

Jobs en el sistema VSE

Jordi Calopa Bosch

Revisión 1998.09

Indice

Indice.....	2
1. Introducción.....	5
1.1 Respecto de este documento.....	5
2. Conceptos preliminares.....	6
2.1 Power.....	6
2.2 Estructura genérica de los Jobs.....	7
3. Jobs - Sistema base.....	8
3.1 Definir librerías y sublibrerías de usuario.....	8
3.2 Listar entradas de una librería.....	8
3.3 Listar programas de una librería.....	8
3.4 Volcar un programa sobre la cola de punch.....	8
3.5 Borrar programas de una librería de usuario.....	9
3.6 Catalogar un programa en la librería.....	9
3.7 Copiar programas entre librerías.....	9
3.8 Listar el contenido de los discos.....	10
3.9 Inicializar un disco FBA.....	10
3.10 Analizar errores hardware de un disco FBA.....	10
3.11 Volcado de un disco FBA.....	10
3.12 Restore de un disco FBA.....	11
3.13 Restore de un fichero de un disco FBA.....	11
3.14 Lanzar comandos del Power.....	11
3.15 Listar asignaciones de las logical units.....	12
3.16 Listar el procedure de standard labels.....	12
3.17 Listar la label information area.....	12
3.18 Listar secuencia de búsqueda en librerías.....	12
4. Jobs – ICCF.....	13
4.1 Listar contenido de una librería ICCF.....	13
4.2 Listar un objeto de una librería ICCF.....	13
4.3 Backup a disco de todas las librerías del ICCF.....	13
4.4 Restore desde disco de todas las librerías del ICCF.....	14
4.5 Backup a cinta de todas las librerías del ICCF.....	14
4.6 Restore desde cinta de todas las librerías del ICCF.....	14
4.7 Restore desde cinta de una librería ICCF.....	14
4.8 Punchear el contenido de una librería a cinta.....	15
4.9 Leer fichas puncheadas desde una cinta.....	15
4.10 Borrar un código de usuario de ICCF.....	15
5. Jobs – VSAM.....	17
5.1 Definir catálogo de usuario.....	17
5.2 Definir fichero secuencial (ESDS).....	17
5.3 Definir fichero indexado (KSDS).....	17
5.4 Copiar fichero de disco a disco.....	18
5.5 Borrar fichero.....	18
5.6 Listar fichero.....	18

5.7 Listar fichero mediante sort.....	19
6. Procedimientos catalogados.....	20
6.1 IPL (VSE/SPx).....	20
6.2 IPL (VSE/ESA).....	21
Anexo I. Resumen de sentencias del JCL.....	22
ASSGN.....	22
CLOSE.....	22
DATE.....	22
DLBL.....	22
EXEC.....	22
EXEC PROC.....	22
EXTENT.....	22
GOTO label.....	22
JOB.....	22
LIBDEF.....	22
LIBDROP.....	23
LIBLIST.....	23
LISTIO.....	23
LOG.....	23
MTC.....	23
NOLOG.....	23
ON.....	23
OPTION.....	23
PAUSE.....	23
PROC.....	23
PWR.....	23
RESET.....	24
SETPARM.....	24
STDOPT.....	24
TLBL.....	24
UPSI.....	24
/.....	24
/*.....	24
/&.....	24
*.....	24
/+.....	24
Anexo II. Resumen de comandos del Librarian.....	25
ACCESS.....	25
BACKUP.....	25
CATALOG.....	25
CHANGE.....	25
COMPARE.....	25
CONNECT.....	25
COPY.....	25
DEFINE.....	25
DELETE.....	25
GOTO.....	26
INPUT.....	26
LIST.....	26
LISTDIR.....	26

LOCK.....	26
MOVE.....	26
ON.....	26
PUNCH.....	26
RELEASE.....	26
RENAME.....	26
RESTORE.....	27
SEARCH.....	27
UNLOCK.....	27
UPDATE.....	27
/.....	27
/+.....	27
/* o END.....	27
Anexo III. Quick Reference.....	28

1. Introducción

1.1 Respecto de este documento

Todos los ejemplos que se muestran en este documento han sido desarrollados por el autor y probados mayoritariamente bajo sistemas operativos VSE/SP2 y VSE/SP3.

No es posible garantizar que funcionen correctamente con los nuevos sistemas operativos VSE y se entregan sin absolutamente ninguna garantía.

2. Conceptos preliminares

2.1 Power

El Power es un sistema de spooling que gestiona las colas de entrada y salida al sistema, entre las que destacaremos la cola de reader (RDR) que es la de entrada de trabajos y las colas de list (LST) y punch (PUN) de salidas impresa y puncheada respectivamente.

El Power soporta hasta 36 clases (A-Z, 0-9) y diez niveles de prioridad (0-9), en todas las colas de entrada y salida.



Existen varias disposiciones para cada uno de los trabajos de las colas del power, veamos alguna de ellas:

- D El proceso se ejecuta y se elimina de la cola
- K El proceso se ejecuta y se mantiene en la cola con disposición L
- H El proceso se almacena en la cola y con PRELEASE cambia a D
- L El proceso se almacena en la cola y con PRELEASE cambia a K

Ejemplos:

```
* $$ JOB JNM=TREBALL1,DISP=K,CLASS=A,PRI=9
```

Al submitir este job se ejecutará por la partición que tenga asignada la clase A, con prioridad máxima y después quedará almacenado en la cola de reader, disponible para ulteriores lanzamientos.

```
* $$ PRT DISP=D,CLASS=B,PRI=0
```

Cuando se genere este listado se imprimirá, con la mínima prioridad, por la impresora que tenga asignada la clase B y después se eliminará de la cola de list.

2.2 Estructura genérica de los Jobs

Los scripts del sistema operativo VSE, a los que habitualmente se denominan Jobs, se estructuran en dos niveles: Los jobs de POWER y los jobs de DOS. Un job de POWER puede contener uno a varios jobs DOS, según se muestra en el siguiente ejemplo.

* \$\$ JOB JNM=JOBPWR1,DISP=D,CLASS=0	Inicio de job de POWER
* \$\$ LST DISP=K,CLASS=A	Atributos salida impresa
* \$\$ PUN DISP=D	Atributos salida puncheada
// JOB JOBDOS1	Inicio job DOS
...	
...	
// EXEC PROGRAM1	Step
/*	
/&	Final job DOS
// JOB JOBDOS2	Inicio job dOS
...	
...	
// EXEC PROGRAM2, SIZE=AUTO	Step
/*	
...	
...	
// EXEC PROGRAM3, SIZE=512K	Step
/*	
/&	Final job DOS
* \$\$ E0J	Final job de POWER

Notas:

- Por defecto en un job power, la clase 0 envía el trabajo a la partición BG, la clase 1 envía el trabajo a la partición F1, la clase 2 a F2, la clase 3 a F3 y así sucesivamente hasta F9.
- Todas las sentencias, contenidas en un job DOS, que tengan dos slash (//) en las primeras posiciones, quedarán obsoletas al final del job, cuando se encuentre la sentencia /&

3. Jobs - Sistema base

3.1 Definir librerías y sublibrerías de usuario

```
* $$ JOB JNM=DEFLIBS,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB DEFLIBS
// EXEC LIBR,PARM='MSHP'
   DEFINE L=PRUEBAS
   DEFINE S=PRUEBAS.TEST
   DEFINE S=PRUEBAS.TEST2
/*
/ &
* $$ E0J
```

3.2 Listar entradas de una librería

```
* $$ JOB JNM=LISTDIR,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB LISTDIR
// EXEC LIBR,PARM='MSHP'
   ACCES S=PRUEBAS.TEST
   LISTDIR *,*
/*
/ &
* $$ E0J
```

3.3 Listar programas de una librería

```
* $$ JOB JNM=LISTPGM,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB LISTPGM
// EXEC LIBR,PARM='MSHP'
   ACCES S=PRUEBAS.TEST
   LIST PGMTEST1.C
   LIST PGMTEST2.C
/*
/ &
* $$ E0J
```

3.4 Volcar un programa sobre la cola de punch


```

* $$ JOB JNM=LIBRPGM,DISP=D,CLASS=0
* $$ PUN DISP=H
// JOB LIBRPGM
// EXEC LIBR,PARM='MSHP'
  ACCES S=PRUEBAS.TEST
  PUNCH PGMTEST.C
/*
/&
* $$ E0J

```

3.5 Borrar programas de una librería de usuario

```

* $$ JOB JNM=DELPGM,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB DELPGM
// EXEC LIBR,PARM='MSHP'
  ACCES S=PRUEBAS.TEST
  DELETE TEST*.*           ' Todos los que empiezan por TEST
/*
/&
* $$ E0J

```

3.6 Catalogar un programa en la librería

```

* $$ JOB JNM=CATALPGM,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB CATALPGM
// EXEC LIBR,PARM='MSHP'
  ACCES S=PRUEBAS.TEST
  CATALOG PGMTEST1.C  REPLACE=YES
. . .
. . .
. . .
/+
/*
/&
* $$ E0J

```

3.7 Copiar programas entre librerías

```

* $$ JOB JNM=COPYPGM,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB COPYPGM
// EXEC LIBR,PARM='MSHP'
  CONNECT S=LIB.ENTRADA:LIB.SALIDA
  COPY *.A  REPLACE=YES   ' Copia solo fuentes assembler
/*
/&
* $$ E0J

```

3.8 Listar el contenido de los discos

```
* $$ JOB JNM=LVT0C,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB LVT0C120
// ASSGN SYS005,SYSLST
// ASSGN SYS004,120
// EXEC LVT0C
/*
/&
// JOB LVT0C121
// ASSGN SYS005,SYSLST
// ASSGN SYS004,121
// EXEC LVT0C
/*
/&
* $$ E0J
```

3.9 Inicializar un disco FBA

```
* $$ JOB JNM=INITFBA,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB INITFBA
// EXEC ICKDSF,SIZE=AUTO
INIT UNIT(121) NVFY PURGE FBAVTOC(712690,99,1024) VOLID(SYSWKA)
/*
/&
* $$ E0J
```

3.10 Analizar errores hardware de un disco FBA

```
* $$ JOB JNM=ANALIFBA,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB ANALIFBA
// ASSGN SYS001,122
// EXEC ICKDSF,SIZE=AUTO
ANALYZE SYSNAME(SYS001) SCAN
/*
/&
* $$ E0J
```

3.11 Volcado de un disco FBA

```
* $$ JOB JNM=FBADUMP,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB FBADUMP
// UPSI 000
```

```
// ASSGN SYS004,120
// PAUSE *** MONTAR CINTA PARA VOLCADO EN UNIDAD 380 ***
// TLBL UOUT,'FCOPY120'
// ASSGN SYS005,380,D0
// EXEC FCOPYFB
DUMP VOLUME LIST
/*
/&
* $$ E0J
```

3.12 Restore de un disco FBA

```
* $$ JOB JNM=FBAREST,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB FBAREST
// UPSI 000
// PAUSE *** MONTAR CINTA EN UNIDAD 380 PARA RESTORE DEL DISCO 120 ***
// TLBL UIN,'FCOPY120'
// ASSGN SYS004,380
// ASSGN SYS005,120
// EXEC FCOPYFB
RESTORE VOLUME NOVERIFY LIST OV=SYSWKB
/*
/&
* $$ E0J
```

3.13 Restore de un fichero de un disco FBA

```
* $$ JOB JNM=FBARESF,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB FBARESF
// UPSI 100
// PAUSE *** MONTAR CINTA EN UNIDAD 380 PARA RESTORE DEL DISCO 120 ***
// TLBL UIN,'FCOPY120'
// ASSGN SYS004,380
// ASSGN SYS005,120
// EXEC FCOPYFB
RESTORE FILE='MI.FICHERO' -
NOVERIFY
/*
/&
* $$ E0J
```

3.14 Lanzar comandos del Power

```
* $$ JOB JNM=PWRMNGR,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB PWRMNGR
NOLOG
// LIBDEF PHASE,SEARCH=IJSYSRS.SYSLIB
// EXEC DTRIJMGR
/.PCMD PALTER LST,TEST*,DISP=D,CLASS=A,NODE=PNODE1
```

```
/.PCMD PDELETE LST,TEST*  
/*  
/&  
* $$ E0J
```

3.15 Listar asignaciones de las logical units

```
* $$ JOB JNM=JLISTUNI,DISP=D,CLASS=0  
* $$ LST DISP=H  
// JOB JLISTUNI  
// LISTIO ALL  
/*  
/&  
* $$ E0J
```

3.16 Listar el precedente de standard labels

```
* $$ JOB JNM=LISTSTD,DISP=D,CLASS=0  
* $$ LST DISP=H  
// JOB LISTSTD  
// EXEC LIBR,PARM='MSHP'  
ACCES S=IJSYSRS.SYSLIB  
LIST STDLABUS.PROC  
/*  
/&  
* $$ E0J
```

3.17 Listar la label information area

```
* $$ JOB JNM=LISTSERV,DISP=D,CLASS=0  
* $$ LST DISP=H  
// JOB LISTSERV  
// EXEC LSERV  
/*  
/&  
* $$ E0J
```

3.18 Listar secuencia de búsqueda en librerías

```
* $$ JOB JNM=LSTCHAIN,DISP=D,CLASS=0  
* $$ LST DISP=H  
// JOB LSTCHAIN  
// LIBLIST PROC,BG  
// LIBLIST PHASE,F1  
// LIBLIST SOURCE, F3  
/*  
/&  
* $$ E0J
```

4. Jobs - ICCF

Nota: Para realizar el tratamiento de librerías del ICCF, con el programa DTSINIT funcionando, es preciso desconectar el fichero DTSFILE, mediante el comando: /DISC DTSFILE

4.1 Listar contenido de una librería ICCF

```
* $$ JOB JNM=LISTLIB,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB LISTLIB
// ASSGN SYS010,121
// EXEC DTSUTIL
DSERV L(31) SORTED
/*
/&
* $$ E0J
```

4.2 Listar un objeto de una librería ICCF

```
* $$ JOB JNM=LISTOBJ,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB LISTOBJ
// ASSGN SYS010,121
// EXEC DTSUTIL
PR M(31 MIPROGRAMA)
/*
/&
* $$ E0J
```

4.3 Backup a disco de todas las librerías del ICCF

```
* $$ JOB JNM=BACKUPI,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB BACKUPI
// ASSGN SYS040,121
// DLBL DTSBKUP,'BACKUP.ICCF',99/360
// EXTENT SYS040,SYSWK1,1,,500000,100000
// EXEC PROC=DTRICCF
// EXEC DTSUTIL
BACKUP
/*
/&
* $$ E0J
```

4.4 Restore desde disco de todas las librerías del ICCF

```
* $$ JOB JNM=RESTORI,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB RESTORI
// ASSGN SYS040,121
// DLBL DTSRSTR,'BACKUP.ICCF',99/360
// EXTENT SYS040,SYSWK1,1,,500000,100000
// EXEC PROC=DTRICCF
// EXEC DTSUTIL
RESTORE SORTED
/*
/&
* $$ E0J
```

4.5 Backup a cinta de todas las librerías del ICCF

```
* $$ JOB JNM=BACKUPT,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB BACKUPT
// PAUSE *** MONTAR CINTA PARA BACKUP EN UNIDAD 380 ***
// ASSGN SYS005,380
// TLBL DTSBKUP,,,111111
// EXEC PROC=DTRICCF
// EXEC DTSUTIL
BACKUP
/*
/&
* $$ E0J
```

4.6 Restore desde cinta de todas las librerías del ICCF

```
* $$ JOB JNM=RESTORT,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB RESTORT
// PAUSE *** MONTAR CINTA DEL ULTIMO BACKUP EN UNIDAD 380 ***
// ASSGN SYS004,380
// TLBL DTSRSTR
// MTC REW,SYS004
// EXEC PROC=DTRICCF
// EXEC DTSUTIL
RESTORE SORTED
/*
/&
* $$ E0J
```

4.7 Restore desde cinta de una librería ICCF

```
* $$ JOB JNM=RESTORT,DISP=D,CLASS=0
* $$ LST DISP=H
```

```
// JOB RESTORT
// PAUSE *** MONTAR CINTA DEL ULTIMO BACKUP EN UNIDAD 380 ***
// ASSGN SYS004,380
// TLBL DTSRSTR,,111111
// MTC REW,SYS004
// EXEC PROC=DTRICCF
// EXEC DTSUTIL
RESTORE TOLIBRARY(38) FROMLIBRARY(38)
/*
/&
* $$ E0J
```

4.8 Punchear el contenido de una librería a cinta

```
* $$ JOB JNM=PUNCLIB,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB PUNCLIB
// PAUSE *** MONTAR CINTA DE SALIDA EN UNIDAD 380 ***
// ASSGN SYSPCH,380
// EXEC PROC=DTRICCF
// EXEC DTSUTIL
PUNCH LIB(99)
/*
// CLOSE SYSPCH,UA
/&
* $$ E0J
```

4.9 Leer fichas puncheadas desde una cinta

```
* $$ JOB JNM=CARDLIB,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB CARDLIB
// PAUSE *** MONTAR CINTA CON FICHAS EN 380 ***
// ASSGN SYSIPT,380
// PAUSE *** ENTER CUANDO ESTE LA CINTA READY ***
// EXEC PROC=DTRICCF
// EXEC DTSUTIL
CARD
/*
// CLOSE SYSIPT,02C
/&
* $$ E0J
```

4.10 Borrar un código de usuario de ICCF

```
* $$ JOB JNM=DELUSER,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB DELUSER
// EXEC PROC=DTRICCF
// EXEC DTSUTIL
DEL USER(XXXX)
END
```

/*
/&
* \$\$ EOJ

5. Jobs – VSAM

5.1 Definir catálogo de usuario

```

* $$ JOB JNM=DEFCATUS,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB DEFCATUS
// EXEC IDCAMS, SIZE=AUTO

DEFINE USERCATALOG NAME(UCATSYSB)      -
        DEDICATE                        -
        VOLUME(SYSWKB))

IF LASTCC NE 0 THEN CANCEL JOB
/*
* Incluir catálogo en las standard labels
// OPTION STDLABEL=ADD
// DLBL CATSYSB,'UCATSYSB',,VSAM
/*
// EXEC IESVCLUP,SIZE=0AUTO
A UCATSYSB
/*
/&
* $$ E0J

```

5.2 Definir fichero secuencial (ESDS)

```

* $$ JOB JNM=DEFESDS,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB DEFESDS
// EXEC IDCAMS, SIZE=AUTO

DEFINE CLUSTER (NAME(mi.fichero.ESDS) -
        SHR(2)                          -
        TRACKS(5,2)                     -
        CONTROLINTERVALSIZE(4096)       -
        RECORDSIZE(80,80)               -
        NONINDEXED)                     -
        DATA (NAME(mi.fichero.ESDS .DATA) )

/*
/&
* $$ E0J

```

5.3 Definir fichero indexado (KSDS)

```

* $$ JOB JNM=DEFKSDS,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB DEFKSDS

```

```
// EXEC IDCAMS, SIZE=AUTO

DEFINE CLUSTER (NAME(mi.fichero.KSDS)           -
              INDEXED                           -
              KEYS(6 1)                          -
              RECSZ(80 80)                       -
              TRACKS(1,1)                       -
              CISZ(4096)                        -
              FREESPACE(5 3) )                  -
DATA
  (NAME(mi.fichero.KSDS .DATA))              -
INDEX
  (NAME(mi.fichero.KSDS .INDEX))             -

/*
/ &
* $$ EOJ
```

5.4 Copiar fichero de disco a disco

```
* $$ JOB JNM=COPYFILV,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB COPYFILV
// BLBL ENTRADA,'MI.FICHERO',0,VSAM,CAT=CATSYSB
// BLBL SALIDA,'COPIA.MI.FICHERO',0,VSAM,CAT=CATSYSB
// EXEC IDCAMS, SIZE=AUTO

REPRO INFILE(ENTRADA) OUTFILE(SALIDA)

IF MAXCC NE 0 THEN CANCEL
/*
/ &
* $$ EOJ
```

5.5 Borrar fichero

```
* $$ JOB JNM=DELFILV,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB DELFILV
// EXEC IDCAMS, SIZE=AUTO

DELETE(mi.fichero) CLUSTER PURGE           -
CATALOG(UCATSYSB)

/*
/ &
* $$ EOJ
```

5.6 Listar fichero

```
* $$ JOB JNM=LISTFILV,DISP=D,CLASS=0
```

```
* $$ LST DISP=H
// JOB LISTFILV
DLBL UCAT,'UCATSYSB',,VSAM
DLBL KSDS,'mi.fichero.KSDS',,VSAM,CAT=UCAT
// EXEC IDCAMS, SIZE=AUTO

PRINT INFILE(KSDS) CHARACTER COUNT(3)

/*
/&
* $$ EOJ
```

5.7 Listar fichero mediante sort

```
* $$ JOB JNM=SortLIST,DISP=D,CLASS=0
* $$ LST DISP=H
// JOB SortLIST
* NOTA: UPSI 10000 CHAR  UPSI 01000 HEX
// DLBL SORTIN1,'MI.FICHERO',,VSAM,CAT=CATSYSB
// EXEC SORT,SIZE=512K
    SORT FIELDS=(1,10,CH,A)
    RECORD TYPE=F,LENGTH=250
    INPFIL VSAM
    OUTFIL SYSLST
    END
/*
/&
* $$ EOJ
```

6. Procedimientos catalogados

6.1 IPL (VSE/SPx)

```

CATALOG $IPLVAE.PROC  REPLACE=YES
01F,$$A$SUP3,VSIZE=30M,VI0=512K,VP00L=64K,P,NOLOG
ADD 01F,3277          Consola
ADD 120,FBA          Discos
ADD 121,FBA
ADD 122,FBA
ADD 123,FBA
ADD 124,FBA
ADD 125,FBA
ADD 126,FBA
ADD 127,FBA
ADD 290,3277        Terminales
ADD 291,3277
ADD 292,3277
ADD 293,3277
ADD 294,3277
ADD 295,3277
ADD 296,3277
ADD 297,3277
ADD 298,3277
ADD 299,3277
ADD 380,3420T9,D0   U. Cinta
ADD 381,3420T0,D0
ADD 02E,PRT1        Impresoras
ADD 03E,PRT1
ADD 04E,PRT1
ADD 05E,PRT1
ADD 06E,PRT1
ADD 07E,PRT1
ADD FEC,3505        POWER dummy reader
ADD FED,3525P       POWER dummy punch
ADD FEE,PRT1        POWER dummy printer
ADD FEF,PRT1        POWER dummy printer
ADD FFA,3505        ICCF internal reader
ADD FFC,3505        ICCF dummy RDR
ADD FFD,3525P       ICCF dummy PUN
ADD FFE,PRT1        ICCF dummy PRT
SET ZONE=WEST/00/00
DEF SYSCAT=DOSRES
DEF SYSREC=SYSWK1
DPD VOLID=DOSRES,BLK=121954,BLK=16864,TYPE=N,DSF=N
DPD VOLID=DOSRES,BLK=442805,NBLK=33725,TYPE=N,DSF=N
DPD VOLID=SYSWK1,BLK=377766,TYPE=N,DSF=N
DLA NAME=AREA1,VOLID=DOSRES,BLK=59118,NBLK=744,DSF=N
SYS JA=YES
SYS BUFSIZE=200

```

```

SY5 BUFLD=YES
SVA PSIZE=534K,SDL=250,GETVIS=64K
/+

```

6.2 IPL (VSE/ESA)

```

CATALOG $IPLESA.PROC  REPLACE=YES
01F,$$A$SUPX,VSIZE=70M,VI0=956K,VP00L=256K,P,NOLOG
ADD 080,3277          Consola
ADD 085,3277
ADD 086,3277
ADD 087,3277
ADD 088,3277
ADD 089,3277
ADD 00C,2540R        RDR
ADD 00D,2540P        PUN
ADD 00E,1403         Impresora
ADD 03E,4248
ADD 400:40F,3791L    Controlador com.
ADD 700:703,3480,00 U. Cinta
ADD 150:157,ECKD     Discos
ADD 510,CTCA
ADD 511,CTCA
ADD 512,CTCA
ADD FEC,3505         POWER dummy reader
ADD FED,2520P2       POWER dummy punch
ADD FEE,PRT1         POWER dummy printer
ADD FEF,PRT1         POWER dummy printer
ADD FFA,3505         ICCF internal reader
ADD FFC,3505         ICCF dummy RDR
ADD FFD,2520P2       ICCF dummy PUN
ADD FFE,PRT1         ICCF dummy PRT
ADD FFF,CONS         dummy Console
SET ZONE=WEST/00/00
DEF SYSCAT=DOSRES
DEF SYSREC=SYSWK1
SYS JA=YES
SYS BUFSIZE=1500
SYS NPARTS=24
SYS SEC=NO
SYS PASIZE=20M
SYS RSIZE=16M
SYS SPSIZE=0K
SYS BUFLD=YES
DPD VOLID=DOSRES,CYL=209,NCYL=12,TYPE=N,DSF=N
DPD VOLID=SYSWK1,CYL=422,NCYL=12,TYPE=N,DSF=N
DPD VOLID=DOSRES,CYL=398,NCYL=12,TYPE=N,DSF=N
DPD VOLID=SYSWK1,CYL=1053,TYPE=N,DSF=N
DLA NAME=AREA1,VOLID=DOSRES,CYL=60,NCYL=3,DSF=N
SVA SDL=300,GETVIS=(768K,3M),PSIZE=(256K,2000K)
/+

```

Anexo I. Resumen de sentencias del JCL

ASSGN

Asigna, en tiempo de ejecución, una dirección de dispositivo específica (unidad física) al nombre de la unidad simbólica utilizada (unidad lógica).

CLOSE

Cierra un sistema o una unidad lógica asignada a una cinta o un disco.

DATE

Contiene una fecha que se coloca en la región de comunicaciones.

DLBL

Contiene información de etiquetas de archivos para la comprobación y creación de etiquetas de disco.

EXEC

Indica el final de las sentencias de control de trabajo y el inicio de la ejecución del step.

EXEC PROC

Llama a un procedimiento catalogado y define valores para parámetros simbólicos.

EXTENT

Define cada área, o extensión, de un archivo de disco.

GOTO label

Hace que el "job control" salte todas las sentencias siguientes (excepto JOB, /&, /+) hasta la sentencia de etiqueta especificada (label).

JOB

Indica el comienzo de un Job DOS con la información de control del trabajo.

LIBDEF

Define cadenas de bibliotecas.

LIBDROP

Borra la definición de cadenas de bibliotecas.

LIBLIST

Lista las definiciones de cadenas de bibliotecas.

LISTIO

Se utiliza para obtener una lista de asignaciones de E/S de SYSLOG o SYSLST.

LOG

Hace que el sistema registre todos los comandos y declaraciones de control de trabajo que ocurren dentro del proceso.

MTC

Controla operaciones en cintas magnéticas.

NOLOG

Detiene la lista de comandos y sentencias de control de trabajos.

ON

Hace que se realice la acción especificada si la condición especificada es verdadera después de cualquier paso en la siguiente secuencia de trabajos.

OPTION

Establece una o más de las opciones de control de trabajos.

PAUSE

Provoca una pausa inmediatamente después de procesar esta instrucción o al final del paso actual del trabajo.

PROC

Define e inicializa parámetros simbólicos en un procedimiento.

PWR

Pasa un comando PRELEASE o PHOLD al power.

RESET

Restablece las asignaciones de E/S a las asignaciones estándar.

SETPARM

Asigna una cadena de caracteres o un código de retorno al parámetro especificado.

STDOPT

Restablece los valores predeterminados del sistema.

TLBL

Contiene información de etiquetas de archivos para comprobar y escribir etiquetas de cinta.

UPSI

Permite a un programa de usuario interactuar con el job control leyendo los valores de la UPSI.

/.
Etiqueta

/*
Final de fichero de datos

/&
Final de job DOS

*
Comentario.

/+
Final de procedimiento o fin de entrada de datos del librarian

Anexo II. Resumen de comandos del Librarian

ACCESS

ACCESS debe ejecutarse antes de cualquier comando que tenga una especificación de miembro como operando.

BACKUP

BACKUP hace que las bibliotecas, las subbibliotecas (incluidos los archivos SYSRES) o los miembros se copien en cinta.

CATALOG

CATALOG provoca que los datos que le siguen sean catalogados bajo el nombre y tipo especificado.

CHANGE

CHANGE se puede utilizar para cambiar el atributo REUSE de una subbiblioteca.

COMPARE

Se utiliza para comparar bibliotecas, subbibliotecas o miembros, y proporciona una lista de las diferencias.

CONNECT

El comando CONNECT debe usarse antes que los comandos COPY, MOVE o COMPARE que tienen una especificación de miembro como operando.

COPY

Se utiliza para copiar bibliotecas, subbibliotecas o miembros.

DEFINE

DEFINE se utiliza para crear bibliotecas del sistema (archivos SYSRES), bibliotecas privadas y subbibliotecas.

DELETE

Permite eliminar miembros, subbibliotecas o bibliotecas.

GOTO

GOTO permite saltar el flujo de ejecución hasta la etiqueta indicada.

INPUT

INPUT hace que Librarian lea los siguientes comandos de la unidad lógica especificada.

LIST

LIST hace que el contenido de uno o más miembros se muestre en SYSLST o SYSLOG.

LISTDIR

Se utiliza para mostrar el contenido de un directorio. El resultado es una lista ordenada alfanuméricamente.

LOCK

LOCK hace que un miembro de la biblioteca se bloquee para cualquier acceso de escritura o actualización (si, por ejemplo, un usuario en una estación de trabajo desea editar un miembro).

MOVE

El comando MOVE funciona de manera similar al comando COPY, excepto que los datos, que se han movido a una biblioteca o subbiblioteca de destino, se eliminan de la ubicación de origen después de haber sido copiados.

ON

ON permite la ejecución condicional de secuencias de comandos de Librarian en modo por lotes.

PUNCH

PUNCH hace que el contenido de uno o más miembros se "perfore" en el dispositivo de salida SYSPCH.

RELEASE

RELEASE anula el atributo de biblioteca o subbiblioteca AUTOMÁTICO.

RENAME

Cambia el nombre y/o el tipo de uno o más miembros, o los nombres de una o más subbibliotecas. Si el nuevo nombre ya existe, la función no se ejecuta.

RESTORE

Restaurar copia de seguridad de biblioteca, subbiblioteca o miembro.

SEARCH

SEARCH se utiliza para buscar un miembro de la biblioteca en la biblioteca o bibliotecas especificadas.

UNLOCK

UNLOCK hace que un miembro de la biblioteca que se ha bloqueado para cualquier acceso de escritura o actualización se desbloquee nuevamente.

UPDATE

UPDATE permite modificar el contenido de un miembro agregando, eliminando o reemplazando líneas.

/.

Declaración de etiqueta que se utiliza en secuencias de comandos condicionales, marcando el punto hasta el que puede llegar desde un comando GOTO.

/+

Declaración de fin de datos para la entrada al comando CATALOG del bibliotecario.

/* o END

Declaración de fin de sesión del bibliotecario.

Anexo III. Quick Reference

Job Control language (JCL)

ASSGN Asigna, en tiempo de ejecución, una dirección de dispositivo unidad física) al nombre de la unidad simbólica unidad lógica).

CLOSE Cierra un sistema o una unidad lógica asignada a una cinta o un disco.

DATE Contiene una fecha que se coloca en la región de comunicaciones.

DLBL Contiene información de etiquetas de archivos para la comprobación y creación de etiquetas de disco.

EXEC Indica el final de las sentencias de control de trabajo y el inicio de la ejecución del step.

EXEC PROC Llama a un procedimiento catalogado y define valores para parámetros simbólicos.

EXTENT Define cada área, o extensión, de un archivo de disco.

GOTO label Hace que el JCL salte todas las sentencias siguientes (excepto JOB, /&, /+) hasta la sentencia de etiqueta especificada (label).

JOB Indica el comienzo de un Job DOS con la información de control del trabajo.

LIBDEF Define cadenas de bibliotecas.

LIBDROP Borra la definición de cadenas de bibliotecas.

LIBLIST Lista las definiciones de cadenas de bibliotecas.

LISTIO Se utiliza para obtener una lista de asignaciones de E/S de SYSLOG o SYSLST.

LOG Hace que el sistema registre todos los comandos y declaraciones de control de trabajo.

MTC Controla operaciones en cintas magnéticas.

NOLOG Detiene la lista de comandos y sentencias de control de trabajos.

ON Hace que se realice la acción especificada si la condición especificada es verdadera.

OPTION Establece una o más de las opciones de control de trabajos.

PAUSE Provoca una pausa inmediatamente o al final del paso actual del trabajo.

PROC Define e inicializa parámetros simbólicos en un procedimiento.

PWR Pasa un comando PRELEASE o PHOLD al power.

RESET Restablece las asignaciones de E/S a las asignaciones estándar.

SETPARM Asigna una cadena de caracteres o un código de retorno al parámetro especificado.

STDOPT Restablece los valores predeterminados del sistema.

TLBL Contiene información de etiquetas de archivos para comprobar y escribir etiquetas de cinta.

UPSI Permite a un programa de usuario interactuar con el job control leyendo los valores de la UPSI.

/. Etiqueta

/* Final de fichero de datos

/& Final de job DOS

* Comentario.

/+ Final de procedimiento o de entrada de datos del LIBR.

Librarian (LIBR)

ACCESS ACCESS debe ejecutarse antes de cualquier comando que tenga una especificación de miembro como operando.

BACKUP BACKUP hace que las bibliotecas, las subbibliotecas (incluidos los archivos SYSRES) o los miembros se copien en cinta.

CATALOG Provoca que los datos que le siguen sean catalogados bajo el nombre y tipo especificado, hasta en contrar un /+

CHANGE Cambia el atributo REUSE de una subbiblioteca.

COMPARE Se utiliza para comparar bibliotecas, subbibliotecas o miembros.

CONNECT Se utiliza antes que los comandos COPY, MOVE o COMPARE que tienen una especificación de miembro como operando.

COPY Se utiliza para copiar bibliotecas, subbibliotecas o miembros.

DEFINE Se utiliza para crear bibliotecas del sistema (archivos SYSRES), bibliotecas privadas y subbibliotecas.

DELETE Permite eliminar miembros, subbibliotecas o bibliotecas.

GOTO Permite saltar el flujo de ejecución hasta la etiqueta indicada.

INPUT INPUT hace que Librarian lea los siguientes comandos de la unidad lógica especificada.

LIST LIST hace que el contenido de uno o más miembros se muestre en SYSLST o SYSLOG.

LISTDIR Se utiliza para mostrar el contenido de un directorio, dando una lista ordenada.

LOCK Provoca que un miembro de la biblioteca se bloquee para cualquier acceso de escritura o actualización.

MOVE Es como el comando COPY, excepto que los datos que se han movido se eliminan de la ubicación de origen.

ON Permite la ejecución condicional de secuencias de comandos de Librarian en modo por lotes.

PUNCH Provoca que el contenido de uno o más miembros se "perfore" en el dispositivo de salida SYSPCH.

RELEASE Anula el atributo de biblioteca o subbiblioteca AUTOMÁTICO.

RENAME Cambia el nombre y/o el tipo de uno o más miembros, o los nombres de una o más subbibliotecas. Si el nuevo nombre ya existe, la función no se ejecuta.

RESTORE Restaurar copia de seguridad de biblioteca, subbiblioteca o miembro.

SEARCH Se utiliza para buscar un miembro de la biblioteca en la biblioteca o bibliotecas especificadas.

UNLOCK Hace que un miembro de la biblioteca que se ha bloqueado para cualquier acceso de escritura o actualización se desbloquee nuevamente.

UPDATE Permite modificar el contenido de un miembro agregando, eliminando o reemplazando líneas.

/. Declaración de etiqueta que se utiliza en secuencias de comandos condicionales, marcando el punto hasta el que puede llegar desde un comando GOTO.

/+ Declaración de fin de datos para la entrada al comando CATALOG del bibliotecario.

/* o END Declaración de fin de sesión del bibliotecario.