza desde un menú. Selecciona una forma, ésta aparece en la pantalla y mete datos. En este punto, dos acciones han sido visibles para el usuario.

- INVOCAR EL MENU.
- INVOCAR LA FORMA.

Ambas acciones están corriendo en la misma tarea, visible en la misma ventana.

RALLY proporciona la habilidad de poner una o más tareas en la ventana del usuario en la pantalla. Los usuarios pueden moverse fácilmente de una tarea a otra con un solo comando.

El cursor siempre aparece en la celda activa.

La ventana que contiene la celda activa es la del frente.

Las ventanas que contienen tareas temporalmente suspendidas pueden aparecer parcial o completamente obscurecidas.

3.1.11 CARACTERISTICAS DE SEGURIDAD

RALLY permite definir múltiples puntos de entrada para los usuarios de las aplicaciones. Según convenga a la compañía en la cual se usa. Por Ejemplo: El acceso de un usuario puede ser una aplicación completa mientras que para otro sólo es a una parte de ésta o solamente a la entrada de datos.

Esto se hace por medio de "passwords".

Cada entrada es una tarea. El uso de múltiples puntos de entrada no es precisamente una característica de seguridad sino la definición del acceso a una área determinada.

3.1.12 COMANDOS DE RALLY

RALLY contiene comandos para permitir a los usuarios hacer

cosas tales como:

- Navegar dentro y entre los menus y las formas o los reportes.
- Manejar datos dentro de las formas o de los reportes.
- Manipular las ventanas.

Imprimir porciones de una aplicación.

La mayoría de los comandos se asignan a teclas de tal manera que los usuarios pueden invocarlos al presionarlas. Por Ejemplo: Los usuarios pueden invocar el comando "next field" para mover el cursor al próximo campo al presionar la tecla RETURN.

Sin embargo, la mayoría de los comandos pueden ser invocados en la línea de comandos la cual aparece en la parte inferior de la pantalla y se invoca con el comando GOLD o GOLD-KP7 en cualquier terminal o con la tecla DO de las terminales VT200.

3.1.13 MACROS

Trabajar en muchas aplicaciones, implican secuencias que se repiten una y otra vez.

RALLY proporciona un modo de asignar secuencias repetitivas a una tecla o a una secuencia de teclas.

Una macro es una manera de capturar los comandos que son parte de una secuencia en particular. Cuando una aplicación del usuario ejecuta una macro, el resultado es el mismo que si invocara todos los comandos a la vez.

Las secuencias de comandos pueden ser escritas en un archivo principal y cargadas a una aplicación por el sistema de ejecución de RALLY.

3.1.13.1 PASOS QUE SE SIGUEN PARA CREAR UN MARCO.

- PASO 1 Presione GOLD-< RALLY, cuando aparezca el prompt teclee un número del 0 al 29 inclusive.
- PASO 2 Teclée el número 1 en respuesta al prompt numérico Macro y presione RETURN.
- PASO 3 Ahora ejecute la secuencia de pasos que quiere que la macro grabe. Presione la flecha hacia abajo, luego la flecha hacia la derecha para mover el cursor al menu "REPORT-PEOPLE", presione RETURN. El reporte aparecerá en la pantalla.
- Presione GOLD-COMMA para invocar el comando "SET QUERY". Los registros desaparecerán del reporte, saliendo solamente los campos en blanco bajo las etiquetas. El reporte está listo ahora para que lo uses como forma y escribas la base de datos.
- PASO 4 Presiona "GOLD-<" para notificar a RALLY que se ha terminado de meter la secuencia para la macro. Ahora presione "ENTER".

Ahora ya puede usar la macro e ir directamente del menu al reporte.

Para correrla invoque la línea de comando con la tecla "DO" o "GOLD-[" y teclée.

"USE MACRO 1".

Presione return y el reporte aparecerá en su pantalla listo para que teclée su query.

3.1.14 PASOS QUE SE SIGUEN EN LA CREACION DE UNA APLICACION USANDO COMANDOS.

Paso 1 Crear un "AFILE", el cual se puede obtener tecleando lo siguiente:

```
$ RALLY
```

- COMMAND: CREATE
- AFILE: FIRST-AFILE

Aunque se puede teclear todo en el mismo renglón.

* RALLY CREATE FIRST-AFILE.

Paso 2 Crear una base de datos.

Del menú principal de RALLY seleccione el número:

2) construcción de bases de datos.

Del menú de definición de bases de datos seleccione:

1) Create a data base.

El sistema regresa un mensaje de "NEW DATABASE" y una pantalla donde pide el nombre de la nueva base de datos y si desea usar parámetros por omisión.

La leyenda al fondo de esta pantalla le dice que hacer.

Create a Database

Name of database: _____

Use default parameters? L

Enter the name of the database you want to create, followed by <RETURN>.

To return to the previous menu without creating a database, use the 'quit action' command.

Use el comando "FINISH ACTION" para dejar esta forma. Para invocarla de nuevo presione "ENTER" o "GOLD-F" o "F10" en las "VT200".

PASO 3 Hacer las definiciones globales de los campos.

Desde el "EDIT A DATABASE MENU", seleccione:

1) "WORK ON GLOBAL FIELD DEFINITIONS"

Desde el "DEFINING GLOBAL FIELD DEFINITIONS MENU" seleccione:

1) " CREATE A GLOBAL FIELD DEFINITION"

En el "CREATE A GLOBAL FIELD DEFINITION FORM teclee:

"NAME_TYPE"

Esta forma contiene solo un campo. Use el comando "FINISH ACTION" para moverse a la siguiente forma. En este momento, la forma contiene un conjunto nuevo de campos, para que usted los llene, determinando así su tipo y su tamaño.

Create a Global Field Definition	
Data type: <u>TEXT</u>	Data types
Maximum size: <u>31</u>	WORD
Missing Value: _____	L WORD
Comment:	Q WORD
	F FLOATING
	G FLOATING
	TEXT
	VARYING STRING
	DATE
Use the 'list_of_values' and 'select value' commands to select a data type. Or, type the name of a data type, followed by <RETURN>.	
Name of database: <u>NEW DATABASE</u>	
Name of field being created: <u>TYPE</u>	

En "DATA TYPE" se acepta por omisión TEXT y luego RETURN.

En "MAXIMUN SIZE" se acepta por omisión 31. Si desea otro tamaño de campo teclee arriba y

después presione RETURN.

En "MISSING FIELD" se tecléa el tamaño máximo que debe aceptar la definición global de los campos en caso de que no se teclée ninguna y luego RETURN.

En el campo de comentario, teclée algo referente al campo que usted ha creado. Termina la definición del campo tecleando el comando "FINISH ACTION".

PARA CREAR OTRA FORMA DE DEFINICION GLOBAL DEBE TECLEARSE:

ID-TYPE

Luego el comando "FINISH ACTION"

Si en el campo de "DATA TYPE" desea teclar el número de identificación de una persona, vea la lista de valores del lado derecho de la pantalla.

Si el número que va a almacenar es mayor de 6 dígitos teclée

L WORD

Donde puede almacenar hasta más o menos 2 billones de números.

La lista de valores aparece oprimiendo la tecla "GOLD-KP" o "GOLD-L" para moverse a la lista y con las flechas hacia abajo hasta L WORD. Use el comando "SELECT VALUE" para teclar:

L WORD

Para salir de esta forma use "FINISH ACTION"

Para crear el último campo global, para el número de oficina de la persona, siga el mismo procedimiento que la creación de las dos definiciones anteriores.

Llame a este campo global final:

OFFICE.

Use "TEXT" como el tipo de datos y la longitud máxima de 4. Ahora ya están terminadas las definiciones globales de los campos.

Salga con el comando "FINISH ACTION"

Del menú "DEFINING GLOBAL FIELD DEFINITIONS, seleccione:

4) " MOVE TO PREVIOUS MENU"

Paso 4 Definir una relación en la cual se basan estos campos.

Del "EDIT A DATABASE MENU" seleccione:

2) WORK ON RELATION DEFINITION.

Del "DEFINING DATABASE RELATIONS MENU" seleccione:

1) CREATE A RELATION

En la forma "DEFINING A RELATION" teclée:

PEOPLE

En esta relación se llamará PEOPLE porque estará compuesta de registros que contengan información acerca de la gente en una organización. Use el comando "FINISH ACTION" para salir de esta forma.

Desde el "EDIT RELATION MENU" SELECCIONE:

1) CREATE A FIELD

En el primer campo de la forma "CREATE A FIELD" teclée un nombre de campo para la relación PEOPLE. Teclée

UID

En el segundo campo escriba un comentario o déjelo en blanco.

En el tercer campo usted puede crear un campo computado. El campo contiene el valor por omisión N para NO acepte esto y presione RETURN.

En el cuarto campo "GLOBAL FIELD BASE ON" teclée:

ID-TYPE

Use el "FINISH ACTION" para dejar esta forma.

Use el "EDIT A RELATION MENU" y seleccione:

1) CREATE A FIELD

En el primer campo, teclée:

NAME

Teclée un comentario si desea. En el siguiente campo, acepte la N y en el campo final, RALLY pide el nombre del campo local. Desde aquí el procedimiento es el mismo hasta terminar con el comando "FINISH ACTION"

Usted se puede ir directamente al menú principal del diálogo "EDIT A RELATION" al teclear:

TOP

De esta manera termina la definición de la base de datos.

Paso 5 Construir una forma de reporte.
Puede:

Crear una definición fuente de datos.

Crear una forma/reporte implícita.

Grupo fuente de datos.

Forma o reporte de salida.

Se construyen al trabajar por medio de menús y formas en el diálogo, del mismo modo que se construyó la base de datos.

3.1.15 CREACION DEL MENU PRINCIPAL Y PRUEBA DE LA APLICACION.

Cuando se tiene todo esto creado, se construye el menú principal de la misma forma y se procede a probar la aplicación.

- Del menú "DEFINIG APPLICATIONS" seleccione:

4) "MOVE TO PREVIOUS MENU "

- Del "MAIN MENU RALLY APPLICATION DEFINER" seleccione:

6) "EXECUTE APPLICATION"

- En la forma "EXECUTE APPLICATION AFILE FORM", sólo tiene que llenar un campo. Este campo pide el punto de entrada de su aplicación. Use la "LISTA DE VALORES" y "SELECT VALUE" para entrar:

MAIN_TASK

- Use el comando "FINISH ACTION" para salir y entrar a su aplicación.

El menú principal de su aplicación aparecerá ahora en su pantalla.

3.2. MAPPER

3.2.1 DEFINICION

Es un lenguaje de cuarta generaci3n orientado al usuario del nivel estrat3gico-t3ctico, es una herramienta interactiva que puede ser usada en computadoras peque1as asi como en m3quinas grandes de la compa1a SPERRY, desde el sistema SPERRY MAPPER 5 hasta MAPPER 10, su nombre se deriva de la letra inicial de los siguientes t3rminos:

- MANAGEMENT
- APPLICATIONS
- PROGRAMMING
- PLANNING
- EXECUTIVE
- REPORTS

Que en espa1ol significan reportes ejecutivos para la planeaci3n y programaci3n de aplicaciones a nivel administrativo.

3.2.2 CARACTERISTICAS GENERALES

- Fue dise1ado por SPERRY para uso exclusivo de sus equipos.
- Trabaja bajo el sistema interprete.
- Traduce las instrucciones a lenguaje ensamblador.
- ES UN PAQUETE QUE CUENTA CON:
 - . Una base de datos ya definida.
 - . Lenguaje manejador de base de datos.
 - . Dise1o de programas en base a las funciones del

lenguaje manejador.

3.2.3 CARACTERISTICAS ESPECIFICAS

Los sistemas MAPPER 5 ofrecen un sinnúmero de características activas para una pequeña oficina, de la misma manera que para una compañía grande que procesa sus datos en forma distribuida y que tiene áreas externas al edificio central.

- Cuenta con numerosas funciones., Una gran variedad de comandos y funciones están disponibles para satisfacer a los usuarios inexpertos así como a los más experimentados.
- Desarrollo de aplicaciones. Las aplicaciones pueden ser desarrolladas por los usuarios.
- Documentación en línea. Con el sistema MAPPER 5 hay poca necesidad de documentación.

El sistema proporciona instrucciones y material de información en línea, de manera que pueda ser accesada desde cualquier terminal.

- Instrucción de ayuda de como usar el sistema en forma computarizada.

No solamente puede darle ayuda e información en línea desde el sistema sino que puede proporcionarle instrucción de como usar el sistema. También puede probar al usuario y corregir las respuestas equivocadas.

- Funciones administrativas simplificadas. La tarea de poder visualizar quien usa el sistema y que información están accesando, simplifica el trabajo.
- El comando HELP.

Si usted olvida una función mientras trabaja, puede pedir ayuda, después de obtener la respuesta del sistema que necesitaba, puede continuar donde se quedó.

- Funciones para definir las bases de datos del usuario final.

Cuando se necesita un reporte mensualmente, por ejemplo, un usuario puede unir las funciones necesarias en un solo grupo, bajo un nombre.

Para producir el reporte cada mes, el usuario solamente tiene que acceder la llave o clave con ese nombre.

- Estructura de la base de datos relacional. Es una forma mas simple de base de datos para facilitar al máximo su uso.
- Compatibilidad con las computadoras mas grandes de SPERRY.

Ya que las aplicaciones que usted desarrolle pueden ser usadas como si estuvieran en MAINFRAMES no existe la necesidad de conversiones costosas que consumen mucho tiempo.

- Las interfaces transfieren los archivos de host a host. Si usa numerosos sistemas MAPPER 5, las interfaces están disponibles para aquellos que intercambien datos.
- Funciones gráficas para dispositivos CTR. Columnas completas de números pueden ser rápidamente transferidas en tablas fáciles de comprender.
- Medio ambiente de oficina. El MAPPER 5 se adapta admirablemente dentro del conjunto de cualquier oficina.
- Procedimientos de mantenimiento e instalación. Los técnicos no son necesarios para la instalación o para comenzar los procedimientos de mantenimiento. Los usuarios pueden hacer uso de estas tareas.
- Estaciones de display UTS locales con opciones impresoras locales.
- Con las impresoras locales, las copias en gran escala pueden ser obtenidas inmediatamente sin usar la impresora del sistema.
- La estación de display en gráficas de color UTS 60. Se puede graficar en color.
- La computadora personal SPERRY proporciona un aumento en la flexibilidad, con las características apropiadas. Usted puede usarlas como estación de DISPLAY UTS, al mismo tiempo que obtiene los beneficios de la computación personal.

- Almacenamiento masivo. Un número de dispositivos pueden ser conectados con el sistema MAPPER 5, para proporcionarle almacenamiento de acuerdo a sus necesidades.

3.2.4 BASE DE DATOS MAPPER 1100

BASE DE DATOS MAPPER 1100

- Inicialmente cuenta con 420 modos
- 8 tipos por cada modo
- Un tipo de formato libre (A)
- Cientos y hasta miles de reportes de cada tipo.
- 500 líneas por cada reporte.
- Cada línea hasta de 132 caracteres.

La efectividad de MAPPER se basa en una base de datos con numerosos reportes pequeños.

Se recomienda usar reportes de 200 líneas como máximo.

Los reportes se identifican mediante el número de reportes y una letra.

EJEMPLO: 2B, 35P, 26I.

Todos los reportes que forman un tipo o que están contenidos en él, tienen los mismos encabezados.

Los modos impares solo se pueden leer, mientras que los pares pueden ser actualizados.

3.2.5 FUNCIONES QUE DEBE REALIZAR UNA PERSONA ASIGNADA COMO COORDINADOR DE MAPPER

- Ser responsable de todas las actividades de procesamiento

de MAPPER.

- Controlar el 'hardware' y el 'software' existentes, planear el crecimiento futuro y coordinar el uso y desarrollo de MAPPER.
- Desarrollar e implementar aplicaciones de MAPPER.
- Proporcionar asesoría MAPPER.
- Trabajar con usuarios finales y diseñadores de aplicaciones para establecer requerimientos de reportes, prioridades, justificaciones, horarios o intercambios con otras aplicaciones.
- Determinar el impacto total en el sistema para evitar duplicaciones e interferencias.
- Proteger la información almacenada en la base de datos mediante una copia en cinta.
- Asignar claves de acceso a usuarios.
- Llevar un registro de terminales e impresoras en uso y su ubicación.

3.2.6 TECLAS ESPECIALES QUE UTILIZA MAPPER

- > Esta tecla se denomina SOE y significa "START OF ENTRY"
- ^ La tecla "CIRCUNFLEJO" es la llamada a MAPPER.

Claves de usuarios y passwords.

- Cada usuario deberá usar una clave y un password único para entrar al sistema.

Ejemplo: > J EDUMAP, 12, CVE

- Los passwords deben cambiarse periódicamente.
- Como seguridad adicional puede usarse un password que varíe automáticamente por la hora, la fecha, el mes y el número de terminal.

- Si el usuario excede de tres intentos para entrar al sistema sin poner la clave y/o password correcto, el sistema da la alerta al coordinador y desactiva la clave en cuestión.
- Los intentos de usar ciertas facilidades pueden alertar al coordinador o a otras personas del personal encargado de la administración del sistema.

3.2.7 PASOS PARA VISUALIZAR UN REPORTE

Paso 1 Presionar el CIRCUNFLEJO (^) para desplegar el logotipo MAPPER con la palabra "IDLE"

```
*****MEXICO*****
*   T H E   M A P P E R   S Y S T E M   *
*
*   UNIT: 551   USER: ( I D L E )       *
*****LEVEL 30 R1*****
```

Paso 2 Teclar la clave de usuario y password

Paso 3 La pantalla desplegará nuevamente el logotipo de MAPPER pero ahora con la clave del usuario y el número del modo al cual está conectado.

```
*****MEXICO*****
*   T H E   M A P P E R   S Y S T E M   *
*
*   UNIT: 551   USER: JVB               *
*****LEVEL 30 R1*****
```

Paso 4 Teclar el número del reporte que se desea acceder seguido de la letra correspondiente al tipo de reporte.

Inmediatamente aparecerá en la pantalla el reporte deseado.

3.2.8 LA LINEA DE CONTROL

- La línea de control se utiliza para posicionar un reporte tanto en forma horizontal como vertical.
- LINE > I FMT > RL > SHFT > HLDCHRS > HLDLN > LCS

DONDE:

LINE >

Muestra el número de la primera línea del reporte en la pantalla.

FMT >

Muestra el formato del reporte en la pantalla. Si está en blanco significa que el formato básico está en pantalla.

RL >

Muestra la dirección en giro: Hacia arriba (+ o blanco). hacia abajo (-).

SHFT >

Muestra la dirección del corrimiento a la izquierda (+ o blanco), a la derecha (-).

HLDCHRS >

Muestra el número de líneas retenidas, contando desde la parte superior de la pantalla.

LCS >

Es un indicador de tipo de reporte que se tiene en la pantalla. Un 'LCS' indica que se tiene desplegado el reporte original, mientras que un 'RESULT' indica que el reporte es un resultado obtenido de la ejecución de alguna función como 'SEARCH' y por lo tanto ninguna modificación se hará directamente a la base de datos, sino exclusivamente en la pantalla. Si el indicador es un 'UPD RESULT' indica que es el resultado de una función de actualización, como por ejemplo 'SEARCH

UPDATE'.

3.2.9 FORMATOS DE UN REPORTE

Cuando no se especifica un formato al desplegar un reporte, el sistema asume el formato b sico.

El formato b sico est  formado por las primeras columnas del reporte original.

Existen 6 formatos distintos (opcionales).

USO DEL REPORTE 0 DE CADA TIPO

En este reporte se especifican las caracter sticas de cada uno de los campos que aparecer n en todos los reportes de este tipo (longitud y tipo de campo).

Se especifican tambi n los campos que aparecer n en cada uno de los formatos.

La actualizaci n de este reporte est  restringida al coordinador de MAPPER.

USO DEL REPORTE 1 DE CADA TIPO

Se utiliza para describir los campos de los que consta un reporte.

Sirve como documentaci n de la base de datos.

NOTA:

MAPPER coloca automaticamente un tabulador entre cada uno de los campos de los reportes.

3.2.10 FUNCIONES DE MAPPER

3.2.10.1 FUNCIONES PRINCIPALES

FUNCION	SIGNIFICADO	ACCION
M	MODE	Cambio automàtico de modo
T	TYPE	Despliega en pantalla la lista de los tipos disponibles para el modo que està trabajando.
>	SOE	Indica el inicio de entradas. Sirve para actualizar los datos de un reporte.
+	ADICION	Se utiliza para agregar líneas en blanco de un reporte. (>] N +)
-	ELIMINACION	Sirve para eliminar líneas de un reporte. (>] N -)
X	DUPLICA	Duplica líneas dentro de un reporte. (>] N X. (G))
AR	ADICIONA REPORTE	Permite agregar un nuevo reporte a la base de datos. Creando los encabezados igual a los del rid o del tipo que se defina.
DR	ELIMINAR REPORTE	Permite borrar un reporte y seguridad solo lo puede hacer la persona que lo creó o quien lo actualizó la última vez.
XR	DUPLICA	Se utiliza para crear un nuevo reporte duplicado de uno ya existente o un duplicado de un resultado.
XR	REPLACE	Esta función reemplaza un reporte con un reporte o resultado que se tiene en pantalla por seguridad sólo se podrá reemplazar un reporte que tenga la misma clave.
PSW	WRITE PASSWORD	Asigna, cambia o quita la clave de acceso a un reporte.

RPSW	READ PASSWORD	Limita el acceso a reportes individuales.
------	------------------	---

3.2.10.1 FUNCIONES DE BUSQUEDA

FUNCION	SIGNIFICADO	ACCION
F	FIND	Busca hasta encontrar la primera ocurrencia de los datos especificados en un reporte resultado o en todo un tipo.
S	SEARCH	Busca en un reporte resultado o en todo tipo, todas las líneas de datos que tienen los caracteres especificados en la mascarilla de la función y crea un resultado.
SU	SEARCH	Permite actualizar un reporte con el resultado de una búsqueda.

3.2.10.2 FUNCIONES DE COMPARACION Y DE CLASIFICACION

FUNCION	SIGNIFICADO	ACCION
SORT	SORT	Esta función reorganiza el orden de las líneas en un reporte creando un resultado clasificado.
MA	MATCH	Compara y mueve el contenido de columnas seleccionadas de dos reportes diferentes creando un resultado.
MAU	MATCH	Permite actualizar el reporte receptor en base al resultado obtenido por esta función.

3.2.10.3 OTRAS FUNCIONES

FUNCION	SIGNIFICADO	ACCION
LOC	LOCATE	Localiza un conjunto de caracteres dentro de un reporte o resultado y muestra la línea donde se localizaron; como primera línea de la pantalla.
CHG	CHANGE	Cambia conjuntos de caracteres en cualquier posición de un reporte o resultado y crea un resultado. Calculador interno en fortran.

3.2.11 CORRIDA DE DISEÑOS

Una corrida es un reporte en MAPPER que consiste en un conjunto de instrucciones de control, las cuales especifican paso a paso las instrucciones para generar reportes o resultados.

- LAS CORRIDAS SON UTILES PARA:

- . Obtener procesos repetitivos.
 - . Elaborar programas que faciliten al usuario la operación de sus sistemas.
- Las corridas se elaboran en base a las funciones de MAPPER.

3.2.12 EJEMPLO DE UN SISTEMA MAPPER

El sistema "sico" se encuentra distribuido en la base de datos MAPPER de la siguiente forma:

MOD0 22/23 Información de proveedores

MOD0 24/25 Información sobre pedidos y Mr's

MOD0 26/27 Información sobre requisiciones

Si tomamos la información de pedidos y Mr's (MOD0 24/25) tenemos:

TIPO	CONTENIDO
B	Maestro de pedidos parte 1 (num. del pedido, descripción, fecha del pedido, Proveedor asignado).
C	Maestro de pedidos parte 2 (num. del pedido, costo total, estado del pedido, comprador).
D	Partidas de pedidos (num. del pedido, num. de partida, cantidad requerida, costo por unidad, estado de la partida).
E	Descripción de partidas (num. de pedido, num. de partida, descripción).
F	Maestro de Mr's (num. de Mr's , num. de pedido, cantidad recibida, fecha).
G	Varios.
H	Reportes de impresión de pedidos y documentación del sistema.
I	Programas.

De esta forma el usuario mediante su clave tiene acceso a los modos impares, mientras que los programas accesan los modos pares, para hacer altas de pedidos, de Mr's, de proveedores, etc.

La distribución de los rids de cada uno de los tipos es siempre la misma, de tal forma que a cada centro de trabajo le corresponde un número determinado de rids:

CIUDAD	RIDS ASIGNADOS.
OFICINAS MEXICO	2, 22, 42, 54
OCOTLAN	3, 23, 43, 55
QUERETARO	4, 24, 44, 56

Algunos son manejados como reportes históricos. Otros como reportes de trabajo y otros como reportes auxiliares.

Para la impresión de pedidos MAPPER activa un programa en COBOL que trabaja en BATCH y se encarga de elaborar los pedidos de acuerdo al formato y luego mandarlos a la impresora especificada.

Existen también encargados de dar mantenimiento a los sistemas y cuentan con claves para acceder los modos pares; de forma que un error de un sistema puede ser corregido directamente sobre la base de datos y luego corregir el programa para continuar con la operación.

Todos los sistemas cuentan con interfaces para pasar resultados o datos de un sistema a otro; de forma que por ejemplo todas las polizas de "sico" son pasadas automáticamente a GL + cada cierto tiempo.

Todos los centros de trabajo usan el mismo formato en sus reportes de forma, de tal manera que si un campo no es actualizado, simplemente se deja en blanco.

3.2.13 ALGUNAS FACILIDADES DE MAPPER.

USUARIOS Permite desplegar en la pantalla una lista de todas las terminales que se encuentran en ese momento conectadas a MAPPER, su número, el usuario, la actividad que esta haciendo y la hora en que ejecutó la última transacción.

USER Despliega en la pantalla una lista de las transacciones que han sido hechas en determinada fecha por la clave especificada.

SEND Envía un reporte o resultado a otra terminal. Si la terminal no está encendida; el reporte se

guarda en una lista de espera; si la terminal está encendida el usuario recibe una señal de que tiene mensaje; pero decide en que momento quiere recibirlo.

SS Envía mensajes de texto (1 pantalla) a cualquier terminal del sistema con las mismas características que SEND.

3.3 LINC

3.3.1. DEFINICION.

Es un lenguaje de alto nivel que lleva implícita una fase formada por un sistema administrador de una base de datos en lenguaje cobol, cuyo diseño se fundamenta en una base también automatizada.

Este lenguaje corre exclusivamente en todos los equipos de Burroughs, excepto Micros. Su sistema operativo es el WPF.

3.3.2. CARACTERISTICAS GENERALES.

Debido a su diseño, el análisis de la información se ve afectado directamente y su filosofía tiende a organizar a esta en componentes eventos y perfiles.

NIVEL SUPERIOR	----- OBJETIVOS Y MENSAJES ANALISIS DISEÑO LOGICO -----
NIVEL MEDIO	----- DISEÑO DE BASE DE DATOS DISEÑO FISICO -----
NIVEL INFERIOR	----- COBOL ADMINISTRADOR DE BD SISTEMA FISICO -----

Los sistemas convencionales no organizan un sistema en el nivel superior. LINC no solamente reorganiza el sistema en este nivel sino que demanda una descripción de él.

Este nivel del sistema surge del análisis funcional o estructurado una técnica que se construye alrededor de los conceptos de objeto y mensaje que forman los sistemas de una

compañía y que LINC convierte en componentes y eventos.

3.3.2.1. CONCEPTOS DE LINC.

- Análisis lógico de las entidades administrativas o de negocios en componentes y eventos.
- El diseño de la base de datos es un sistema de transacciones en línea en tiempo real.
- El sistema LINC es manejado por terminales dentro de una red.
- Las transacciones son realizadas en tiempo real y actualizadas en la base de datos haciéndolas inmediatamente transparentes para todos los usuarios del sistema.
- No va más allá de una función en el proceso de la implementación.
- El sistema se concreta a la vida real y su filosofía al mundo real también.
- La técnica de análisis y la herramienta de implementación están diseñadas a la par.
- No existe conflicto ideológico.

Los lenguajes de cuarta generación tienen un enfoque de diseño "top down"

El enfoque top-down de análisis lógico de las entidades y las especificaciones de diseño en forma lógica para la creación del sistema coloca a LINC en la cima de los lenguajes de cuarta generación.

El análisis que se realiza en el nivel más alto:

- Establece los requerimientos del sistema.
- Las funciones y las actividades diarias.
- Los requerimientos estratégicos de la información.

El diseño de este mismo nivel se encarga del proceso de establecer los mapas de las funciones en actividades lógicas

basadas en la filosofía de implementación.

3.3.2.2. TOPICOS DE LA METODOLOGIA.

- Análisis top-down
- Análisis de requerimientos
- Flujo de datos
- Análisis de datos
- Análisis de caminos de acceso
- Normalización en LINC
- Prototipos

El análisis estructurado es la clave del éxito en la creación del Sistema Administrativo.

3.3.2.3. CADA COMPONENTE O EVENTO REQUIERE DE TABLAS.

Las siguientes tres secciones principales deben ser definidas.

SECCION DE INTERFASE

La interfase describe cómo un componente o evento será relacionado con el total del sistema LINC

La interfase por omisión se activa, así como la entrada automática.

SECCION DE DATOS

En esta sección se describe la salida visual a la pantalla, una pantalla tiene 24 líneas por 80 columnas. Las instrucciones "display" le dicen que texto aparecerá y dónde en la pantalla.

Los verbos "line" y "position" localizan los campos texto y los campos datos para ser colocados en los lugares exactos en la pantalla.

SECCION LOGICA

Las instrucciones lógicas describen cómo los campos de datos de las hileras de las tablas son manipuladas. Todas las funciones aritméticas se incluyen; todos los buffers del sistema pueden ser direccionados y usados.

El lenguaje también tiene tablas para acceder las instrucciones; para traerlas de las hileras requeridas a la memoria de los buffers.

Las instrucciones de LINC son descripción y manipulación de tablas orientadas, que requiere de menos conocimientos profundos del medio ambiente del SOFTWARE. Las tablas físicas almacenan hechos y desde aquí están bajo la responsabilidad del producto LINC así como la estructura física de la base de datos.

3.3.3. CARACTERISTICAS ESPECIFICAS.

3.3.3.1. COMPONENTES.

Los componentes describen la información estática y representan la razón de ser del sistema.

El comando usado para declarar un componente es:

Component; <nombre>

Este nombre no puede ser de más de 5 caracteres y debe ser único.

Hay tres tipos de componentes.

- Regulares o estándares
- Memo
- Tabla

Componente regular o estándar.- Es un conjunto de datos con un campo como llave natural de acceso. A la llave se le da el nombre de ORDINATE.LINC trae a memoria principal un

bloque al primer acceso.

Componente tabla.- Es un conjunto de datos con un campo como llave de acceso (ordinate). Se utiliza para pequeños volúmenes de información.

Componente memo.- Es un conjunto de datos sin llave natural de acceso se permiten registros duplicados en este acceso.

3.3.3.2. EVENTOS.

Son los que describen las transacciones que afectan a los componentes.

Los eventos son declarados con el comando event; <nombre>.

El nombre no debe de pasar de cinco caracteres y debe ser único.

Todos los eventos de un sistema son colocados por linc en el mismo archivo, mientras que cada componente memo ocupa un archivo.

El sistema linc se desarrolla a través de una especificación que se escribe en lenguaje de definición linc (LDL).

El propósito de la especificación de linc es describir los diferentes elementos del sistema de especificación de la información al compilador de linc. La especificación linc se divide en varios grupos lógicos, llamados especificaciones. No existe la necesidad de que cada especificación en linc sea etiquetada ya que el compilador reconoce el principio de la especificación dentro del entorno de los comandos asociados con él. Cada SPEC (excepto los datos y las especificaciones lógicas) tienen un comando particular que debe aparecer primero para indicar el principio del SPEC.

3.3.3.3. ESPECIFICACIONES OPCIONALES.

- Crear una base de datos y un sistema nuevo.
- Reorganizar o reconstruir una base de datos II

originalmente creada por linc.

- Crear una versión de prueba de base de datos que permita un sistema mejor a ser desarrollado separadamente de un sistema creado por linc.
- Especificación de entrada decimal en campos numéricos.
- Especificación LOG de transacciones.
- Generar listados de la especificación.
- Establecer o reconstruir una red.
- Facilitar la entrada OFF-LINE.
- Especificar el tipo MCS,
- Permitir la entrada GLI (solamente en sistemas grandes y pequeños.)
- Permitir el uso del formato americano de datos.

3.3.3.4. ESPECIFICACIONES DEL USUARIO.

- Especificar el identificador del usuario, el nombre de la base de datos y el nombre del sistema.
- Declarar el número de periodos contables.
- Definir la distribución de la base de datos sobre los paquetes de disco.

3.3.3.5. ESPECIFICACIONES DE RED.

- Declarar la configuración de red (línea, canal, terminal, dirección, estaciones, etc.).
- Definir el nivel de privilegio para seguridad.
- Especificar los programas ejecutables.

3.3.3.6. ESPECIFICACIONES DE BASE DE DATOS LOGICA.

- Definir la base de datos l gica para la informaci n del DMS II.
- Declarar los caminos de acceso a la base de datos.

3.3.3.7. ESPECIFICACION GLOBAL.

- Definir la l gica global.
- Definir el establecimiento global de los ITEMS DE DATOS.
- Definir las palabras claves.

3.3.3.8. ESPECIFICACIONES DE INTERFASE.

- Especificar el formato de pantalla.
- Especificar el regreso autom tico al formato de pantalla despu s de procesar la entrada a la pantalla.
- Asignar privilegios de seguridad al archivo.
- Especificar l neas de repetici n.
- Identificar la fuente de datos.
- Definir los vol menes del sistema.
- Definir las funciones de TEACH y de ayuda.

3.3.3.9. ESPECIFICACIONES DE DATOS.

- Definir el formateo de la pantalla.
- Describir los datos.
- Definir las especificaciones de campo (tipo, tama o, decimales, datos, justificaci n, iluminaci n, requerimientos, etc.).

- Definir el acceso a la base de datos en los componentes "ordinate."
- Definir el uso del campo (solamente de entrada, de entrada y salida o solamente de entrada).
- Especificar la lógica de la pre-pantalla.
- Insertar la lógica global.

3.3.3.10 ESPECIFICACIONES LOGICAS.

- Especificar funciones aritméticas.
- Definir acceso controlado de usuario y actualización de la base de datos.
- Especificar los comandos lógicos.
(DO, WHEN, JUMP.TO, ETC.)
- Insertar la lógica global.
- Definir el manejo de mensaje y error.
- Especificar la edición y verificación de datos.

3.3.3.11 COMANDOS QUE SE UTILIZAN PARA CREAR LAS ESPECIFICACIONES.

3.3.3.11.1 COMANDOS DE LAS ESPECIFICACIONES OPCIONALES.

Dentro de estas especificaciones tenemos:

- INPUT LOG: Compila el programa LOG. Este programa LOG, crea y accesa la bitácora de transacciones.
Por omisión.
- NO INPUT LOG: Contraria a la opción anterior.
- LIST: Permite listar todas las tareas.
- NO LIST: Contraria a la opción anterior. Por omisión.

NETWORK: Permite generar una red. Es usado cuando un sistema utilizará una red específica.

NO NETWORK: Por omisión. Indica que no se generará una red dentro de la especificación.

NETWORK ONLY: Indica que esta especificación generará una red sin crear una Base de Datos.

NEW DATABASE: Construir una nueva base de datos.

REORG.DB: Reorganiza una base de datos ya existente y genera una nueva base de datos.

SEPARATOR (char): Se utiliza para separar caracteres de un campo en la entrada y salida. Char., puede ser cualquier carácter excepto de A-Z ó 0-9.

NO SEPARATOR: No permite separar caracteres. Por omisión.

TEST VERSION: Crear una base de datos y un sistema en modo de prueba.

US DATE: El formato de la fecha americana (MM/DD/AA).

ZERO SUPPRESS: Suprime los ceros a la izquierda.

NO SUPPRESS: Por omisión. No suprimir los ceros a la izquierda.

DECIMAL.CHAR (char): Especifica el carácter para representar un decimal Char puede ser un (.) ó una (,).

DECIMAL KEYED: Obliga a teclear el punto decimal al capturar un campo real en todos los campos del sistema.

NO DECIMAL KEYED: Es por omisión. No obliga a teclear el punto decimal al capturar los campos de tipo real en todas las variables del sistema.

3.3.3.11.2. COMANDOS DE LAS ESPECIFICACIONES DEL USUARIO.

Las especificaciones que están dentro de esta categoría son:

ACTIVE MONTHS (AM): Combinando con el comando MONTHLY. VOL en las especificaciones de interfase de los eventos determinan la cantidad de espacio reservado para los eventos en el disco.

Sintaxis AM; <número>

El número especifica la cantidad de meses que se desean almacenar los eventos, antes de que sean removidos de la Base de Datos a través de LIRC.

AUDIT PACK (AP): Especifica el paquete en el cuál reside el archivo de auditoría. Si este comando se omite, el sistema utilizará el paquete especificado por DICT.PACK.

Sintaxis: AP; <pack>

DATABASE (DB): Esta especificación se utiliza para declarar el nombre de la base de datos.

Sintaxis: DB;<nombre>

El nombre deberá tener como máximo 10 caracteres y se recomienda que los tres primeros caracteres deberán ser únicos.

DEFAULT PACK (DF): Declara el paquete en donde se almacenaron los eventos y componentes. Si se omite este comando LINC utilizará el paquete llamado 'SYSTEM'.

Sintaxis: DF; <paquete>

DICT PACK (DP): Declara el paquete en donde reside el diccionario de la base de datos y el archivo de re-inicio de auditoría. Si este comando se omite el sistema utilizará el DEFAULT PACK.

Sintaxis: DP; <paquete>

EVENT.PACK (EP): Especifica el paquete en donde se almacenarán los eventos. Este comando tiene prioridad sobre el DEFAULT PACK. Si se omite se utilizará el pack definido por DEFAULT.PACK.

Sintaxis: EP; <Paquete>

SYSTEM NAME (SN): Declara el nombre del sistema. Es obligatorio colocar esta especificación en todos los sistemas.

Sintaxis: SN; <nombre>

El nombre deberá tener como máximo 6 caracteres.

TEST PACK (TSP): Especifica el paquete en donde residirán los registros de la base de datos y perfiles de la versión de prueba del sistema.

Cuando se utiliza TEST.VERSION (Especificaciones opcionales) y no se utiliza TEST.PACK, LINC utiliza el paquete por omisión.

Sintaxis: TSP; <paquete>

USER: Declara el nombre del usuario del sistema. Este comando es requerido y deberá ser el primero en la sección de especificaciones del usuario.

Sintaxis: USER; (<encabezado>)

El encabezado deberá tener como máximo 50 caracteres, sin incluir los paréntesis.

EJEMPLO:

```
USER; (FACPYA)
DATABASE; ESCOLARDB
SYSTEM NAME; ESCOLAR
DICT PACK; B
DEFAULT PACK; SYSTEM
AUDIT PACK; C
EVENT PACK; B
TEST PACK; A
```

3.3.3.11.3. ESPECIFICACIONES DE RED.

Estas especificaciones describen la red asociada con el sistema.

NETWORK: Indica el inicio de la red y también le asigna un nombre. Este comando es requerido. Si este comando se utiliza, es necesario declarar las terminales que usará el sistema, cuales serán privilegiadas etc.

Sintaxis: NETWORK; <nombre>

El nombre deberá ser único y máximo deberá tener 10 caracteres.

Las especificaciones que describen a los componentes y eventos se dividen en 3 secciones.

INTERFASE: Sólo requiere del comando COMPONENT; <nombre> y declarar el tipo de componente. Si fuera un evento entonces el comando sería

EVENT; <nombre>

DATA: Consiste de los comandos DISPLAY y DATA utilizados para formar la pantalla del componente o evento. En esta sección se especifican los comandos de edición.

LOGICA: Se coloca la lógica asociada a los componentes o eventos.

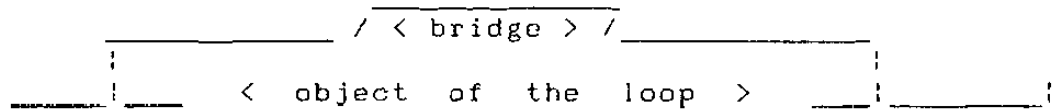
3.3.3.12 SINTAXIS DE LOS DIAGRAMAS.

Los diagramas de LDL se leen de izquierda a derecha y se llaman RAILROAD.

- El final del diagrama se indica por una barra vertical. La continuación por un número al final de la primera línea y el principio de la segunda.
- La información contenida dentro de los paréntesis angulares es información variable que es suministrada por el escritor de la SPEC. Cada cosa que se teclee aparecerá en la misma forma que se haga, si hay abreviaciones deberán ser subrayadas.
- El signo (=) igual puede ser substituido por (;) punto y

coma en todas las ocurrencias.

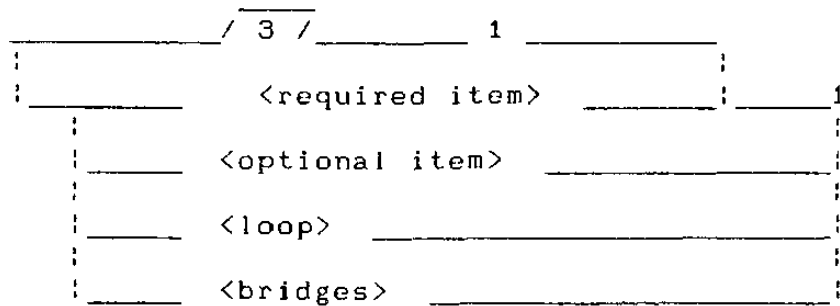
3.3.3.13 EJEMPLO DE UN DIAGRAMA RAILROAD.



Required items.- No alternan el curso a través del diagrama existen para items requeridos.

Optional items.- Los items listados verticalmente abajo de la línea principal del diagrama indican, items opcionales.

Loops.- Un Loop o lazo es un camino RECURRENTE a través de un diagrama RAILROAD el formato es:



Un loop o lazo debe ser usado en la dirección indicada por la flecha. Sin exceder los límites especificados.

Bridges.- PUENTES.- Un puente indica el número de veces que se va a realizar un lazo.

El número en el puente puede ser precedido por un asterisco (*). Esto significa que el lazo se puede realizar al menos una vez.

3.3.3.14. SEMANTICA.

Un operador es cualquier cosa después de (;) esto es, una variable u otro comando.

3.3.3.15. COMO SE DEFINE UNA ENTRADA DE DATOS EN LINC.

INSTRUCCIONES DEL LENGUAJE LINC.

INTERFACE: Event; Sales: Order entry **
 Repeat. From = 6

DATA:	Display; (order number)	Li; 3 pos; 15
	Display; (customers number)	Li; 3 pos; 45
	Data; orderno	Li; 3 pos; 30 Le; 6
	Data; custname	Li; 3 pos; 60 Le; 8
	Display; (part number)	Li; 5 pos; 15
	Display; (quantity)	Li; 5 pos; 30
	Data; partnumber	Li; 6 pos; 15 Le; 9
	Data; quantity	Li; 6 pos; 30 Le; 6

LOGIC:

(NONE)

- Linc genera la salida de la pantalla completa como se lista abajo.
- No se requiere lógica para la entrada de los datos básicos.
- Linc automáticamente generará el código necesario para cada línea que entre en la base de datos del sistema.
- El 'Last-Line' es un indicador proporcionado automáticamente por el sistema, al usar el comando 'REPEAT.FROM'.

Su valor para la primera línea es 001 y se incrementa en 1 por cada línea o grupo de líneas que se repite hasta que el operador tecléa el 999 para indicar que la transacción actual es la última.

PANTALLA DE SALIDA.

```

[1 S M]          [TRANMO]          [DATE]

ORDER NUMBER [    ]  CUSTOMER NUMBER [    ]

PART NUMBER          QUANTITY          LAST LINE
[    122]           [    1]           [001]
[    222]           [    1]           [002]
[    311]           [   12]           [003]
[    333]           [    3]           [004]
[    544]           [    2]           [999]
[      ]           [    ]           [  ]
[      ]           [    ]           [  ]
[      ]           [    ]           [  ]

```

"Repeat from" permite la entrada múltiple de líneas de transacción comenzando de cierto número de línea en la pantalla.

El "order number" y el "customer order" son comunes y se duplican automáticamente en cada entrada en la línea de transacción.

3.3.3.16. IDENTIFICACION Y CLASIFICACION DEL DOMINIO.

Un sistema generador de necesidades necesita conocer los límites de los elementos de los datos para que sea posible generar la relación apropiada entre esos conjuntos de datos.

Dimensionar el dominio en linc es parte de la llamada conversión de datos.

3.3.3.16.1. CLASIFICACION HORIZONTAL DE DOMINIO.

TABLE =	ORDER		VENDOR			AMOUNT
PAYMENTS	NUMBER		NUMBER			
TABLE =	ORDER		VENDOR	PART	QUANTITY	AMOUNT
INGOODS	NUMBER		NUMBER	NUMBER		
TABLE =	ORDER	CUSTOMER		PART	QUANTITY	AMOUNT
SALES	NUMBER	NUMBER		NUMBER		
TABLE =	ORDER	CUSTOMER				AMOUNT
RECEIPS	NUMBER	NUMBER				

Linc reconoce las dimensiones de los dominios horizontales y verticales que se establecen automáticamente al asignar el nombre correcto de un elemento de datos.

CLASIFICACION DEL DOMINIO HORIZONTAL. Cada evento tiene referencias del dominio únicas a su tipo de transacción. Las referencias son elementos de identificación de una transacción (=evento).

TABLA =	ORDER	CUSTOMER	PART	DATE	QUANTITY	AMOUNT
SALES	NUMBER	NUMBER	NUMBER			
	REF	REF-KEY	REF-KEY	REF		
	111	15	122	831111	1	8000
	111	15	222	831111	1	12000
	111	15	311	831111	12	9600
	111	15	333	831111	3	3600
	111	15	544	831111	2	10000

Una transacción linc (=evento) registra un cambio en el estado.

Solamente se mantienen los cambios y las referencias en una transacción.

3.3.3.16.2. CUALIFICACION DE DATOS.

COLOM NAME	HORIZONTAL	DOMAIN	DIMENSION
QUANTITY	:	SALES-QUANTITY	
	:		
AMOUNT	:	SALES-AMOUNT	

Los conjuntos de datos son cualificados con sus nombres de tablas.

3.3.3.16.3. REFERENCIAS DE DOMINIO.

Example:

QUANTITY SOLD es un "número" de partes requerido por un "customer" en un "order" particular en un momento particular en "time".

SALES se refiere a:---- order number
 ---- customer number
 ---- part number
 ---- date stamp

Todas las referencias a los eventos son checadas automáticamente y deben ser encontradas como válidas; para aceptar el evento como una transacción válida, de otro modo

los mensajes de error serán desplegados.

3.3.3.16.4. CLASIFICACION DEL DOMINIO VERTICAL.

Los tipos de transacciones son normalmente complementarios.

El resultado de esos juntos forma el estado actual del negocio.

Ejemplo: "In goods" < - > payments.

El "vendedor" esta pagado, cuando los pagos son iguales a la cantidad "ingoods".

ingoods < - > sales

Para cada "part number" la cantidad de "ingoods" menos la "sales quantity" determina el nivel de "stock-on-hand".

sales < - > receipts

La factura esta pagada, cuando las recepciones totales en esa factura son iguales a la cantidad en la factura.

TABLE =	ORDER		VENDOR			AMOUNT
PAYMENTS	NUMBER		NUMBER			
TABLE =	ORDER		VENDOR	PART	QUANTITY	AMOUNT
INGOODS	NUMBER		NUMBER	NUMBER		
TABLE =	ORDER	CUSTOMER		PART	QUANTITY	AMOUNT
SALES	NUMBER	NUMBER				
TABLE =	ORDER	CUSTOMER				AMOUNT
RECEIPTS	NUMBER	NUMBER				

Las dimensiones del dominio vertical estan cruzando los limites de la tabla lógica.

Esto se establece con los "perfiles" que hacen que dos o más tablas de eventos lógicos parezcan una nueva tabla.

Un "profile" dá un punto de vista del estado particular

del negocio.

3.3.3.16.5 LINC ES AHORA UN LENGUAJE DE PROCEDIMIENTO.

Los componentes y los eventos se definen como conjuntos separados de los demás con su propia interface y sus descripciones de datos y lógica.

```
-----|
| EVENT = BALANCE
-----|
| EVENT = STOCK-ON-HAND
-----|
| EVENT = PAYMENTS
-----|
| EVENT = INGOODS
-----|
| EVENT = RECEIPTS
-----|
| EVENT = SALES
-----|
| COMPONENT = PART
-----|
| COMPONENT = CUSTOMER
|
| INTERFACE SPECIFICATION
| DATA AND DISPLAY SPECIFICATION
| LOGIC SPECIFICATION
|-----|
```

El reconocimiento de los límites de dominio por el generador de linc.

- Permite la definición separada de componentes y eventos.
- Revisa los errores relacionales en todos los elementos de datos definidos entre cada componente y evento.
- Establece la interface en forma automática entre evento y componente en forma apropiada en todo el sistema linc.

La relación de datos es generada y verificada automáticamente por el compilador linc, pero este está programado manualmente en lenguajes de tercera generación y

es una de las áreas más propensas a errores en la programación convencional.

El aumento de tiempo en el desarrollo del sistema como función de complejidad cuando se encuentra en un medio ambiente de lenguajes de tercera generación no es reconocido por Linc.

3.3.3.17 QUE ES LIRC?

- Es un sistema de transacciones en línea y en tiempo real.
- LIRC complementa a LINC en el área de reposicionamiento.
- LIRC es un compilador que genera programas separados (standalone), permitiendo al usuario final procesar todos los tipos de reportes y los programas BATCH para hacer de LINC un sistema completo de soluciones.
- LIRC direcciona las siguientes áreas:

3.3.3.17.1. REPORTE:

Todos los tipos de reportes de listados y resúmenes pueden ser producidos por aparatos de salida:

- LINE PRINTER
- TERMINAL PRINTER
- VIDEO TERMINAL

3.3.3.17.2. OMISIONES FISICAS:

Linc puede solamente hacer bajas lógicas. Las bajas físicas deben ser programadas con LIRC.

Este proporciona un mecanismo de seguridad y control apropiado para administrar las transacciones en línea.

3.3.3.17.3. ARCHIVOS EXTRACT Y ARCHIVOS SORT.

- Los archivos extract contienen los datos extraídos de la base de datos linc.
- Estos archivos pueden ser ordenados y usados para la entrega a la impresora.
- Los archivos extractados pueden también ser almacenados como información histórica.

3.3.3.17.4. COMUNICACION ENTRE DATOS.

Lirc puede transferir registros de transacciones llamados "auto.entries", con un programa LIRC de un sistema LINC a otro sistema LINC

3.3.3.17.5. INTERFASES DEL SISTEMA NON-LINC

Los archivos extraídos pueden ser usados en una basebatch para transferir datos de un sistema LINC y otro sistema NON-LINC o viceversa.

3.3.3.18. COMO SE DEFINE UN REPORTE EN LIRC ?

Una página de reporte consiste de un número específico de líneas (longitud de página y una línea consiste de un número específico de caracteres (amplitud de página).-

De acuerdo con sus líneas de funcionabilidad una página de reporte las agrupa en ciertos tipos de bloques de información que pueden ser llamados "FRAMES."

La información de una página de reporte puede ser colocada en bloques o en paneles en cualquier tipo de grupo de líneas, por ejemplo:

- Líneas de página o de principio de página.
- Encabezados de columnas.

- Línea(s) de detalle.
- Línea(s) de total y subtotal.
- Línea(s) de información general.

3.3.3.18.1. SALIDA DEL PANEL (FRAME).

El panel es una imagen del "BUFFER" de una cierta parte de una página de reporte.

El texto es posesionado con las instrucciones "DISPLAY" en esos paneles.

Las instrucciones de datos posesionados describen los campos a ser cargados con información de una columna de una tabla seleccionada de la base de datos LINC.

3.3.3.18.2. LOGICA PRINCIPAL.

La lógica principal describe el orden en el cual los paneles se imprimen; esta lógica también determina el número de las páginas de reporte a ser impresas.

Como se compone un reporte en LIRC.

FACTURA NUMERO DE ORDEN	JONES MAPLE ROAD DEARBORN			FRAME 1 LOS CAM POS DE DATOS ESTAN SUBRAYADOS
NUMERO DE PARTE	DESCRIPCION	CANTIDAD	PRECIO	
122	LINC	1	8000	FRAME 2
222	PROCESSOR	1	12000	REPETIDO
311	VIDEO TERMINAL	12	9600	UNO X
333	TERMINAL PRINTER	3	3600	NUMERO
544	DISK DRIVE	3	15000	DE VECES
CANTIDAD TOTAL			48200	FRAME 3

Los campos de texto y los campos de datos son descritos en un panel con instrucciones "display" y "data" y son posesionados en esos paneles con las instrucciones de línea; posición y longitud.

LOGICA PRINCIPAL:

```

PAGE - - - - - DETERMINA IMPRESION ACTUAL
LOOP          |          PRINT FRAME: 1
              |
LINE- - - - - DETERMINE: CADA FACTURA (ORDEN DE
LOOP          |          PRINT FRAME: 2      CLIENTE)
              |
              | - - -END:
              |   PRINT FRAME:3
              |   ADVANCE: NEW.PAGE
              |
              | END

```

La lógica principal describe el orden en el cual los paneles se imprimen.

Las transacciones correctas son cargadas con instrucciones "DETERMINE."

3.3.3.18.3. PANEL LOGICO:

Los campos de datos cualificados de la base de datos LINC son cargados automáticamente dentro de los campos panel antes de imprimir este.

La hilera de la derecha de la tabla debe ser cargada en el panel lógico.

El panel 1 y el 2 contienen una búsqueda lógica para cargar la hilera de la derecha de la tabla de clientes y la de la tabla de partes.

3.4. POWERHOUSE

3.4.1. DEFINICION:

Es un lenguaje de cuarta generaci3n que surgi3 en 1981, en Ottawa, C3nada en la empresa Quasar Systems, debido a la inquietud de hacer el desarrollo de sistemas de una manera m3s r3pida, f3cil y a menor costo. Al principio s3lo se pod3a procesar en equipos HP3000, cuyo sistema operativo es el MPE, y gracias a un subsidio parcial por parte de Digital y Data General se logr3 el desarrollo de una versi3n para los equipos VAX y la serie MV respectivamente.

Debido a sus caracter3sticas generales, espec3ficas y a su orientaci3n, as3 como todos los conceptos que mencionamos a continuaci3n, el POWERHOUSE es considerado como un lenguaje generador de reportes y de aplicaciones para profesionistas en sistemas y para el usuario final como un explotador de la informaci3n.

3.4.2. CARACTERISTICAS GENERALES:

Powerhouse est3 formado por cuatro componente b3sicos:

- El diccionario de datos. Dictionary plus.
- El dise3ador de pantalla para procesos interactivos. Quick.
- El generador de reportes. Quiz.
- El procesador de datos en lote. QTP.

Y por dos componentes opcionales:

- El m3dulo de consultas "no programadas" THE EXPERT.
- El m3dulo creador de gr3ficas administrativas. GRAPHICS.

3.4.3. CARACTERISTICAS ESPECIFICAS.

3.4.3.1. DICTIONARY PLUS.

El diccionario de datos permite utilizar el dictionary/3000 (HP) o el QDD (COGNOS).

El QDD establece la estructura completa de información que va a contener el diccionario de datos definiendo cuatro secciones.

- Definición del usuario.
- Definición de archivos..
- Definición de elementos.
- Definición de registros / ITEMS (campos)

En la definición de usuarios se incluyen todos los "PASSWORDS" para acessar la información que forma parte del sistema.

En la definición de archivos se especifican todos los archivos, sus capacidades, tipos y restricciones.

En la definición de elementos se deben incluir los elementos, sus tipos y tamaños; de manera opcional la descripción del elemento y algunas características como encabezado, letrero para la terminal, máscaras etc.

En la definición de registros se especifican todos los registros mencionados y cuales son llave.

La relación entre los diferentes archivos que componen el diccionario se establece al definir los registros detalle y especificar sus ligas correspondientes.

3.4.3.2. QUICK.

El módulo Quick es con el que se diseñan las pantallas, estas pueden ser tan sofisticadas como el usuario las desea agregando solamente dentro de un programa una serie de verbos relacionados con el manejo de pantallas de muy fácil uso.

Powerhouse por omisión diseña pantallas tomando como base la

longitud de campos y las características de estos, definidas en el diccionario de datos.

3.4.3.2.1. COMANDOS DE QUICK.

QDESIGN: Es el que genera las pantallas del QUICK.

COMANDOS DEL QDESIGN:

_ Para la descripción de la pantalla o de un campo.

```
DESCRIPCION [OF] SCREEN / field string  
[[,]string]....
```

SCREEN: Especifica que se está proporcionando la descripción de la pantalla.

FIELD: Especifica que la descripción del campo puesto en field se está proporcionando. El campo debe haberse declarado previamente en un estatuto "field".

STRING: Especifica una línea de texto descriptiva.

_ Para dibujar una línea o cuadro en la pantalla:

```
DRAW: [option] [from] [row] [, column1] [to][row2]  
[column2] [from] [row1] [,column1] [tolrow2]  
[,column2]
```

Proporciona las coordenadas del cuadro que se dese dibujar. La primera es la esquina superior izquierda; y la segunda es la inferior derecha.

Las opciones son:

CHAR: Dibuja líneas usando el carácter especificado.

DOUBLE: Dibuja líneas dobles.

THICK: Dibuja líneas gruesas.

THIN: Dibuja líneas delgadas.

- Para crear campos en la pantalla:

Field item [option]....

ITEM: Especifica un item grabado, definido o temporal.

Algunas opciones son; character, date format, fill, display, etc.

- Para obtener ayuda en líneas:

Help [keyword]

- Para asignar características de luminosidad.

hilite [object] [highlight] [[,] [object]
[highlight]

- Para llamar QSHOW desde QDESIGN se tecllea QSHOW.

Sirve para preguntar sobre las entidades en el diccionario tales como elementos, archivos, registros.

El prompt del QSHOW es el guión.

- Para grabar los estatutos fuente de QDESIGN.

save mpename [clear]

MPENAME: Es el nombre del archivo permanente donde se guardarán los estatutos fuente.

CLEAR: Limpia el contenido de los estatutos fuente de QDESIGN después de haber grabado.

- Para poner "candado" a un archivo o base de datos:

lock file [base/file] [file [base / file]]

- Para desplegar un valor en un campo display field [from item]

FROM ITEM: Se usa para especificar que serán desplegados los valores obtenidos del item.

3.4.3.3. QUIZ.

Este módulo es el encargado de generar los reportes en POWERHOUSE.

Los puede generar como reportes automáticos o como reportes estilizados; los dos son "non-procedual" facilitando así la programación.

Interactua con el diccionario de datos para obtener los encabezados de los elementos que forman el reporte así como su tipo, tamaño y máscaras de edición.

Se pueden definir las columnas que forman el reporte, los saltos de página, de renglón, los letreros especiales etc. Así como los cortes dentro del registro, sumarizar la información deseada, sacar promedios, valores máximos, mínimos y porcentajes.

3.4.3.3.1. COMANDOS DE QUIZ.

- Para especificar el contenido y los formatos de pie de página.

FINAL FOOTING [REPORT - GROUP]

- Para especificar el formato y contenido de los mensajes que aparecerán en cada corte de control.

FOOTING AT SORT-KEY [REPORT - GROUP]

- Para especificar el contenido y formato de los encabezados de control.

HEADING AT SORT-KEY [REPORT - GROUP]

- Para especificar el contenido y formato de las líneas detalle de los reportes.

REPORT [SUMMARY] [REPORT - GROUP / ALL [SKIP n]]

- Para seleccionar las condiciones de los archivos y registros a reportar:

SELECT [FILE] [IF CONDITION] AND SELECT [IF CONDITION]

- Para definir los cortes:

SORTED [ON ITEM [A / D] RESET PAGE [n]]].....]

3.4.3.4. QTP.

Es el módulo que maneja los archivos y esta orientado a los procesos en batch.

Una corrida de QTP está formada por una o más requisiciones para realizar procesos relacionados.

Una requisición de QTP lee a través de uno o varios archivos de entrada generalmente para actualizar uno o más archivos de salida.

Dentro de la requisición hay tres fases: DE ENTRADA, ORDENACION Y SALIDA.

En la fase de entrada del QTP construye un conjunto de transacciones basadas en los campos de entrada.

Las transacciones son clasificadas y finalmente se actualizan los archivos.

3.4.3.4.1. COMANDOS DEL QTP.

- Para clasificar archivos de entrada y sus relaciones lógicas.

ACCESS file 1 / * subfilename [alias name 1]

[link linkage file 2/ * subfilename 2

[alias name 2] [optional].....

- Para grabar una corrida compilada de QTP en el archivo especificado en mpename.

BUILD [mpename]

- Para extraer registros del archivo primario por el valor de la llave:

CHOOSE KEY [GENERIC / NOGENERIC] [VALUE [TO VALUE] [(,] VALUE [TO VALUE]].....]

- Para editar items y parámetros de tiempo de ejecución.

EDIT ALL / file / item [option].....

- Para ejecutar una corrida de QTP compilada. El INPUT LIMIT es el número máximo de transacciones a ser seleccionadas de entrada, y el process limit, especifica el número máximo de transacciones que serán procesadas.

EXECUTE mpename [INPUT [LIMIT] n1] [PROCESS [LIMIT] n]

- Para iniciar la ejecución de una corrida de QTP

GO

- Para definir los archivos de salida y acciones como ADD, UPDATE, DELETE, ADD UPDATE.

3.4.3.5. THE EXPERT.

Es un sistema de información "úselo usted mismo"; puede ser utilizado por cualquier tipo de usuario; analista, usuario final o administradores expertos consta de tres componentes: EL EXPERT, EXPERTUSER UTILITY Y EXPERT - INDEX UTILITY.

Todos los usuarios pueden usar seis pantallas desplegando información y creando reportes. Para crear un reporte, el usuario selecciona un índice de la pantalla y da el nombre del reporte en la pantalla de reportes. Luego selecciona las columnas que quiere en la pantalla de columnas y los valores en la pantalla de valores.

Finalmente en la pantalla de resultados el usuario decide si desplegar el reporte en la terminal o en la impresora o enviarla como entrada al powerplan, un paquete financiero.

3.4.3.6. GRAPHICS.

Es una herramienta de software que crea gráficas administrativas presentables en segundos. Consta de más de

15 diferentes gráficas o combinaciones de gráficas que pueden producirse: verticales, horizontales, líneas, puntos, pie stacked, bar.

Para manejarlas no se necesita ser usuario experimentado.

Graphics utiliza muchos de los comandos del quiz. Algunos de los comandos especiales son: colour, frame graph, etc.

3.4.4. COMO MANEJAR POWERHOUSE.

En editor se insertan todas las definiciones y descripciones de datos grabándolas con el "nombre esquema".

Se compila el diccionario de datos llamando A QDD y dando el "nombre esquema" que se desea compilar. Si no surgen errores inmediatamente se crea el esquema y el archivo raíz de la base de datos " IMAGE" de los archivos MPE o KSAM. Si hubo errores, se tienen que corregir con el editor y volver a compilar.

Hay que crear la base de datos image, indicando que realice la creación física de la base o bases de datos definidas en el diccionario.

Luego se desarrollan los programas; estos se pueden realizar de dos formas:

- Las instrucciones pueden grabarse por medio del editor como un archivo de texto en quick o QTP generando un archivo de código que se graba con el "nombre objeto".
- Las instrucciones que se graban en QUIZ son interpretadas al momento de ejecutarse.

La operación de estos programas se hace llamando a los comandos respectivos de acuerdo con el módulo en el que se hallan desarrollado. Por ejemplo: si se desea ejecutar un programa desarrollado en quick se llama a los comandos de quick y de la misma forma se operan los demás módulos.

3.4.5. AYUDA EN LINEA.

Si se está trabajando en el módulo quick y se desea ayuda

sobre ese campo basta con teclear el signo "?" para recibirla. Lo que hace powerhouse al teclear "?" es checar si ese campo tiene la opción HELP, si no la tiene va al diccionario de datos QDD y checa si se incluyó la opción de HELP en la definición del campo, si se incluyó, despliega un mensaje de ayuda, si no, por default el mensaje de ayuda contiene el campo, tipo y tamaño.

También existe una opción de ayuda llamada "DESCRIPTION" que es implementada en el diccionario de datos y que puede ser accesada por medio de los signos "??", los cuales despliegan una pantalla de información acerca de lo que deseamos conocer se regresa al sistema tecleando "RETURN".

3.4.6. RUTINAS EN OTROS LENGUAJES.

Por medio del powerhouse se pueden hacer llamadas a rutinas en otros lenguajes con el comando "do external". Estas rutinas pueden estar en COBOL, FORTRAN Y SPL.

El comando es: DO EXTERNAL, parametro 1, parametro 2.

Los parametros pueden ser archivos, campos de un archivo o variables temporales. Las rutinas se deben compilar y almacenar en el grupo o cuenta con la que se esta trabajando o en librerías segmentadas del sistema.

Al terminar la rutina externa, el programa continua su proceso en la siguiente instrucción "DO EXTERNAL"

3.4.7. MANEJO DE DIFERENTES TIPOS DE ARCHIVOS.

Se pueden usar y manejar tres tipos diferentes de archivos en POWERHOUSE.

MPE.	SECUENCIALES
KSAM.	INDEXADOS
IMAGE.	MANUAL, AUTOMATICO Y DETALLE.

Los archivos que van a ser usados en los programas se

definen con la instrucción file. Si se usan varios archivos relacionados entre si, se agrega la opción LINK TO.

Los procedimientos de manejo de lectura, escritura, bajas, cambios y consultas a archivos no necesitan especificarse para los programas hechos en quick ya que se generan automáticamente.

La base de IMAGE tiene estructura de RED lo que permite mayor flexibilidad de acceso a los archivos y a los registros en cualquier momento.

- Las condiciones de error por manejo equivocado de archivos se presentan al momento de la ejecución y son manejados automáticamente desplegando un mensaje que explica la causa del error. Las condiciones de error se pueden manejar manualmente.

3.4.8. USUARIOS A LOS CUALES ESTA ENFOCADO EL POWERHOUSE.

Los módulos quiz y quick están enfocados a los usuarios finales.

- Los módulos QDD (diccionario de datos) y QTP están enfocados hacia personal de sistemas que conozcan estructuras de datos y actualización de procesos.

El módulo expert está enfocado a utilizarse por programadores casuales.

3.4.9. UTILERIAS DE APOYO.

Entre las utilerías de apoyo que proporciona el POWERHOUSE tenemos:

QUTIL: Crea y borra archivos físicos y bases de datos definidas en el diccionario de datos.

ITOQ: Construye un diccionario de datos fuente, a través de definiciones existentes en la base de datos.

QCOBLIB: Genera definiciones de datos en COBOL de los archivos declarados en el diccionario de datos.

QSHOW: Provee el contenido del diccionario de datos en pantalla o papel.

QUIZ 400p y QTP 400p: Son utilerías para mejorar el "PERFORMANCE" en ejecución de un programa escrito en QUIZ o QTP.

3.4.10. OTRAS VERSIONES DE POWERHOUSE.

Powerhouse de Cognos es un producto de programas para el mercado cooperativo de la compañía Cognos. Powerhouse es un sistema generador de aplicaciones MIS / DP Powerhouse usa su propio sistema de directorio o interfaces con VAX CDD para que los usuarios de enable tengan acceso de lectura y de escritura a los archivos existentes RMS. Una ventaja clave de este producto es que influencia las ventas VAX en medio ambiente MIS de DATA GENERAL o HEWLETT-PACKARD para clientes que ya usan o preguntan por Powerhouse.

3.4.10.1. TIPO DE USUARIO.

Powerhouse es usado por profesionales de Informática quienes desarrollan aplicaciones para la producción en compañías pequeñas, medianas y en unidades divisionales o departamentales.

Otro tipo de usuarios son los usuarios finales sofisticados (analistas financieros y de negocios, ingenieros y técnicos), que tienen conocimiento para desarrollar aplicaciones y que al mismo tiempo son usuarios de Powerhouse. Las aplicaciones terminales pueden ser realizadas al darle al usuario final su propio escrito de reportes a la medida.

3.4.10.2. TIPO DE DESARROLLADOR.

El Powerhouse es un lenguaje usado por profesionales de procesamiento de datos. Las aplicaciones se desarrollan usando el enfoque prototipo de "ruta directa". Al concentrarse en las funciones y las características más bien que en las apariencias visuales, los desarrolladores del

Powerhouse pueden rápidamente entregar aplicaciones terminadas que satisfagan los requerimientos de los usuarios finales.

Los usuarios finales sofisticados pueden construir sus propias aplicaciones y típicamente desarrollará sus propios reportes ad - hoc.

Otra de las características del Powerhouse es la habilidad para generar las definiciones de datos compatibles C.D.D.

3.4.10.3. CODIGO GENERADO/INTERPRETE.

El Powerhouse no genera ningún código, ni lenguaje de tercera generación, ni ninguno otro. Los compiladores del Powerhouse producen archivos de parametros binarios que son leídos por módulos ejecutables del Powerhouse. Powerhouse está en lenguaje "C". Powerhouse no es un interpretador aunque su arquitectura manejada por tablas permite utilizarlo como un interpretador. Los usuarios pueden ejecutar sus programas fuentes o compilados.

3.4.10.4. GENERADOR DE APLICACIONES.

Es su principal objetivo. Se usa para generar aplicaciones de producción.

Contexto del centro de aplicaciones o del centro de información.

3.4.10.5. APLICACIONES TIPICAS.

El Powerhouse está disponible para aplicaciones que implican tareas simples y complejas respaldando un número medio de terminales; por ejemplo: Una VAX 8200 con 5 megabytes, con 30 o 35 usuarios simultáneos.

3.4.10.6. INTERFASE CON QBM.

El Powerhouse corre en VAX, HEWLETT- PACKARD y Computadoras DATA GENERAL.

No tiene su propia interface con IBM, pero cualquier interface ingeniosa DECnet / SNA a la VAX puede ser usada.

3.4.10.7. MEDIO AMBIENTE BASE DE DATOS.

El Powerhouse está diseñado para ser separado y distinto de los sistemas administradores de archivos que lo soportan. Un diccionario de datos comprensivo describe todos los archivos y los elementos usados en una aplicación incluyendo los formatos de campos y las reglas de edición tan bien como la seguridad de las aplicaciones.

Powerhouse es considerado un producto del centro de aplicaciones. Su propósito es ser procesador de reportes, también es ser usado por usuarios finales.

Cuales son los clientes que la usan ahora?

En la mayoría de las situaciones en las que se vender el Powerhouse de la VAX el cliente está usando un lenguaje de tercera generación tal como el COBOL, EL BASIC O EL DIBOL. Si ellos ya están usando un lenguaje de cuarta generación para su centro de información usarán Powerhouse para su centro de aplicaciones.

3.4.10.8. TAMANO DE LA BASE DE DATOS QUE SOPORTA.

Cualquier tamaño de archivo soportado por RMS esta disponible para el Powerhouse y no tiene límite el número de archivos usado en un sistema de aplicación.

Más de 31 archivos pueden ser declarados en cualquier pantalla simple en línea, reporte o proceso.

3.4.10.9. VOLUMEN DE TRANSACCIONES.

El Powerhouse está diseñado para un volumen bajo o medio de

transacciones en el procesamiento de aplicaciones.

3.4.10.10. NUMERO DE USUARIOS.

El número de usuarios es independiente de la complejidad de la aplicación y la configuración del sistema.

3.4.10.11. MEDIO AMBIENTE FISICO.

En computadora grande no existe, en la familia mini-vax, desde micro vax I hasta vax 8800 corriendo en vax / vms. En hewlett-packard HP 3000, desde las series 37 a las series 70, en data general, desde MV 2000 hasta MV 20000. En micros - no existe.

3.4.10.12. BASE INSTALADA ACTUAL.

Hay más de 4000 licencias para instalación de Powerhouse con más de 400 copias en los sistemas VAX de clientes. Digital también tiene una licencia corporativa para el uso de Powerhouse en sus sistemas de información internos. Los servicios de software en Canadá tienen Powerhouse corriendo en 17 centros de datos de servicios por computadora.

3.5. OMNIS 3

3.5.1. DEFINICION.

Es un lenguaje de cuarta generaci3n dise1ado para correrse exclusivamente en m1quinas MACINTOSH.

Este lenguaje hace uso muy intenso de menus para la estructuraci3n de la base de datos, el cual es administrado por el System Finder del sistema operativo.

En la mayoria de los casos, el usuario s3lo tiene que ir seleccionando las alternativas que le parezcan m1s propicias para armar las especificaciones de su base de datos (producida por Blyth Software).

3.5.2. CARACTERISTICAS GENERALES DEL OMNIS 3.

Ofrece muchas facilidades para la r1pida consulta de cualquier tipo de informaci3n que usted requiera.

Ofrece privacidad y seguridad a la informaci3n a trav3s de "passwords".

Mantiene un control centralizado de su informaci3n.

Permite reducir la redundancia de la informaci3n.

Facilita el desarrollo de las aplicaciones.

Permite dar una respuesta r1pida a los requerimientos no planeados de informaci3n.

Permite tener y manejar r1pidamente archivos de un tama1o que no sobrepase los 64 millones de caracteres.

3.5.3. ESPECIFICACIONES TECNICAS.

3.5.3.1. ARCHIVOS DE BIBLIOTECA.

- N1mero de bytes 2 megabytes m1ximo.

- Número de formato 240 máximo.
- Número de formato
para un tipo 60 máximo.
- Tamaño de un formato 100 kbytes máximo.

3.5.3.2. ARCHIVOS.

- Número de archivos 60 máximo por base
de datos.
- Número de Archivos 12 máximo, simultaneamente
- Número de índices 12 máximo por archivo
- Número de Campos 120 máximo por archivo
- Número de caracteres 25000 máximo para archivos
abiertos y 60 por campo.

- Significancia 11 lugares decimales
- Decimales 6 máximo.

3.5.3.3. PANTALLAS DE ENTRADA.

- Número de pantallas 12 máximo
de entrada.
- Número de campos 120 máximo por pantalla.
- Número de secuencias 60 máximo.
- Número de comandos 60 por secuencia.

3.5.3.4. REPORTE.

- Número de renglones 240 máximo por layout
- Número de columnas 240 máximo por layout
- Número de campos 9600 máximo por reporte,
40 máximo por renglón
de layout.
- Número de totales 60 máximo por reporte.

3.5.3.5. MENUS.

- Número de Líneas de menú 18 máximo por menú
- Número de comandos 9 máximo por opción de menú

3.5.3.6. FORMATOS DE BUSQUEDA.

- Número de líneas 50 comparaciones por
búsqueda.

3.5.3.7. ARCHIVOS DE DATOS.

- Número de bytes 64 megabytes máximo.
- Número de registros límite del tamaño
del archivo.
- Número de archivos 24 archivos máximo.

3.5.4. BASES DE DATOS.

Las bases de datos que utiliza tienen excelentes funciones relacionares y jerárquicas.

Su estructura está definida por sus menús y una buena manera

de conocer sus capacidades es la de recorrer cada opción de dichos menús, haciendo incapié solamente en los aspectos más relevantes de la definición de archivos, reportes, pantallas de entrada, secuencias y menús.

La base de datos puede ser hasta de 64 MB de información y no hay límite para el número de registros en el archivo.

Existe la versión multiusuario que permite hasta 64 usuarios trabajando simultáneamente.

Al entrar al OMNIS 3 se nos presenta la primera pantalla, la cual muestra las alternativas de:

FILE
EDIT
OPTION

Los comandos de FILE permiten definir:

- Las bibliotecas que se usarán durante la sección.
- La configuración de la impresora.
- El cambio de fecha.
- El cambio de password para los diferentes niveles de los usuarios.
- Para cerrar ventanas.
- Para terminar la sesión.
- Así como para determinar si el menú OPTION debe estar visible o no para el usuario.

También existe un menú opcional para determinar el tamaño de la pantalla existente, que es operacional si se usa en el modelo XL de MACINTOSH.

Los comandos EDIT son muy parecidos a todos los de los programas que se usan en la MAC:

Permite CORTAR
PEGAR
COPIAR

ELIMINAR LO SELECCIONADO.

Tambièn existe el comando SELECT ALL que permite seleccionartoda la informaciòn de la ventana activa o sòlo la parte que serÀ eliminada.

3.5.5. El MENU OPTION

Es el que hace diferente al OMNIS 3 de otras aplicaciones.

Consta de las siguientes opciones:

- "ENTRY LAYOUT" Salidas a la pantalla para la entrada de datos.
- "AMEND LAYOUT" Salidas a la pantalla para hacer modificaciones.
- "SECUENCES" Secuencias.
- "ENTER DATA" Datos de entrada.
- "REPORTS" Reportes.
- "AMEND REPORT" Reporte que se puede modificar.
- "PRINT REPORT" Reporte a imprimir.
- "FILE FORMATS" Formatos de archivo.
- "SERCH FORMAT" Formatos de búsqueda.
- "MENUS" Menùs.
- "CHANGES DATA FILES" Archivos en los que se pueden cambiar los datos

3.5.6. PASOS PARA CREAR UNA APLICACION EN OMNIS 3.

PASO 1 Definir los formatos de archivos.

PASO 2 Diseñar las pantallas de captura.

- PASO 3 Definir los comandos y las secuencias de la aplicación.
- PASO 4 Capturar la información.
- PASO 5 Definir los formatos de los reportes.
- PASO 6 Crear los menus.
- PASO 7 Crear los passwords para la protección de la información.

Trataremos de explicar estas opciones en la forma lógica en que un usuario las seguirá:

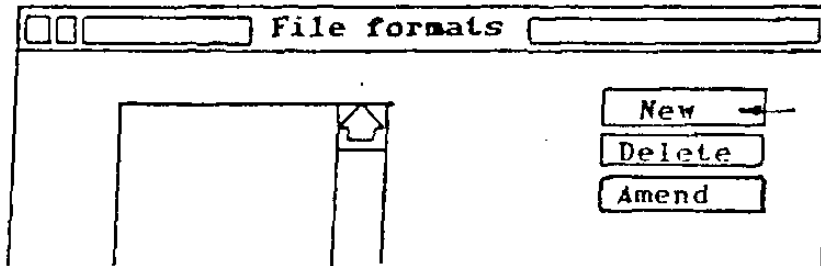
3.5.6.1 DISEÑO DE ARCHIVOS.

- a) Primero para definir los archivos se escoge la opción SHOW OPTIONS del menú File.

File Edit	Options
	Entry layout Amend Layout Sequences Enter Data
	Reports Amend Reports Print Reports
	File Formats ←
	Search Menus
	Change data files
	List files names

- b) Seleccione la opción "FILE FORMATS" del menú options.

Al teclear la opción, se define el nombre del archivo que se desea crear "NEW", modificar "AMEND", o borrar "DELETE".



Cuando ya se define el nombre del archivo aparece una ventana en la que se muestra información básica:

PARA CADA CAMPO.

TIPO DE CAMPO.

LONGITUD.

NUMERO DE LUGARES DECIMALES que tendrá el campo.

Y SI EL CAMPO ES INDEXADO O NO.

- "FIELD NAME" Nombre del archivo
- "TYPE" Tipo
- "LENGTH" Longitud
- "DECIMAL PALACES" Lugares decimales
- "INDEXED ?" Indexado?

Por ejemplo al archivo que definimos lo llamamos ACLIE presione <TAB>. Aparecerá el cursor en el recuadro de FIELD NAME Realice los siguientes pasos:

- 1.- En FIELDNAME se define el nombre del primer campo que es CNUM. Capture el nombre y presione <TAB>.

- 2.- Dentro del recuadro de FIELD TYPE hay siete opciones diferentes. Seleccione la opción NUMERIC con un clic y presione <TAB>.
- 3.- La longitud de un campo numérico está fija a 11 y puede tener hasta seis decimales; así que automáticamente se descarta esta opción.
- 4.- Como CNUM es numérico hay que especificar el número de decimales con las que desee trabajar. Teclee cero y presione <TAB>.
- 5.- Especifique el campo CNUM como indexado seleccione la pequeña caja que se encuentra del lado izquierdo de indexed.
- 6.- En INDEX LENGTH solamente presione <TAB>, ya que OMNIS 3 tomará la longitud supuesta por él.

Para indicarle a OMNIS 3 que estamos listos para especificar el siguiente campo presione <ESPACIO>, o seleccione con un clic el siguiente renglón en la sección de resumen.

Presione <TAB> para viajar a la sección de especificación y realice los pasos siguientes.

- 1.- Teclee CNOMBRE y presione <TAB>
- 2.- Seleccione NATIONAL y presione <TAB>
- 3.- Teclee 30 para la longitud del campo y presione <TAB>
- 4.- Seleccione el campo como indexado y presione <TAB>
- 5.- Presione <TAB> para que la longitud del indexamiento sea supuesta por OMNIS 3 a 30 caracteres.
- 6.- Presione <ESPACIO> para continuar con el resto de los campos.

Por medio de la opción "SET CONNECTIONS" del menú "AMEND FILE FORMAT" de "FILE FORMATS";

OMNIS 3 permite relacionar tablas entre si.

Estas relaciones pueden ser de 1 a N.

Existen también las opciones:

- "REORGANIZE DATA"
- "ESTIMATE DISK USAGE"
- "PRINT FILE DETAILS"

Que permiten cambiar la configuración de los archivos si se quieren agregar nuevos campos o hacer nuevas conexiones entre las tablas.

También se puede estimar la cantidad de espacio que se ocupará en disco de acuerdo con las características de la tabla definida y con el número de registros específicos.

Así como la impresión de las características definidas para los campos de la tabla activa.

3.5.6.2. DISEÑO DE PANTALLAS.

Ya que se definen los campos en los que se va a almacenar la información, se debe proceder al diseño de pantallas de captura para dicha información.

OMNIS 3 permite la actualización simultánea a diversas tablas.

- "ENTRY LAYOUTS" Salidas a la pantalla para la entrada de datos.

Si se vuelve al menú principal OPTIONS y se selecciona "ENTRY LAYOUTS" se nos muestra una pantalla en la cual se puede:

- CREAR
- CAMBIAR DE NOMBRE
- BORRAR

- MODIFICAR

- METER INFORMACION A

O DEFINIR SECUENCIAS DE UNA PANTALLA EN PARTICULAR.

Al dar el nombre, se pide la creación de una nueva pantalla y aparece una ventana.

Con el cursor se puede definir la información que se va a imprimir como letrero o prompt en dicha pantalla, si son necesarias más de una pantalla de entrada, se puede tener acceso a diferentes pantallas de entrada con la ayuda del scroll bar.

El formato de la pantalla se maneja como texto y se puede modificar fácilmente cortándolo y pegándolo en diversas partes de la pantalla.

Por lo tanto la pantalla encargada de capturar los datos generales del cliente tendrá la siguiente forma.

Número:

Nombre:

Dirección:

Teléfono:

Límite de Crédito:

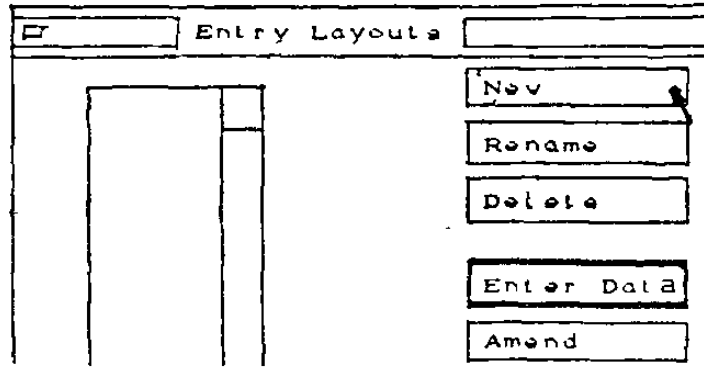
Saldo:

Para diseñar la pantalla anterior realice los pasos siguientes:

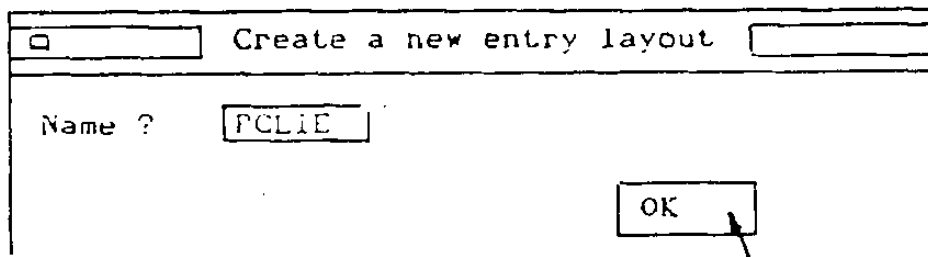
1.- Seleccione ENTRY LAYOUTS del menú OPTIONS.

Options
Entry Layouts
Amend Layout
Sequences
Enter Data
Reports
Amend Reports
Print Reports

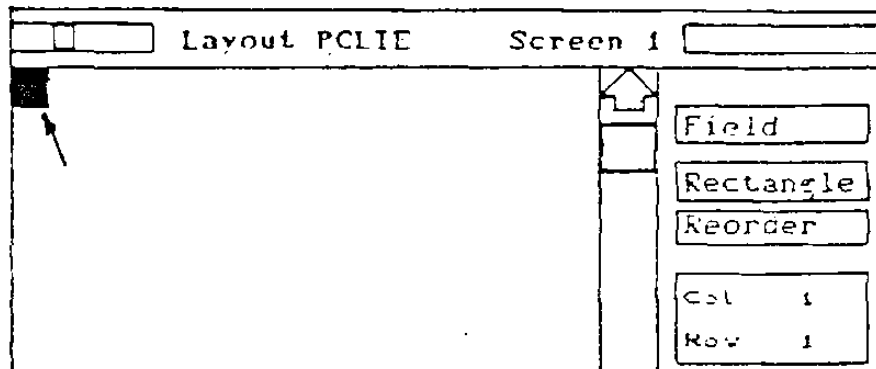
- 2.- Cuando aparezca la ventana de ENTRY LAYOUTS seleccione el botón NEW.



- 3.- Teclee PCLIE y seleccione el botón OK.



- 4.- Enseguida aparecerá la ventana de la pantalla de PCLIE



Con las teclas TAB y BACKSPACE o con los comandos de edición de la MACINTOSH, se pueden ajustar los letreros a la columna y al renglón que se desee.

En el recuadro inferior derecho aparece la posición del cursor por renglón y columna.

3.5.6.3. DEFINICION DE CAMPOS.

Los campos de la pantalla se definen colocando el cursor en el lugar donde se desea que empiece y oprimiendo el botón de "command field" (o bien "doble-clickeando" en el lugar en cuestion).

Esto provoca que se abra una ventana en la que se especifica el campo que se capturará en dicha posición.

Existe una opción en el menú "DESIGN & AMEND LAYOUT" que muestra una ventana a través de la cual se puede clickear el nombre directamente sin teclearlo.

Así como seleccionar campos de diversas tablas, seleccionando simplemente la tabla que se desea usar.

La caja del diálogo de definición de los campos trabaja simplemente para el LAYOUT en que se define y supersede aunque sin cambiar a la definición del campo hecha anteriormente.

El objetivo de esta selección es indicar ciertas preferencias útiles, como son el que se muestre un campo nada más con mayúsculas. Si el número 0 aparece como vacío si se desean números negativos, inserción de blancos, justificación de números a la derecha o a la izquierda o simultáneamente, pero sólo para ese reporte.

También se pueden validar los datos de entrada o usar información por "DEFAULT".

Cada nombre de campo contendrá la longitud específica tanto en la definición de la pantalla como en la definición del campo.

3.5.6.4. DEFINICION DE COMANDOS.

Los comandos y botones que se utilizaran en la captura de los datos generales del cliente por medio de la pantalla de PCLIEson comandos que ya están definidos por OMNIS 3.

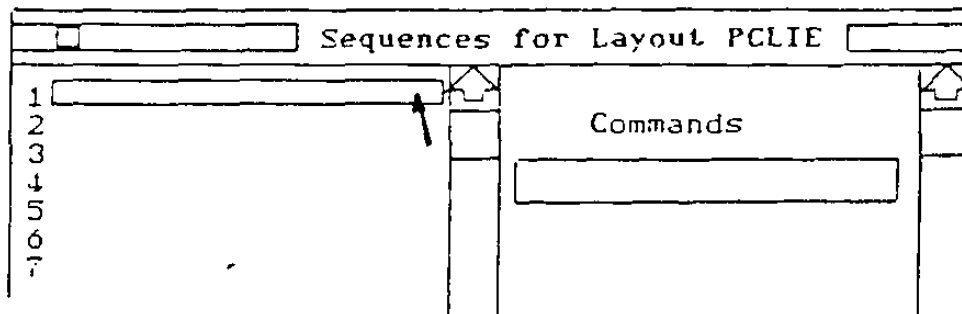
OMNIS 3 proporciona la facilidad de definir sus propios comandos y botones para una pantalla de captura. Con estos comandos se puede indicar la manera precisa en que se desea capturar y manipular la información para ser almacenada en los archivos

Efectue los siguientes pasos para definir algunos comandos para la pantalla de captura PCLIE.

- 1.- Seleccione ENTRY LAYOUTS del menú OPTIONS.
- 2.- Cuando la ventana de ENTRY LAYOUTS aparezca, seleccione la pantalla de PCLIE y oprima el botón SEQUENCES con un click.
- 3.- La ventana de SEQUENCES (comandos) aparecerá.

Esta consta de dos partes:

- a).- El área de comandos y
 - b).- El área de instrucciones.
- 4.- El menú de instrucciones que vamos a definir consta de las siguientes instrucciones:
- Próximo (desplegará el próximo registro del archivo).
 - Anterior (desplegará el registro anterior del archivo).
 - Encontrar (encontrará un registro determinado).
 - Borrar (borrará un registro del archivo).
 - Insertar (insertará un registro en el archivo).
- 5.- Dé un clic en la línea de comandos de la ventana de comandos.



- 6.- Teclee COMANDOS. Este será el nombre que llevará el

menú el cual contendrá las instrucciones antes mencionadas.

- 7.- Dé un clic en la segunda línea y teclee Próximo.
- 8.- Seleccione la opción Find/Next/etc de la ventana de instrucciones y después oprima el botón de ANOTHER COMMAND.

Para definir los demás comandos se siguen más o menos los mismos pasos.

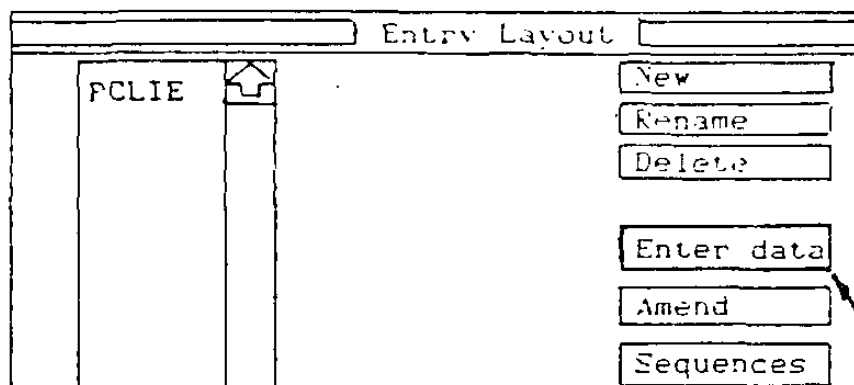
3.5.6.5. CAPTURA DE INFORMACION

Hasta este momento se han definido los formatos de los archivos, diseñado las pantallas de captura y las secuencias de comandos.

El siguiente paso es capturar la información.

Para capturar los datos generales del cliente realicemos los siguientes pasos:

- 1.- Seleccione ENTRY LAYOUTS del MENU OPTIONS.
- 2.- Cuando aparezca la ventana de ENTRY LAYOUT seleccione la pantalla de PCLIE y oprima el botón de ENTER DATA con un clic.



- 3.- La pantalla de PCLIE aparecerá. A la derecha de ella encontraremos los botones de comandos que pueden ser utilizados en esta pantalla. También aparecerá un nuevo

menú ENTER DATA el cual nos ayuda en la captura de información.

Enter Data
Next Previous Find Search Insert Edit Delete Print Record
Go to screen Main file Select report Select Search
Exact match find Clear before insert

Para capturar el primer registro seleccione el botón de INSERT y realice lo siguiente:

Teclee en cada uno de los espacios la información correspondiente seguida de <TAB>.

Si los datos que se han tecleado se encuentran correctos seleccione el botón de OK, si no ubique el apuntador en la información que desea modificar presionando la tecla <TAB> y vuelva a escribir la información.

3.5.6.6 SEGURIDAD

MACINTOSH permite una poderosa protección por medio de claves y capacidad de multiusuario.

Para asignar los passwords se oprime:

Primero el botón de "MODIFY PASSWORDS".

Se oprime la tecla TAB para teclear el password para el MASTER.

Luego se teclean los "passwords" para los diferentes usuarios y se especifican los niveles de acceso para cada uno de ellos.

Se entra por el menú FILE y se selecciona la opción

CHANGE USER PASSWORDS

Aparece las siguiente ventana.

```
-----  
| USER PASSWORD_____  MODIFY PASSWORD |  
| CURRENT USER IS MASTER.  OK           |  
|-----|
```

En la cual se cambia el password.

El usuario MASTER es el único que puede cambiar los passwords porque sólo el tiene acceso a todas las partes y funciones de OMNIS 3.

3.5.6.7. REPORTEES.

Si se usa la opción REPORTS para diseñar el reporte se nos presentará una pantalla como la que se muestra en la siguiente página, en la cual debemos colocar los campos, en la forma que deseamos que aparezcan en el reporte.

El reporte debe constar además de las siguientes cuatro secciones: una cabeza, un encabezado de detalle, detalle y una sección de totales.

La sección de cabeza contiene la información de encabezado para cada página del reporte y puede contener la fecha, paginación, etc.

Las secciones de detalle contiene la información en si que se desea y puede contener subtotales ya que su formato es muy flexible y se puede determinar el espacio entre cada parte del reporte o entre cada detalle distinto.

Se pueden hacer SORTS para diversos campos antes de prepararlos para el reporte o hacer cálculos que no afecten

a la base de datos y que sólo aparecerán en el reporte o hacer simulaciones de procesador de palabras que a su vez tomen información de la base de datos.

En fin se puede modificar totalmente la estructura del reporte incluyendo la cantidad de líneas permitidas por página, el espacio en los márgenes, la cantidad de columnas, etc.

Todo lo que acabamos de mencionar es confeccionado a través de menús presentados en la pantalla por el sistema.

3.5.6.8. MENUS.

Para crear los menús, se puede activar el comando menús a través de la opción "OPTION".

El comando menú activa opciones como abrir archivos, meter datos, imprimir reportes, usar criterios de búsqueda para encontrar registros o instalar otro menú.

Cada una de estas opciones activa una caja de diálogo donde se recaba la información del archivo que se desea abrir, el criterio de búsqueda, etc.

3.5.7. CONFIGURACION MINIMA REQUERIDA.

- MACINTOSH con 512 k mínimo.
- Unidad con disco adicional.
- Disco duro altamente recomendado.
- Impresoras con las que puede trabajar: Imagewriter, Laserwriter.

4. METODOLOGIAS PARA MEDIR LA PRODUCTIVIDAD DE SOFTWARE.

Los lenguajes de cuarta generación pretenden lograr un mayor incremento en la productividad de las empresas cumpliendo así con el principal objetivo para lo que fueron creados.

Para medir esta productividad se debe contar con alguna metodología, el objetivo de éste tema es tratarlo someramente para que el lector pueda tomarlo de base en investigaciones o aplicaciones futuras.

Es muy difícil aplicar modelos de aceptación general; porque la medición está ligada a la cultura o mentalidad de quien la utiliza o en el mejor de los casos del departamento de procesamiento de datos.

4.1 CARACTERISTICAS PRINCIPALES

Las metodologías para la medición de la productividad deben contar con dos características principales:

- a) Tener propiedades estadísticas para que el que los va a emplear pueda calcular e interpretar los resultados.
- b) Que además de proveer la información necesaria no sea costosa.

Las metodologías según en lo que se basan pueden ser:

- 1) Análisis estático de texto del programa.
- 2) Datos generados por el monitoreo de la ejecución del programa.
- 3) Desempeño humano.

4.2 METODOLOGIAS

4.2.1. LINEAS DE CODIGO.

Consiste en medir el número de líneas de código producidas. El valor varía dependiendo del tipo de líneas, pueden dejarse sin valor algunas, como los comentarios, o se les puede dar alguno.

También varía el valor dependiendo si se toma en cuenta o no la lógica del programador.

Como ventajas de este método podemos decir que es muy simple.

Como desventajas podemos decir que no es aplicable o no puede haber comparación entre los diferentes lenguajes de programación.

4.2.2. METODO DE HALSTEAD.

Predecir la longitud de un programa es su objetivo.

Usa cuatro variables básicas que pueden ser obtenidas de cualquier listado de un algoritmo en cualquier lenguaje.

- 1) n_1 = Representa el número de los diferentes operadores que aparecen en el programa.
- 2) n_2 = Representa el número de los diferentes operandos que aparecen en el programa.
- 3) N_1 = Representa la cuenta acumulativa del uso total de los operadores en el programa.
- 4) N_2 = Representa la cuenta acumulativa del uso total de los operandos en el programa.

Definiendo como operandos las variables o constantes y como operadores; los símbolos aritméticos, delimitadores, nombres de función, nombres de programa y estatutos "GOTO".

Las variables se combinan de la manera siguientes:

$$n = n_1 + n_2 \implies \text{vocabulario}$$

$N = N_1 + N_2 \implies$ longitud

$N =$ Es el número de BITS necesarios para representar todas las cosas que existen dentro de un programa.

$$N = n_1 \log_2(n_1) + n_2 \log_2(n_2)$$

$$N = N$$

$V =$ El volúmen de comparaciones mentales para hacer un programa.

$$V = N \log_2(n)$$

$L =$ Es el nivel del programa, es el recíproco de la dificultad, que a su vez es el número de discriminaciones básicas para completar una comparación.

$$L = \left[\frac{2}{n_1} \right] \left[\frac{n_2}{N} \right]$$

$E =$ Es el esfuerzo requerido por el programa:

$$E = V / L = N \log_2 n / \left[\frac{2}{n_1} \right] \left[\frac{n_2}{N} \right]$$

$$E = n_1 N_2 (N_1 + N_2) \log_2(n_1 + n_2) / 2 n_2$$

Donde E puede ser definido como el número de discriminaciones mentales requeridas para generar un programa.

$T =$ Es tiempo requerido para desarrollar un programa; lo podemos determinar así:

$S:$ Es el tiempo requerido para el cerebro humano, para ejecutar una discriminación mental.

Según estudios: $S = 20$ segundos

Por lo tanto $T = E / S$

Halstead estimó $S = 18$ seg., pero éste puede variar de un ambiente a otro.

4.2.3. PUNTOS DE FUNCION

El objetivo del método de puntos de función es contar con una medida relativa del valor de la función o aplicación que se entrega al usuario, independientemente de la tecnología o enfoque aplicado.

Esta metodología permite medir la productividad de una herramienta de Software independientemente de la máquina en la que se vaya a implementar y del modo de operación que vaya a tener.

Esta metodología parte del hecho de que el valor de la aplicación es proporcional en forma consistente al conteo ponderado del número de entradas, salidas, consultas y archivos maestros que maneja la aplicación debido a que estas operaciones son las que manejan y reflejan el trabajo externo de la misma.

El primer paso que hay que dar en el uso de ésta metodología consiste en clasificar en tres niveles de complejidad (facil, promedio o difícil) todas las salidas y entradas externas, los archivos maestros lógicos, así como los enlaces con otras aplicaciones y las consultas que la aplicación permite al usuario, después se cuentan las entradas y salidas para dichas funciones, tomando en cuenta las aplicaciones con las que esté ligada o con versiones anteriores de la misma aplicación que se esté midiendo.

4.2.3.1. CRITERIOS PARA EVALUAR LAS ENTRADAS EXTERNAS.

Las entradas externas del usuario deben ser contadas independientemente del medio que se utilice para su uso en la aplicación.

Si las entradas son archivos, serán considerados como tales, sólo las que la aplicación lea pero no actualice.

Los criterios que se siguen para determinar la complejidad pueden ser los siguientes:

COMPLEJIDAD BAJA: Entrada con poca diversidad de elementos de datos o consideraciones que los humanos tengan que hacer en el momento de la captura de la información.

COMPLEJIDAD MEDIA: Se clasifica la entrada como de esta complejidad cuando no cae dentro de la baja, ni de la alta.

COMPLEJIDAD ALTA: Entrada con muchos tipos de elementos de datos o consideraciones de los humanos en la entrada.

Las entradas deben ser consideradas por la función que cumplen aunque en un mismo Display sirva para altas, bajas o modificaciones, se deben contar por separado.

Las entradas necesarias debido al sistema operativo que se este usando no serán consideradas como tales ya que se pretende evaluar el desarrollo del programador.

Las consultas no se considerarán como entradas estas serán consideradas con criterios aparte.

La siguiente tabla es un ejemplo de como deben contarse las entradas de una aplicación:

DESCRIPCION	NUMERO DE TIPOS DE ENTRADA
- Pantalla de entrada de datos.	1
- Pantalla de entrada con multiples funciones.	1 por funcion
- Pantalla de entrada de funciones.	1
- Consulta seguida por una actualizacion.	1
- Entrada de respaldo con la misma lógica de procesamiento que la entrada primaria.	1
- Dos pantallas con el mismo formato y diferente lógica de procesamiento.	2
- Tablas o archivo mantenido por el	1 por lo menos

usuario.

- Forma de entrada por OCR

1

Esta clasificación es de acuerdo a la guía de complejidad descrita anteriormente. El peso de cada función varía de acuerdo a la complejidad y a factores tales como movimientos complejos de cursor, conversión de datos brutos, etc.

4.2.3.2. CRITERIOS PARA EVALUAR LAS SALIDAS EXTERNAS DEL USUARIO.

Las salidas que proporcionen información al usuario para la toma de decisiones o cursos de acción deben ser consideradas en forma individual.

Por ejemplo; los "displays" por terminal, los mensajes del operador, las transacciones que alimentan otra aplicación, etc.

Las salidas pueden ser clasificadas de acuerdo al criterio que se utilizó para las entradas en cuanto a complejidad.

Los reportes impresos que sean generados, se pueden clasificar de acuerdo con los criterios siguientes:

COMPLEJIDAD BAJA: Un reporte impreso con una o dos columnas que tenga una elaboración relativamente fácil (un sort sobre una sola llave) puede ser considerado como un reporte de baja complejidad.

COMPLEJIDAD MEDIA: Si el reporte tiene diversas columnas con cierta elaboración como subtotales y formateo gráfico (como el formato de un balance contable), se podría considerar de complejidad media.

COMPLEJIDAD ALTA: Si se hacen elaboraciones considerables correlaciones diversas entre datos, el listado se puede considerar con alta complejidad las elaboraciones pueden ser sorts alfabéticos de apellidos paternos, maternos y nombre dentro de una o más llaves adicionales.

Las salidas deben ser consideradas bajo los mismos criterios

de las entradas; si hay entradas que proporcionen datos distintos deben ser consideradas como salidas distintas.

Los reportes que contengan los mismos datos pero en diversos formatos no deben ser considerados como reportes distintos.

También se contarán como salidas, los listados de error o reportes de procesamiento de trabajo; que formen parte de la ejecución.

No se deben tomar en cuenta los mensajes de error que aparezcan en la pantalla que no requieran de algún procesamiento lógico.

Las salidas ocasionadas por la mera implementación del sistema no deberán ser contadas, ni las consultas deben ser contabilizadas como salidas, porque serán consideradas posteriormente.

Las salidas pueden contarse de la siguiente manera:

DESCRIPCION	NUMEROS DE TIPOS DE SALIDA
- Pantalla de salida de datos.	1
- Datos para transacciones de otras aplicaciones.	1
- Mensaje del operador desde la aplicación.	1
- Reporte impreso por aplicación en BATCH.	1
- Reporte de ejecución en BATCH.	1
- Tabla o archivo mantenido por el usuario.	1 por lo menos
- Pantalla de salida de inicio.	1
- Pantalla de salida de final.	1

4.2.3.3. CRITERIOS PARA EL CONTEO REFERENTE A LOS ARCHIVOS DEL USUARIO.

Los archivos maestros son los que son mantenidos por la aplicación dando de altas, bajas, modificaciones o para producir salidas de valor al usuario.

Cada grupo de datos deberá ser contado, incluyendo cada archivo lógico que es accesado por el computador. Si se utiliza tecnología de base de datos cada vista utilizada por el usuario que se genere, se use o se mantenga a través de la aplicación.

La complejidad de los archivos maestros es hecha de la siguiente manera:

COMPLEJIDAD BAJA: Cuando la recuperación de la información es sencilla y el tipo de datos de recuperar también.

COMPLEJIDAD MEDIA: Por exclusión.

COMPLEJIDAD ALTA: Si el tipo de los datos es complejo y/o existe un número de tipos grande, de tal manera que la recuperación de la información tiene consideraciones importantes.

El archivo maestro debe ser contado como archivo lógico independientemente del archivo físico.

Como evaluación podríamos tomar la siguiente:

DESCRIPCION	NUMEROS DE TIPOS DE ARCHIVO
- Cada entidad lógica desde el punto de vista del usuario.	1
- Archivos internos lógicos generados o mantenidos por la aplicación.	1
- Archivo utilizado para datos o control para aplicaciones BATCH consecutivas.	1
- Archivo de trabajo de clasificación.	0

4.2.3.4. CRITERIOS PARA EL CONTEO DE INTERFASES CON OTRAS APLICACIONES.

Las interfases son archivos creados por la aplicación que se está midiendo y que son pasados a otra aplicación o viceversa. Pueden ser considerados de este tipo los que son de referencia para la ejecución del proceso, no los que se requieren para la ejecución del programa.

Los archivos de interfase no deben ser considerados como salidas. Si existen archivos comunes a varias aplicaciones, estos deben ser considerados como archivo o como interfase, no como ambos.

Ejemplos de criterios para la evaluación de interfases señalan las siguientes:

DESCRIPCION	NUMERO DE TIPOS DE ARCHIVO
- Archivo de registros obtenido por otra aplicación.	1
- Archivos de registros enviados a otra aplicación.	1
- Archivo de registro a muchas otras aplicaciones.	1
- Bases de datos compartida con otra aplicación.	1

4.2.3.5. CRITERIOS PARA EL CONTEO DE CONSULTAS.

Las consultas son las que tienen entradas y salidas en línea en las cuales no se hagan actualizaciones.

Si los datos de la consulta se usan posteriormente para actualizar se deben contar como consulta y salida.

Si una aplicación emite el manejo de actualizaciones sucesivas tipo cola, se deben considerar como actualizaciones ya que constituyen de hecho entradas y salidas, que es donde deberán de ser contadas.

La complejidad de las consultas pueden ser clasificadas de la siguiente manera:

COMPLEJIDAD BAJA: Cuando la consulta tiene pocos tipos de registros, baja dificultad de captura o pocos tipos de datos lógicos.

COMPLEJIDAD MEDIA: A esta clasificación se llega por exclusión.

COMPLEJIDAD ALTA: Cuando la consulta tiene muchos tipos de registros, grupos de datos lógicos o dificultad de captura por el ser humano.

Una ejemplificación para el conteo de las consultas sería la siguiente:

DESCRIPCION	NUMERO DE TIPOS DE CONSULTAS
- Entrada y salida en línea sin actualización de datos.	1
- Entrada y salida de pantalla de ayuda.	1
- Entrada y salida de pantalla de menú.	1

4.2.3.6. FACTORES DE AJUSTE DE VALOR.

Para construir el índice de Puntos de Función, es necesario dar un valor a ciertos factores presentes que afectan el valor de la aplicación según la percibe el usuario. Para los factores enlistados a continuación se debe de asignar un valor de acuerdo a la siguiente tabla:

0	No estuvo el factor presente / no hubo influencia
1	Influencia casual / circunstancial
2	Influencia pequeña
3	Influencia moderada
4	Influencia importante

5 Influencia fuerte generalizada

- a. Los datos utilizados en la aplicación se envían o reciben a través de las redes establecidas de comunicación. (No se debe de considerar a terminales conectadas en línea como redes de comunicación.)

- b. Se utiliza el concepto de procesamiento o base de datos distribuidas.

- c. La productividad de la aplicación en el tiempo de respuesta o en su totalidad son básicamente consideraciones del diseño / implementación/mantenimiento.

- d. Existe un compromiso alto por el usuario de usar el hardware disponible.

- e. Un alto volumen de transacciones tuvo influencia en la estructura del diseño, la implantación, y mantenimiento de la aplicación.

- f. La captura de datos en línea es proporcionada por el programa.

- g. La captura de datos en línea es conversacional y requiere que una transacción se construya con varias operaciones.

- h. La base de datos o los archivos maestros se actualizan en línea.

i. Existe un procesamiento en la aplicación, entendiéndose como complejo si existen muchos puntos de decisión, una gran cantidad de ecuaciones matemáticas o lógicas, o un número considerable de excepciones que ocasionan el tener muchas transacciones incompletas que deberán resolverse posteriormente.

j. La aplicación fue diseñada específicamente para ser implantada y mantenida en otros sitios o para otras aplicaciones. Los siguientes porcentajes aplicarían en el grado que esto es cierto.

RANGO	PORCENTAJE
0	0.0 a 10.0 %
1	10.0 a 20.0 %
2	20.0 a 30.0 %
3	30.0 a 40.0 %
4	40.0 a 50.0 %
5	50.0 y más %

k. La conversión e implantación fueron llevadas a cabo bajo condiciones controladas y de acuerdo a un plan estructurado que incluyó un período de pruebas.

l. La actividad operativa fue resuelta fácilmente desde el principio por la aplicación ya que ésta fue considerada en el diseño e implantación, por lo que el manejo manual de la información se reduce al mínimo.

m. La aplicación se diseñó primordialmente para ser instalada en sitios múltiples de múltiples organizaciones.

n. El mantenimiento de la aplicación es fácil.

TOTAL DE GRADO DE INFLUENCIA

4.2.3.7. FORMATO PARA EL CALCULO DEL INDICE DE PUNTOS DE FUNCION.

CONCEPTO	NUMERO CONTADO
I. Entradas del usuario.	___ sencilla * 3 = ___ promedio * 4 = ___ compleja * 6 =
TOTAL DE ENTRADAS	___
II. Salidas del usuario	___ sencilla * 4 = ___ promedio * 5 = ___ compleja * 7 =
TOTAL DE SALIDAS	___
III. Archivos maestros lógicos	___ sencilla * 7 = ___ promedio * 10 = ___ compleja * 15 =
TOTAL DE ARCHIVOS	___
IV. Interfases	___ sencilla * 5 = ___ promedio * 7 = ___ compleja * 10 =
TOTAL DE INTERFASES	___
V. Consultas	___ sencilla * 3 = ___ promedio * 4 = ___ compleja * 6 =
TOTAL DE CONSULTAS	___

Total de puntos de función no ajustados ____

VI. Ajuste de valor

$0.65 + (0.01 * \text{total de grado}) = \text{ajuste de valor de influencia.}$

VII. Índice de punto de función

$\text{Total de puntos de función no ajustados} * \text{Ajuste de valor} = \text{Índice de punto de función}$

(V)

(VI)

Complementario a este índice, se pueden sacar diversos indicadores que relacionan el costo y horas de trabajo empleado en la elaboración de la aplicación por índice de punto de función. Sin embargo, el análisis será útil o no, dependiendo de la organización en la que se esté llevando a cabo.

Adicionalmente se debe de hacer notar que los factores constantes para la ponderación de las complejidades y para el ajuste de valor fueron obtenidos a base empírica, y que pueden variar de medio ambiente de trabajo a medio ambiente de trabajo.

4.2.3.8. GUIA PARA SELECCIONAR LOS FACTORES DE COMPLEJIDAD PARA LA ENTRADA.

	1 a 4 TED	5 a 15 TED	16 o m TED
0 a 1	S	S	P
2 TAR	S	P	C
3 o m TAR	P	C	C

FUNCIONES PARA LA ENTRADA.

- TED = TIPOS DE ELEMENTOS DE DATOS
- TAR = TIPOS DE ARCHIVOS REFERENCIADOS
- TR = TIPOS DE REGISTOS
- S = SENCILLA (baja o fácil)
- P = PROMEDIO (media)
- C = COMPLEJA (difícil)

4.2.3.9. GUIA PARA SELECCIONAR LOS FACTORES DE COMPLEJIDAD PARA LA SALIDA.

	1 a 4 TED	5 a 15 TED	16 o m TED
0 a 1	S	S	P
2 TAR	S	P	C
3 o m TAR	P	C	C

PARA LA SALIDA

- TED = TIPOS DE ELEMENTOS DE DATOS
- TAR = TIPOS DE ARCHIVOS REFERENCIADOS
- TR = TIPOS DE REGISTROS
- S = SENCILLA (baja o fácil)
- P = PROMEDIO (media)
- C = COMPLEJA (difícil)

4.2.3.10. GUIA PARA SELECCIONAR LOS GRADOS DE COMPLEJIDAD EN
LOS ARCHIVOS LOGICOS DEL USUARIO.

	1 a 4 TED	5 a 15 TED	16 o m TED
0 a 1	S	S	P
2 TAR	S	P	C
3 o m TAR	P	C	C

TED = TIPOS DE ELEMENTOS DE DATOS

TAR = TIPOS DE ARCHIVOS REFERENCIADOS

TR = TIPOS DE REGISTROS

S = SENCILLA (baja o fácil)

P = PROMEDIO (media)

C = COMPLEJA (difícil)

4.2.3.11. GUIA PARA LA SELECCION DE GRADOS DE COMPLEJIDAD EN LAS INTERFACES CON OTRAS APLICACIONES.

	1 a 4 TED	5 a 15 TED	16 o m TED
0 a 1 TAR	S	S	P
2 TAR	S	P	C
3 o m TAR	P	C	C

- TED = TIPOS DE ELEMENTOS DE DATOS
TAR = TIPOS DE ARCHIVOS REFERENCIADOS
TR = TIPOS DE REGISTROS
S = SENCILLA (baja o fácil)
P = PROMEDIO (media)
C = COMPLEJA (difícil)

4.2.3.12. GUIA PARA SELECCIONAR LOS FACTORES DE COMPLEJIDAD PARA LA CONSULTA.

	1 a 4	5 a 15	16 o m
	TED	TED	TED
0 a 1	S	S	P
2 TAR	S	P	C
3 o m TAR	P	C	C

TED = TIPOS DE ELEMENTOS DE DATOS

TAR = TIPOS DE ARCHIVOS REFERENCIADOS

TR = TIPOS DE REGISTRO

S = SENCILLA

P = PROMEDIO

C = COMPLEJA

5. PROVEEDORES DE DBMS Y LENGUAJES DE CUARTA GENERACION

Enseguida se enumeran los proveedores representativos de Sistemas Administrativos de Base de Datos y Lenguajes de Cuarta Generación. Parte de ésta lista fue tomada de la publicación Software News de Julio de 1984.

PRODUCTO/VENDEDOR	MEDIO AMBIENTE	TIPO
ADF & DMS IBM 1133 WESTCHESTER AVE WHITE PLAINS, NY 10604	IBM 370/4300/30XX DOS/VSE,MVS,CICS IMS/DC	4GL
ADS/O USA Cullient Software 400 Blue Hill Dr. Westwood, MA 02090	IBM 370/4300/30XX Dos, VM/CMS.MVS	QUERY, 4GL, IDMS DBMS
APS utiliza Sage Systems Inc. Painter 3200 Monroe St. Rockville, MD 20852	IBM 370/4300/30XX	4GL que DataBase DBMS, Screen Painter 4GL
ASI Inquiry Applications Software 21515 Hawthorne Blvd. Torrance, CA 90503	IBM 370/4300/30XX CICS/TSO/CMS	Query
dBase II Ashton-Tate 10150 W. Jefferson Culver City, CA 90280	CP/M-80,-86 MS-DOS, PC-DOS	DBMS, 4GL

CA-Universe Computer Associates 125 Jericho Tnpke Jericho, NY 11753	IBM 370/4300/30XX DOS/VSE, MVS, VM/CMS IBM-PC, PC-DOS	DBMS, Query 4GL
Condor Series 20 Condor Computer Corp. 2051 S. State St. Ann Arbor, MI 48104	8080, 8085, Z80 8086 CP/M, MP?M, MS-DOS	DBMS, 4GL Editor de Pantallas
Conquer Sydney Development Co 600-1385 W. 8th Ave. Vancouver, BC V6H3V9	IBM 370/4300/30XX MVS/TSO, VM/CMS	DBMS, 4GL Query
DataBase Tominy 4221 Malsbary Rd. Cincinnati, OH 45242	IBM 370/4300/43XX 34, 36, IBM-PC, PC-DOS	DBMS, 4GL Query
Day One Day One Software 618 Shoemaker King of Prussia, PA 19406	IBM-PC, Apple II, TRS, Kaypro, Televideo, Compaq	DBMS, 4GL Query
Dayflo Dayflo Software, Inc 2500 Michigan Dr. Irvine, CA 92715	IBM-PC & XT	DBMS, 4GL Query
DNA-4 Exact Systems & Progr 1 Labriola Court Armonk, NY 10504	Data Gen. MV & Nova Eclipse RDOS, AOS, RDOS, AOS M68000	DBMS, 4GL Query

Express Mgt. Decisions Systems 200 Fifth Ave. Waltham, MA 02254	IBM 370/4300/30XX Express E3000	DBMS Query
--	------------------------------------	---------------

Falcon Peregrine Systems 15530 Rockfield Blvd. Irvine, CA 92714	IBM 370/4300/30XX DEC VAX, Unix, IBM-PC	DBMS, 4GL Query
--	--	--------------------

Focus Information Builders 1250 Broadway New York, NY 10001	IBM 370/4300/30XX Dos, VM, MVS, IBM-PC, Wang, TI, PC-DOS, MS-DOS	DBMS, 4GL Query
--	--	--------------------

Ideal Applied Data Research Route 206 & Orchard Road Princeton, NJ 08540	IBM 370/4300/30XX DOS, MVS, VM/CMS	4GL
---	---------------------------------------	-----

Info Henco Software Inc. 100 Fifth Ave Waltham, MA 02154	IBM 370/4300/30XX VM/CMS, Prime, Dec VAX, Harring, Honeywell DP86	DBMS, 4GL Query
---	--	--------------------

Infocen 3CI 155 W. Harvard Fort Collins, Co 80525	DEC VAX, DG MV	DBMS, 4GL Query
--	----------------	--------------------

Informix 3.0, Ace Relational DataBase S 2471 E. Bayshore Rd., Ste. 600 Palo Alto, CA 94303	DEC VAX, PUP-11, IBM, Altos, MS-DOS Lisa Unix, PC-DOS	Informix DBMS usa query Informer, 4GL
--	---	---

Ingres Relational Technology 2855 Telegraph Ave. Berkeley, CA 94705	DEC VAX, VAX/VMS Unix, PDP-11	DBMS, usa Quel Query 4GL
--	----------------------------------	--------------------------------

Inquire Infodata Systems 5205 Lessburg PK Falls Church, VA 22041	IBM 370/4300/30XX DOS/VSE, MVS, OS/VS1	DBMS, 4GL
---	---	-----------

Intellect Artificial Intelligen Corp. 100 Fifth Avenu Waltham, MA 02254	IBM 370/4300/30XX DOS/VSE, MVS, VM/CMS	Query
---	---	-------

IP-3 Computing Productivity Rte. 1-433-A Waitsfield, VT 05673	IBM 370/4300/30XX MVS	4GL query genera IMS/VS DBMS
--	--------------------------	---------------------------------------

Linc Burroughs Co. One Burroughs Pl. Detroit, MI 48232	B1700-B1900 B2700-B4900 B6700-B7900	4GL
---	---	-----

Magnum Tymshare 20705 Valley Green Cupertino, CA 95014	DEC 10, 20, Tops-20 DEC VAX, VMS	DBMS, 4GL Query
---	-------------------------------------	--------------------

Mantis Cincom Systems 2300 Montana Ave. Cincinnati, OH 45211	IBM 370/4300/30XX DOS, MVS, VM/CMS	4GL, usa Total, VSAM DBMS
---	---------------------------------------	------------------------------------

Mapper Sperry Univac Box 500 Blue Bell; PA 19424	Toda la linea 1100	DBMS, 4GL Query
---	--------------------	--------------------

Mark V Informatics General 21050 Vanowen St Canoga Park, CA 91304	IBM 370/4300/30XX	4GL, usa Inquiry IV Query
--	-------------------	---------------------------------

Metafile Sensor Based Systems 15 E. Second St. Chatfield, MN 55923	IBM-PC y PC-DOS compatibles	DBMS, 4GL
---	--------------------------------	-----------

Natural Software AG of N.A. 11800 Sunrise Valley Reston, VA 22091	IBM 370/4300/30XX DOS, MVS, VM, CMS	Query, 4GL usa DBMS Adabas
--	--	----------------------------------

Nomad 2 D&B Computing Services 187 Danbury Rd. Wilton, CT 06897	IBM 370/4300/30XX VM/CMS	DBMS, 4GL Query
--	-----------------------------	--------------------

NPL Info. Mgt. System Desktop Software Corp. 228 Alexander St. Princeton, NJ 08540	IBM PC&XT, Vic 9000 Apple, Burroughs B20, Sage II, DEC 350 y Rainbow, HP9816	DBMS, 4GL
---	---	-----------

Omnis 3 Apple Computer Inc. 10260 Bandley Dr. Cupertino, Cal. 95014	MACINTOUCH	DBMS, 4GL
--	------------	-----------

Oracle Oracle Corp. 2710 Sand Hill Rd. Menlo Park, Ca 94205	IBM 370/4300/30XX DG MV, DEC VAX, PDP-11 Harris, Stratus, 6800 8086	DBMS, 4GL usa query SQL
--	--	-------------------------------

Pacbase CGI Systems, INC. 8200 Greensboro Dr. Ste. 1010 McLean, Va 22102	IBM 370/4300/30XX DOS/VS/VSE, CIVS VSAM, DL/1, IMS DB/DC, OS/VS/MVS	4GL, usa IMS/DB y DBMS de Codasyl
--	--	--

Pearlsoft Pearlsoft Division 3700 River Rd. N. Ste. 3 Salem, OR 97303	Z80a, 8080, 8085 CP/M 2.2	DBMS, 4GL
--	------------------------------	-----------

PowerHouse Cognos Systems Ltd. 275 Slater, St. Ottawa, Ont. Canada K1P 5H9	HP 3000, DEC VAX, DG MV	Un 4GL que Utiliza Quiz, QTP y Quick Query
--	----------------------------	--

Pro-IV Pro-IV 119 Russell St. Littleton, MA 01460	DEC VAX, PDP-11, RSX11M, RSTS/E, 8088/8086, 68000	DBMS, 4GL
--	---	-----------

Queo-IV Computer Techniques 1622 Main Ave. Olyphant, PA 18447	Prime 250-9950 Primeos	DBMS, 4GL
--	---------------------------	-----------

Rally
Digital Equipment
Corporation
Main Street
Maynard, Mass.

DEC VAX

DBMS, 4GL

Ramsis II
Mathematica
P.O. Box 2392
Princeton, NJ 08540

IBM 370/4300/30XX
DOS, VM/CMS, MVS

DBMS, 4GL
Query

Revelation
Cosmos
P.O. Box AH
Morton, WA 98356

PC & XT,
Eagle 1600

DBMS, 4GL

Salvo
Software
14333 Proton Rd
Dallas, TX 75238

CP/M CP/M-86,
MS-DOS

DBMS, 4GL

Speed II
The Office Manager
127 SW 156th St.
P.O. Box 66596
Seattle, WA 98166

Wang VS, Wang 2200

Query, 4GL

Umbrella
Hogan Systems, Inc.
5080 Spectrum Dr.
Dallas, Tx 75248

IBM 370/4300/30XX
DOS/VSE, MVS, OS/VSI

4GL

C O N C L U S I O N E S

De lo expuesto aquí podemos concluir que el SOFTWARE de computadoras está cada vez más a disposición y al alcance del usuario.

Su uso se ha generalizado y ha invadido nuestro mercado, sobre todo en el nivel de Micros.

A nivel Mini o Mainframe ha alcanzado gran importancia sobre todo en las compañías que tienen que cambiar los precios de los productos constantemente.

La selección del lenguaje adecuado depende de las necesidades particulares de las empresas, del conocimiento de los diferentes productos o lenguajes de los que pueden hacer uso; así como del equipo físico con el que se cuenta.

ANEXO

SISTEMA DE APLICACION DESARROLLADO EN UN LENGUAJE DE CUARTA GENERACION.

ESTE SISTEMA ES PARA CONTROLAR LA RENTA DE PELICULAS DE UN CENTRO DE SERVICIO AL PUBLICO.

EL LENGUAJE EMPLEADO ES EL RALLY.

EL SISTEMA CONSTA DE TRES RELACIONES O REGISTROS.

- 1.- CLIENTES
- 2.- HISTORIA
- 3.- PELICULAS

EN LA RELACION CLIENTES SE DEFINIERON LOS SIGUIENTES CAMPOS CON SUS CLAVES Y SU AMPLITUD.

DESCRIPCION DEL CAMPO	NOMBRE DEL CAMPO	TIPO	AMPLITUD
Número de Cliente	NUMCLI	Num	4
Nombre	NOM	TXT	30
Dirección	DIR	TXT	30
Teléfono	TEL	TXT	8
Fecha de alta de cliente	FECHALT	DATE	8

EN LA RELACION PELICULAS

DESCRIPCION DEL CAMPO	NOMBRE DEL CAMPO	TIPO	AMPLITUD
Número de Película	NUMPE	NUM	4
Nombre	NOM	TXT	30
Clasificación	CLASIF	TXT	1
Tipo	TIPO	TXT	10

EN LA RELACION HISTORIA

Número de Cliente	NUMCLI	NUM	4
Número de Película	NUMPE	NUM	4
Fecha de Renta	FECHREN	DATE	8
Fecha de Regreso de la Película	FECHDREG	DATE	8
Fecha real de Devolución	FECHDEV	DATE	8

PARA EMPEZAR A TRABAJAR EN RALLY SE TECLEA LA SIGUIENTE ORDEN:
RALLY CREATE RENTA

APARECE LA SIGUIENTE PANTALLA Y SE TECLEA EL NUMERO 1.

Main Menu

Rally Definition System

- 1 Databases
- 2 Builder tools
- 3 Edit application
- 4 Utilities
- 5 Miscelaneous
- 6 Run application

Your choice: 1 _____

GOLD-Q	'quit action'	Exit RALLY Definition System
HELP	'help'	For more help

DESPUES DE TECLEAR EL NUMERO 1 APARECERA EL MENU PARA LA CREACION DE LA BD.

Databases

- 1 Create a database
- 2 Edit a database
- 3 delete a database

Your choice: 1 _____

GOLD-Q	'quit action'	Exit menu without making a choice
HELP	'help'	for more help

UNA VEZ CREADA LA BD APARECE LA SIGUIENTE PANTALLA PARA DARLE NOMBRE.

Create a Database

Mode of integration: NOT USED _____

C D D/Plus path name: _____

R d b file name: RENTA _____

Use default parameters? Y__

ENTER	'finish action'	Create the database
BACKSPACE	'previous field'	Select a different database name to create
GOLD-Q	'quit action'	Return to previous menu without creating a database

LA SIGUIENTE PANTALLA ES UN MENU PARA LA CREACION DE:

- CAMPOS GLOBALES
- RELACIONES
- INDICES

Edit a Database

Object:	Operations:
1 Global field definition	1 Create
2 Relation	2 Edit
3 Index	3 Delete

GOLD-Q	Exit menu without making
HELP	For more help

Select an object and an operation: 1__1__

PARA CREAR UN CAMPO GLOBAL NUMERICO SE USA ESTA FORMA.

Create a Global Field definition

Data type: WORD SIGNED
Scale factor: 0
Missing Value: _____
Comment: _____

Type a comment about this global field definition.

Name of database: **RENTA**
Name of field being created: **NUM_4**

EL LETRERO **WORD SIGNED** EN EL CAMPO Data Type: SIGNIFICA HASTA 4
NUMEROS (1 BYTE).

LA FORMA SIGUIENTE ES PARA LA DEFINICION GLOBAL DE UN CAMPO DE FECHA.

Create a Global Field Definition

Data type: ABSOLUTE DATE TIME _____
Scale factor: 0 _____

Missing Value: _____

Comment:

Type the missing value for this field (optional).

Name of database: RENTA
Name of field being created: fecha

EN LA SIGUIENTE PANTALLA SE ENTRA PARA CREAR UNA RELACION

Edit a Database

Object:	Operations:
1 Global field definition	1 Create
2 Relation	2 Edit
3 Index	3 Delete

GOLD-2	Exit menu without making
--------	--------------------------

HELP	For more help
------	---------------

Select an object and an operation: 2_1_____

EN ESTA PANTALLA SE TECLEA EL NOMBRE DE LA RELACION.

Create a Relation

Relation name: CLIENTES

Copy from TEAMDATA table? N

ENTER	'finish action'	Commit this information
BACKSPACE	'previous field'	Move to previous field
GOLD-Q	'quit action'	Leave this forms without committing this information

Database being edited: **RENTA**

LA SIGUIENTE PANTALLA ES PARA DEFINIR LOS CAMPOS DE LA RELACION.

Edit the Fields in a Relation

Local field name	Computed	Global field name
NUMERO_CLIENTE	N	NUM_4
NOMBRE	N	TXT_30
DIRECCION	N	TXT_30
TELEFONO	N	TXT_8
FECHA_ALTA	N	FECHA

List of
global fields
DATE
FECHA
INTEGER
MONEY
NUM_4
REAL
TEXT
TXT_1
TXT_10
TXT_30
TXT_8

GOLD-SELECT	'list_of_values'	move to list
GOLD-Q	'QUIT ACTION'	of values
		Return to the
		previous menu

Database being edited: **RENTA**
Relation being edited: **clientes**

LA SIGUIENTE PANTALLA ES PARA CREAR LOS INDICES

Edit a Database

Object:	Operations:
1 Global field definition	1 Create
2 Relation	2 Edit
3 Index	3 Delete

GOLD-Q	Exit menu without making
HELP	For more help

Select an object and an operation: 3__1_____

EN LA SIGUIENTE PANTALLA SE DEFINE EL INDICE PARA LA RELACION CLIENTE.

Create an Index

Index name: IND CIE_____

Relation name: CLIENTES_____

Duplicates allowed? N_____

Comment:

Fields_in_the_index:
NUMERO_CLIENTE

ENTER	'finish action'	When you have finished entering the fields in this index
INSERT HERE	'Insert record after'	Add another field to this index

Database being edited: RENTA

DESPUES REGRESAMOS AL MENU PRINCIPAL, A LA OPCION PARA CREAR FORMAS Y/O REPORTES

Main Menu
Rally Definition System

- 1 Databases
- 2 Builder tools
- 3 Edit application
- 4 Utilities
- 5 Miscelaneous
- 6 Run application

Your choice: 2_____

GOLD-Q	'quit action'	Exit RALLY Definition System
HELP	'help'	For more help

AL TECLEAR LA OPCION 2 SE PRESENTA LA SIGUIENTE PANTALLA

Builder Tools

- 1 Create a data source definition
- 2 Create a form/report
- 3 Create the application main menu

Your Choice: 1_____

GOLD-Q	'quit action'	Exit menu without making a choice
HELP	'help'	For more help

LA PANTALLA SIGUIENTE ES PARA LA CREACION DEL NOMBRE DE UNA DS

Create a Data Souce Definition

Name of Data source Definition: **CLIENTES DS**

Type of data source definitions: **RDB**

ENTER	'finish action'	Create the data source
GOLD-BACKSPACE	'first field'	Select a different data
		source to create
GOLD-Q	'quit action'	Return to previous menu
		without creating the
		the data source definition

LA SIGUIENTE PANTALLA ES PARA LISTAR LAS RELACIONES EN LA DS

List of Relations

Local_relation_name	Based_on_relation
CLIENTES	

ENTER	'finish action'	when finished specifying
		relations for RSE
INSERT HERE	'insert record	Specify additional
	after'	relations for this RSE

Data source definition being created: **CLIENTES_DB**

LA SIGUIENTE PANTALLA ES EL MENU DE EXPRESION PARA LA SELECCION DE REGISTROS (OPCION 2 SORTING)

Rdb record Selection Expression

- 1 List of relations
- 2 Record Sorting
- 3 Restrictions
- 4 Projections
- 5 Record count

Your choice: 2_____

GOLD-Q	'quit action'	Exit menu without making a choice
HELP	'help'	For more help

Data source definition being created: CLIENTES_DB

EL LA SIGUIENTE PANTALLA SE DECLARA EL TIPO DE SORT

Record Sorting

Relation.field	Direction of sort	List of sort directions
CLIENTES.NUMERO_CLIENTE	ASCENDING	ASCENDING DESCENDING

Type the name of a sort direction, or use

GOLD-SELECT	'list_of_values'	move to list of values
-------------	------------------	------------------------

Data source definitions being created: CLIENTES_DB

EN ESTA PANTALLA SE PRESENTA EL MODO DE ACCESO DEL DATA SOURCE
(EN ESTE CASO LECTURA Y ESCRITURA COMPARTIDA).

Create RDB Data Source Definition

Name of database: RENTA_____

Record locking option: SHARED_WRITE_____
Wait if locked: N

Edit collection of fields? N

ENTER	'finish action'	Create the data source definitions
GOLD-Q	'quit action'	Return to previous menu without creating the data source definition

Data source definition being created: CLIENTES_DB

LA SIGUIENTE PANTALLA ES PARA TECLEAR LA OPCION PARA CREAR UNA
FORMA O REPORTE

Builder Tools

- 1 Create a data source definition
- 2 Create a form/report
- 3 Create the application main menu

Your Choice: 2_____

GOLD-Q	'quit action'	Exit menu without making a choice
HELP	'help'	For more help

EN LA SIGUIENTE PANTALLA SE PONE EL NOMBRE A LA FORMA

Create a Form/Report

Name of form/report: CLIENTES_ABC_FM

Type the name of the form/report you want to create.

GOLD-Q 'finish action' Return to previous menu without
creating the form/report

EN LA SIGUIENTE PANTALLA SE TECLEAN LOS DATOS GENERALES DE LA
FORMA

Create a Form/Report

Title of form/report: ACTUALIZACION DE CLIENTES

Size of form/report: SCREEN

Comment: PANTALLA PARA ACTUALIZAR DATOS GENERALES DE CLIENTES

ENTER 'finish action' Continue creating this
form/report
GOLD-Q 'quit action' Return to previous menu without
creating a form/report

Name of form/report created: CLIENTES_ABC_FM

EN LA SIGUIENTE PANTALLA SE DEFINE LA FORMA

Groups in a Form/Report

Group name: G1 _____

Comment:

Name of data source definition: CLIENTES DS _____

Formatting:

Labels style: LEFT Repetition: DOES NOT REPEAT _____

Scrolling: N

ENTER	'finish action'	If this form/report has only one group
INSERT HERE	'insert record after'	Add another group to the form/report

Name of form/report created: CLIENTES_ABC_FM

CON LA PANTALLA ANTERIOR SE GENERO LA SIGUIENTE FORMA

ACTUALIZACION DE CLIENTES

Numero de Cliente 99999999

Nombre XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Direccion XXXXXXXXXXXXXXXXXXXXXXX

Telefono XXXXXXXX

Fecha Alta DDDDDDD

Form/Report: CLIENTES_ABC_FM GOLD-A for additional options
Group name: G1 Field name:

LA SIGUIENTE PANTALLA PRESENTA OTRO ESTILO DE CREAR UNA FORMA O REPORTE

Groups in a Form/Report

Group name: **G1**_____

Comment:

Name of data source definition: **PELICULA_DS**_____

Formatting:

Labels style: **ABOVE** Repetition: **DOWN**_____

Scrolling: **N**

ENTER	'finish action'	<i>If this form/report has only one group</i>
INSERT HERE	'insert record after'	<i>Add another group to the form/report</i>

Name of form/report created: **CLIENTES_ABC_FM**

LA SIGUIENTE PANTALLA PRESENTA LA OPCION DEL MENU PARA LA APLICACION

Main Menu
Rally Definition System

- 1 Databases
- 2 Builder tools
- 3 Edit application
- 4 Utilities
- 5 Miscelaneous
- 6 Run application

Your choice: **3**_____

GOLD-Q	'quit action'	Exit RALLY Definition System
HELP	'help'	For more help

LA SIGUIENTE PANTALLA ES EL MENU DE EDICION DE LA APLICACION (OPCION 3 2 ES PARA VER EN DETALLE LA FORMA O REPORTE). LA OPCION 2 2 ES PARA MODIFICAR UN MENU, POR OMISION GENERA UNO).

Edit Application

Primary Object:	Operations:
1 task	1 Create
2 Menu	2 Edit
3 Form/report	3 Delete
4 Form/report packet	4 Rename
5 Data source definition	5 Copy
6 Message	6 Verify

Procedural objects:

7 ADL procedure	:		:
8 Global variable	:	GOLD-D	Exit without saving
9 External program link	:		a choice
10 Parameter packet	:	HELP	For more help
11 Action list	:		:

Formal objects:

12 Date format
13 Number format
14 Number character set

Select an object and an operation: 3 2_____

LA PANTALLA QUE SIGUE ES PARA LA EDICION DE UN MENU

Edit a menu

1 Screen editor
2 Size of menu and response area
3 Menu-choice style options
4 Menu attributes
5 Menu choices
6 Action sites
7 Comment

Your choice: _____

GOLD-Q 'quit action' Exit menu without making a choice
HELP 'help' For more help
Menu being edited: MAIN-MENU

EN ESTA PANTALLA SE TECLEA UNA OPCION DEL MENU

Menu Choices

	Call_type	Object_type	Object_name	Edit options	List of
1	Call	FORM/REPORT PKT	CLIENTES_ABC_PKT		Call types
2	Call	FORM/REPORT PKT	PELICULA_ABC_PKT	Y	CALL
3	CALL	FORM/REPORT PKT	HISTORIA_ABC_PKT	Y	EXECUTE
4	CALL_CMD	COMMAND	FINISH ACTION	Y	RETURN_TO FORK START RESUME CALL_CMD EXECUTE_CMD

To call an action in the same task, select CALL.

To call a RALLY command, select CALL_CMD.

To put the user in a new task, select FORK.

GOLD-SELECT 'list_of_value' Move to list of values

SELECT 'local_functions' Edit the object named

Menu being edited: MAIN_MENU

EN LA SIGUIENTE PANTALLA, CON EL 6 CORREMOS LA APLICACION PARA PROBARLA

Main Menu
Rally Definition System

- 1 Databases
- 2 Builder tools
- 3 Edit application
- 4 Utilities
- 5 Miscelaneous
- 6 Run application

Your choice: 6 _____

GOLD-Q 'quit action' Exit RALLY Definition System
HELP 'help' For more help

EL SITEMA NOS PRESENTA LA SIGUIENTE PANTALLA

SISTEMA DE RENTA DE PELICULAS

- > 1. Actualizacion de Clientes
- 2. Actualizacion de Peliculas
- 3. Actualizacion de Historia
- 4. Terminar

Tecllee Opcion _ _ _ _ _

ACTUALIZACION DE CLIENTES

Numero Cliente 1
Nombre CLIENTE NUMERO 1
Direccion AVE. MADERO 2020 OTE.
Telefono 52-78-53
Fecha Alta 29-Abr-1988

ACTUALIZACION DE PELICULA

Numero Pelicula	Nombre	Clasif	Tipo
1	EL HOMBRE ELEFANTE	B	DRAMA
2	EL HOMBRE LOBO EN LONDRES	C	DRAMA
3	ATRACCION FATAL	C	DRAMA
4	RAMBO	B	ACCION
5	ROCKY	B	ACCION
6	DUMBO	A	INFANTIL
7	EL REGRESO DEL NINJA	B	ACCION
8	LA GUERRA DE LAS GALAXIAS	A	FICCION

ACTUALIZACION DE HISTORIA DE CLIENTES

Numero Cliente	Numero Pelicula	Fecha Renta	Fecha Regreso	Fecha Dev
1	1	29-Abr-1988	30-abr-1988	02-may-1988
1	4	29-Abr-1988	30-Abr-1988	02-May-1988
2	2	29-Abr-1988	30-Abr-1988	30-Abr-1988
5	3	29-Abr-1988	30-Abr-1988	04-May-1988
5	5	29-Abr-1988	30-Abr-1988	05-May-1988

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

- * LENGUAJE DE CUARTA GENERACION Y PRODUCTIVIDAD EN EL DESARROLLO DE SISTEMAS.

DOCUMENTO DE TRABAJO No. 9
MARZO 1983.

LA PRODUCTIVIDAD EN EL DESARROLLO DE APLICACIONES PARA COMPUTADORA.

EBERHARD E. RUDOLPH
DOCTORADO EN FISICA

UNIVERSIDAD DE AUCKLAND
DEPARTAMENTO DE ESTUDIOS ADMINISTRATIVOS

AUCKLAND, NEW ZELAND.

- * METODOLOGIA PARA SELECCION DE EQUIPOS DE COMPUTO.
TESIS PRESENTADA POR EL ING. M. ALBERTO VAZQUEZ PARA OBTENER EL TITULO DE MASTER EN SISTEMAS COMPUTACIONALES EN EL ITESM.
JULIO DE 1985.

- * BRION BOYLE
"SOFTWARE PERFORMANCE EVALUATION"
BYLE FEB. 1984.

- * JOHN A. WINBERT
"AN OVERVIEW OF FOURTH GENERATION LANGUAGE".
CONFERENCIA DEL DECIMO SIMPOSIUM INTERNACIONAL DE SISTEMAS COMPUTACIONALES. I T E S M

- * EBERHARD E. RUDOLPH
"PRODUCTIVITY IN COMPUTER APPLICATION DEVELOPEMENT"
DEPT. OF MANAGEMENT STUDIES
UNIVERSITY OF AUCKLAND.

- * "FOURTH GENERATION LANGUAGES"
SOFTWARE NEWS, JULY. 1984.

- * LENGUAJES DE CUARTA GENERACION
TANDEM COMPUTER DE MEXICO, S.A. DE C.V.

AVE. JUAREZ No. 14, 2o. PISO, C. P. 06050

- * BENEFITS OF SYSTEM DEVELOPMENT WITH FORTH GENERATION TOOLS.
JAMES MARTIN

- * LOGIC AND INFORMATION NETWORK COMPILER (LINC).
REFERENCE MANUAL BURROGHS
(RELATIVE TO THE 10.2 SYSTEM SOFTWARE RELEASE)
COPYRIGHT 1984. BURROUGHS CORPORATION, DETROIT MICHIGAN
48232

PRICED ITEM
PRINTED IN USA
JANUARY 1984.

- * REFERENCE MANUEL BURROUGHS
LOGIC AND INFORMATION
REPORT COMPILER (LIRC)

(RELATIVE TO 10.2 SOFTWARE RELEASE)
COPYRIGHT 1983 BURROUGHS CORPORATION, DETROIT, MICHIGAN
48232

PRINCED ITEM
PRINTED USA
OCTOBER 1983.

- * HEWLETT-PACKARD
BUSINESS REPORT WRITER
HP 3000
19420 HOMESTEAD ROAD, CUPERTINO, CALIFORNIA 95014
COPYRIGHT BY HP COMPANY 1984.

HEWLETT-PACKARD
HP SYSTEM DICTIONARTY
FOR MPE V BASED 900 SERIES
HP 3000 SYSTEMS
19420 HOMESTEAD ROAD, CUPERTINO CALIFORNIA 95014
COPYRIGHT BY HP. COMPANY 1984.

- * DISCOVERING
DATA DICTIONARIES
HEWLETT-PACKARD
19420 HOMESTEAD ROAD, CUPERTINO, CALIFORNIA 95014
COPYRIGHT BY HP COMPANY 1984

- * A NOTE ON PRODUCT NAME SUFFIXES.
COPYRIGHT BY HOWLETT-PACKARD COMPANY 1984.

C O N T E N I D O

- INFORMATION STORAGE
- REPORTING AND PRESENTATION
- APPLICATION DEVELOPMENT
- INTEGRATION AND TRANSPARENCY

- * 1172 FORTH GENERATION LANGUAGES. 1975-1985 (CITATIONS FROM THE INSPEC ; INFORMATION SERVICES FOR THE PHYSICS AND ENGINEERING COMMUNITIES DATA BASE.)

NAT TECHNICAL INFORMATION SERVICE, PB 86-853637.
DEC 1985, 96p.

INSTRUCCIONES REFERENTES AL DESARROLLO E IMPLEMENTACION DE LOS LENGUAJES DE PROGRAMACION DE 4a. GENERACION.

- * MAPPER
MANUAL DE REFERENCIA DE SPERRY UNIVAC
BOX 500
BLUE BELL, PA 19424

- * LINC
MANUAL DE REFERENCIA DE BURROUGHS COMPANY
ONE BURROUGHS, PL
DETROIT MICHIGAN 48232

- * RAPID
MANUAL DE REFERENCIA DE HOWLETT-PACKARD
BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232

- * POWERHOUSE
MANUAL DE REFERENCIA DE HOWLETT-PACKARD

- * OMNIS 3
MANUAL DE REFERENCIA DE
MACINTOUCH
APPLE, COMPANY

- * RALLY Y TEAM DATA
MANUAL DE REFERENCIA DE DIGITAL EQUIPMENT CORPORATION
NEW YORK No. 115 ESQ. ARIZONA
MEXICO, D.F. 03810

* TRABAJOS DE ALUMNOS DEL ITESM, PRESENTADOS EN LA MATERIA:
TECNICAS DE PROGRAMACION Y DESARROLLO AVANZADO DE SISTEMAS
DE INFORMACION DE LA TEORIA DE SISTEMAS.
IMPARTIDA POR EL LIC. SERGIO H. MARTINEZ
OCTUBRE DE 1986.

- TECNICAS DE PROGRAMACION

- RAPID

- STRUCTURED QUERY LANGUAGE (S.Q.L.)

- POWERHOUSE

- OMNIS 3

- DATA MANAGER

- COMPARACION ENTRE RAPID/3000 Y LINC 10.2

- ANALISIS COMPARATIVO RAPID VS LINC

- SPEEDWARE

* SOFTWARE DE CUARTA GENERACION
SOFTWARE PARA DESARROLLO DE APLICACIONES DISPONIBLE
PARA SISTEMAS CENTRALES, MINIS Y MICROCOMPUTADORAS.

