

Ejecutar pgm con DB2 en Batch

```
DB2BATCH
-----
      JMP                Ejecutar pgm con BD2 en Batch                USUARIO: USRJMPP
-----
Command ==>

Este procedimiento submite un Job, para ejecutar un programa cobol-DB2

Programa.....: KD01069_ Cargar ddnames memorizadas.: S (S/N)
File del JCL a submitir: RAIZ.USRJMP.DB2BATCH.JCL
Submitir JCL generado..: N (S/N) Editar Jcl: N (S/N) Editar lib: N (S/N)

      DDname                DSN                Disp  Lrecl  RecFM
-----
      EJ001E_ RAIZ.USRJMP.CADENA69.PARAM8_      OLD_
      EJ002I_ RAIZ.USRJMP.CADENA69.KD01069.ESTADIS_ NEW_ 133_ FBA
      EJ003S_ RAIZ.USRJMP.CADENA69.KD01069.INFORME_ NEW_ 95_ FB_
      EJ415I_ RAIZ.USRJMP.CADENA69.KD01069.LISTADO_ NEW_ 133_ FB_
      EJ416S_ RAIZ.USRJMP.CADENA69.KD01069.INSERTAR_ NEW_ 46_ FB_

Parametros DCBs nuevas: TRK Cantidad Primaria: 100 Secundaria: 010

Comandos de linea: E V B Pulse F3 para salir o INTRO para continuar
```

1	Intro	1
2	Valores por defecto	2
3	Librería de tablas	3
4	Cargar Ddnames	4
5	Ejemplo practico	4
6	JCL generado	5
7	JCL completo:	6
8	Fichero de comandos REXX: DB2BATCH	7
9	Panel: DB2BATCH	16

1 Intro

Esta utilidad permite parametrizar desde un panel de control todas las ddnames que se utilizan en la ejecucion de un programa con DB2 en Batch.

Ademas, memoriza estos parametros en una librería de usuario para no tener que volver a escribirlos la proxima vez.

Comparando esta utilidad con el tener guardados distintos JCLs, tiene la ventaja de que todos los ficheros que intervienen asi como sus parametros asociados (ddname y dcb) se muestran en un unico panel.

3 Librería de tablas

Además de esto, guarda todos los parámetros especificados en una tabla de una librería de usuario. Esta librería si no existe se crea automáticamente la primera vez que la necesita, de modo que cada usuario tenga sus propios parámetros para sus propios programas.

Solo existe una excepción a lo explicado en el párrafo anterior: Si no se ha especificado ninguna DDM, entonces ni borra ni memoriza, pues se entiende que ha sido un error.

La siguiente figura muestra el contenido de la librería de tablas.

```
Menu  Functions  Utilities  Help
-----
ISREPO01  RAIZ.USRJMP.DB2BATCH.DDNAMES          Row 00001 of 00005
Command ==>                               Scroll ==> CSR
  Name      Prompt      Size      Created      Changed      ID
.  AAAA
.  FE
.  KD01069
.  PRUEBA
.  PRUEBA2
**End**
```

Esta otra figura muestra el contenido de la tabla que ha servido de ejemplo

```
File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
-----
ISREDDE2  RAIZ.USRJMP.DB2BATCH.DDNAMES (KD01069) - 01.00  Columns 00001 00072
Command ==>                               Scroll ==> CSR
***** ***** Top of Data *****
000001  EJ001E  RAIZ.USRJMP.CADENA69.PARAM8          OLD
000002  EJ002I  RAIZ.USRJMP.CADENA69.KD01069.ESTADIS  NEW 133 FBA
000003  EJ003S  RAIZ.USRJMP.CADENA69.KD01069.INFORME  NEW 95 FB
000004  EJ415I  RAIZ.USRJMP.CADENA69.KD01069.LISTADO  NEW 133 FB
000005  EJ416S  RAIZ.USRJMP.CADENA69.KD01069.INSERTAR  NEW 46 FB
000006
000007
***** ***** Bottom of Data *****
```

4 Cargar Ddnames

Usando la opción cargar ddnames memorizadas recuperaremos todos los parámetros asociados a la ejecución de ese programa.

```
Programa.....: KD01069_ Cargar ddnames memorizadas.: S (S/N)
```

Con lo que obtendremos:

5 Ejemplo practico

La siguiente figura muestra el resultado del panel, ya sea:

- la primera vez que se especifican los parametros
- o el resultado de especificar nombre de programa y especificar **S** en cargar ddnames memorizadas.

```
DB2BATCH
-----
JMP          Ejecutar pgm con BD2 en Batch          USUARIO: USRJMP
-----
Command ==> _____

Este procedimiento submite un Job, para ejecutar un programa cobol-DB2

Programa.....: KD01069_ Cargar ddnames memorizadas.: S (S/N)
File del JCL a submitir: RAIZ.USRJMP.DB2BATCH.JCL
Submitir JCL generado.: N (S/N)  Editar Jcl: N (S/N)  Editar lib: N (S/N)

  DDname                DSN                Disp  Lrecl  RecFM
-----
  EJ001E  RAIZ.USRJMP.CADENA69.PARAM8      OLD
  EJ002I  RAIZ.USRJMP.CADENA69.KD01069.ESTADIS  NEW  133   FBA
  EJ003S  RAIZ.USRJMP.CADENA69.KD01069.INFORME  NEW   95   FB
  EJ415I  RAIZ.USRJMP.CADENA69.KD01069.LISTADO  NEW  133   FB
  EJ416S  RAIZ.USRJMP.CADENA69.KD01069.INSERTAR  NEW   46   FB

Parametros DCBs nuevas: TRK  Cantidad Primaria: 100  Secundaria: 010

Comandos de linea: E V B  Pulse F3 para salir o INTRO para continuar
```

El resto de opciones no necesitan explicación, pues son:

Submitir JCL Generado: si queremos ejecutar el JCL enviándole a la cola de procesos batch.

Editar JCL: si queremos variar por ejemplo el tamaño de los ficheros, ya que por defecto, todos los ficheros nuevos se crean como se muestra a continuación, basándose en los parámetros especificados en la línea de Parámetros de DCBs nuevas:

6 JCL generado

La siguiente figura muestra el JCL que se genera, después de haber usado la opción **S** en editar JCL

```
File Edit Edit Settings Menu Utilities Compilers Test Help
-----
ISREDDE2   RAIZ.USRJMP.DB2BATCH.JCL           Columns 00001 00072
Command ==>                               Scroll ==> CSR
000026 //KD01069 EXEC PRDB2BTC,DEF=
000027 //STEPLIB DD DSN=DSN.DSNLOAD,DISP=SHR
000028 //EJ001E DD DSN=RAIZ.USRJMP.CADENA69.PARAM8,
000029 //          DISP=(OLD,KEEP,KEEP)
000030 //EJ002I DD DSN=RAIZ.USRJMP.CADENA69.KD01069.ESTADIS,
000031 //          DISP=(NEW,CATLG,CATLG),
000032 //          LRECL=133,
000033 //          RECFM=FBA,
000034 //          SPACE=(TRK,(100,010),RLSE)
- - - - - 10 Line(s) not Displayed
000045 //EJ416S DD DSN=RAIZ.USRJMP.CADENA69.KD01069.INSERTAR,
000046 //          DISP=(NEW,CATLG,CATLG),
000047 //          LRECL=46,
000048 //          RECFM=FB,
000049 //          SPACE=(TRK,(100,010),RLSE)
000050 //SYSTSIN DD *
000051     DSN SYSTEM(SYSD)
000052     RUN PROG(KD01069) PLAN(KD01069)
000053     END
000054 /*
```

Nota: En este caso habría que submitir ejecutando el comando **submit** desde línea de comando, ya que si se hace desde el panel se perderían las modificaciones realizadas pues se genera siempre antes de submitir.

7 JCL completo:

```
//USRJMPZ JOB (DE0,MSM),
//      CLASS=A,
//      MSGCLASS=X,
//      REGION=4M,
//      NOTIFY=USRJMP
//*
//LIBPROC JCLLIB ORDER=RAIZ.PROCS
//*****
//* JCL GENERADO POR USRJMP
//* FECHA GENERACION 1998-08-12 08:34
//*****
//***-----
//*** BORRADO DE FICHEROS
//***-----
//BORRAR01 EXEC PRP0013,DEF=
//SYSIN DD *
        DEL RAIZ.USRJMP.CADENA69.KD01069.ESTADIS
        DEL RAIZ.USRJMP.CADENA69.KD01069.INFORME
        DEL RAIZ.USRJMP.CADENA69.KD01069.LISTADO
        DEL RAIZ.USRJMP.CADENA69.KD01069.INSERTAR
        SET MAXCC = 0
/*
//***-----***
//***          EJECUTAR TSO EN BATCH          ***
//***-----***
//KD01069 EXEC PRDB2BTC,DEF=
//STEPLIB DD DSN=DSN.DSNLOAD,DISP=SHR
//EJ001E DD DSN=RAIZ.USRJMP.CADENA69.PARAM8,
//      DISP=(OLD,KEEP,KEEP)
//EJ002I DD DSN=RAIZ.USRJMP.CADENA69.KD01069.ESTADIS,
//      DISP=(NEW,CATLG,CATLG),
//      LRECL=133,
//      RECFM=FBA,
//      SPACE=(TRK,(250,50),RLSE)
//EJ003S DD DSN=RAIZ.USRJMP.CADENA69.KD01069.INFORME,
//      DISP=(NEW,CATLG,CATLG),
//      LRECL=95,
//      RECFM=FB,
//      SPACE=(TRK,(250,50),RLSE)
//EJ415I DD DSN=RAIZ.USRJMP.CADENA69.KD01069.LISTADO,
//      DISP=(NEW,CATLG,CATLG),
//      LRECL=133,
//      RECFM=FB,
//      SPACE=(TRK,(250,50),RLSE)
//EJ416S DD DSN=RAIZ.USRJMP.CADENA69.KD01069.INSERTAR,
//      DISP=(NEW,CATLG,CATLG),
//      LRECL=46,
//      RECFM=FB,
//      SPACE=(TRK,(250,50),RLSE)
//SYSTSIN DD *
        DSN SYSTEM(SYSD)
        RUN PROG(KD01069) PLAN(KD01069)
        END
/*
```

8 Fichero de comandos REXX: DB2BATCH

```
/* REXX */
/*****
/* genera jcl para ejecutar un programa cobol-db2 en batch */
/*****
/** ----- Ciclo para la pantalla db2batch -----**/
/** Se pide: */
/** Nombre de los objetos a incluir en el JCL */
/*****

address ispexec "vget jclende shared"
address ispexec "vget med shared"
address ispexec "vget pri shared"
address ispexec "vget sec shared"
user = userid()
if user = USRJMP then user = USRJMP
DEBUG= "SIx"
call editar_fecha
if med = "" then med = TRK
if pri = "" then pri = 100
if sec = "" then sec = 010
cargar = N
submitir = N
editar = N
editlib = N
libtab = 'RAIZ.'user'.db2batch.ddnames'

do forever
  if jclende = "" then jclende = "RAIZ."user".db2batch.jcl"
  /*
  say jclende
  call mostrar_variables_memorizadas
  pull seguir
  */
  address ispexec " display panel(db2batch)"
  rc_panel = rc
  /* -----**/
  /* respuesta del panel db2batch correcta -----**/
  /* -----**/

  if rc_panel = 0 then do
    if editlib = S then do
      "ispexec edit dataset('"libtab"')"
      editlib = N
    end
    if cargar = S then do
      call leer_tabla
      cargar = N
    end
    else do
      call cargar_tabla_con_variables
      if debug = "SI" then do
        call mostrar_tabla
      end
      /* ----- */
      /* examinar los comandos de linea -- */
      /* ----- */
      call examinar_comandos_linea
    /*
```

```

if comando_ejecutado = "si" then do
  cargar = N
  editar = N
  submitir = N
  editlib = N
end */
/* ----- */
/* todo ok --> se prepara el JCL a submitir -- */
/* ----- */

call montar_fichas_job
PASO='BORRAR01'
call montar_paso_borrar
call montar_paso_ejecutar
PASO='BORRAR02'
call montar_paso_borrar
/* ----- */
/* pasar la cola a fichero -- */
/* ----- */

call allocar_file_jcl
call pasar_cola_a_fichero
/* ----- */
/* memorizar variables en tabla de memoria -- */
/* ----- */

call add_tabla
/* ----- */
/* submitir el jcl generado -- */
/* ----- */

if submitir = "S" then do
  address tso "submit 'jclende'"
  say "Se ha submitido el JCL " jclende
end
submitir = N
/* ----- */
/* editar el jcl generado -- */
/* ----- */

if editar = "S" then do
  "ispexec edit dataset('jclende')"
  editar = N
end
/* todo bien ---> salgo del ciclo y comienzo proceso ---*/
/* se sale del bucle pulsando F3 por eso esta anulado
leave */
end /* fin del else */
end /* fin del: if rc_panel=0 ==> la parte del then jclana01 --*/

/* respuesta del panel db2batch incorrecta -----*/
else do
  if rc_panel = 8 then do /* se dio f3 -----*/
    exit /* termina la ejecucion -----*/
  end
  else do /* otro rc = error en el panel */
    say ' error al visualizar el panel DB2BATCH. RC:' RC
    exit
  end
end
end /* fin del: if rc_panel=0 de la pantalla db2batch ----- */
end /* fin del: do forever de la pantalla db2batch -----*/

return /* para volver al programa llamante */

```

```

/* ***** */
/* ***          rutinas          *** */
/* ***** */
montar_fichas_job:

queue '//USER'Z JOB (DE0,MSM),          '
queue '//          CLASS=A,            '
queue '//          MSGCLASS=X,         '
queue '//          REGION=4M,         '
queue '//          NOTIFY='USER'      '
queue '//*                               '
queue '//LIBPROC JCLLIB ORDER=RAIZ.PROCS'
queue '//*****'
queue '//* jcl generado por ' user
queue '//* fecha generacion ' fecha_mod
queue '//*****'

return
/* ***** */
montar_paso_borrar:

if PASO = 'BORRAR02' then do
    return
end

existe = "NO"
do i = 1 to index
    upper disp.i
    if disp.i == 'NEW' then existe = "SI"
end
if existe == "NO" then return

queue '//***-----'
queue '//*** BORRADO DE FICHEROS'
queue '//***-----'
queue '//PASO' EXEC PRP0013,DEF=
queue '//SYSIN DD *
do i = 1 to index
    upper disp.i
    if disp.i == 'NEW' then do
        queue '          DEL 'dsn.i'
    end
end
end
queue '          SET MAXCC = 0
queue '/*

return
/* ***** */
montar_paso_ejecutar:

queue '//***-----***'
queue '//***          ejecutar TSO en batch          ***'
queue '//***-----***'
queue '//pgmdb2 'EXEC PRDB2BTC,DEF=
queue '//STEPLIB DD DSN=DSN.DSNLOAD,disp=shr

do i = 1 to index
    upper disp.i
    if strip(dsn.i) ^= " " then do

```

```

        queue '//ddname.i' DD DSN='dsn.i',
        if disp.i = "SHR" then do
            queue '//          disp=SHR'
        end
        if disp.i = "OLD" then do
            queue '//          disp=(old,keep,keep) '
        end
        if disp.i = "NEW" then do
            queue '//          DISP=(NEW,CATLG,CATLG), '
            queue '//          LRECL='lrecl.i',
            queue '//          RECFM='recfm.i',
            queue '//          SPACE=('med',('pri','sec'),RLSE) '
        end
    end
end

queue '//SYSTSIN DD *
queue '    DSN SYSTEM(SYSD)
queue '    RUN PROG('pgmdb2') PLAN('pgmdb2')
queue '    END
queue '/*

return
/* ***** */
allocar_file_jcl:

x = outtrap('Off')
x = outtrap('On')
address tso "delete "jclende""
x = outtrap('Off')

ref="tracks space(1,1) lrecl(80) recfm(f b) dsorg(ps) blksize(23440)"
"alloc da("jclende") f(salida) mod "ref" reuse"

return
/* ***** */
pasarCola_a_fichero:

QUEUE 'JMP-FIN'
pull linea
j = 0
do until length(linea) = 0
    j = j + 1
    jcl.j = linea
    if debug = "SI" then do
        say j ":" linea
    end
    pull linea
    upper linea
    IF LINEA = 'JMP-FIN' THEN LEAVE
end
/*
say 'pulsar intro para seguir'
pull seguir
*/

/* escritura de la tabla en el fichero de salida */

"EXECIO * DISKW SALIDA (FINIS STEM jcl."

```

```

"FREE fi(SALIDA)"

return
/* ***** */
add_tabla:

if index = 0 then return

Tabla = pgmdb2

/* ----- */
/* examinar si existe la libreria */
/* ----- */

if sysdsn("'"libtab"'")='DATASET NOT FOUND' then do
    call crear_libreria
end

/* ----- */
/* examinar si existe la tabla */
/* ----- */

if sysdsn("'"libtab'"("tabla")') = "OK" then do
    call borrar_tabla
end

/* ----- */
/* cargar campos de cada fila */
/* ----- */

do i = 1 to index
    registro = insert(' ' , ' ' , 0,80,' ')
    registro = insert(ddname.i ,registro, 0, 8,' ')
    registro = insert(dsn.i ,registro, 8,40,' ')
    registro = insert(displ.i ,registro,48, 4,' ')
    registro = insert(lrecl.i ,registro,52, 4,' ')
    registro = insert(recfm.i ,registro,56, 4,' ')
    registro = substr(registro ,1,60)
    filas.i = registro
end

/* ----- */
/* escritura de la tabla en el fichero de salida */
/* ----- */

"alloc da("'"libtab'"("tabla")) f(tablaF) mod"
"EXECIO * DISKW tablaF (FINIS STEM filas."
"free Fi(TablaF)"
if rc = 0 then do
    say ' acabo de salvar la tabla' pgmdb2 ' ispf. rc: 'rc
    pull seguir
end
return
/* ***** */
crear_libreria:

address tso "alloc dd(isptabl) dsn("'"libtab"'") new ",
            "tracks space(100,10) dir(5) reuse lrecl(80) recfm(f,b) ",
            "blksize(23440) dsorg(po)"

```

```

if rc != 0 then do
    say ' error al crear la libreria ' libtab ' RC:' rc
    pull seguir
    return
end
say 'se ha creado la libreria' libtab
pull seguir
/*
"free dd(isptabl)"
say 'liberada la libreria ' libtab ' rc:' rc
pull seguir
*/
return
/* ***** */
crear_tabla:

address ispxec "tbcreate" tabla ,
    "NAMES(Tddname,Tdsn,Tdisp,Tlrecl,Trecfm)" ,
    "REPLACE"
if rc != 0 & rc !=4 then
do
    say ' error al crear la tabla ispf rc:'rc
    pull seguir
end

say ' Se ha creado la tabla ispf ' tabla ' RC:'rc
pull seguir

return
/* ***** */
borrar_tabla:

x = outtrap('Off')
x = outtrap('On')
delete ("'"libtab'"("tabla")'")
return

address ispxec "tbdelete" tabla
if rc != 0 then
do
    say ' error al borrar la tabla ispf ' tabla ' rc:'rc
    pull seguir
end

say ' Se ha borrado la tabla ispf ' tabla ' RC:'rc
pull seguir

return
/* ***** */
salvar_tabla:

address ispxec "tbsave" tabla
if rc != 0 then
do
    say ' error al salvar la tabla ispf ' tabla ' rc:'rc
    pull seguir
end

say ' Se ha salvado la tabla ispf ' tabla ' RC:'rc

```

```

pull seguir
exit

return
/* ***** */
leer_tabla:

tabla = pgmdb2

/* ----- */
/* examinar si existe la tabla */
/* ----- */

if sysdsn("'"libtab'"("tabla"')) ->= "OK" then do
    say 'No existe la tabla ' libtab'"("tabla")'
    return
end

/* ----- */
/* leer registros desde la tabla de parametros */
/* ----- */

address TSO
"alloc dataset('"libtab'"("tabla"')) FI(TablaI) shr"
if rc ->= 0 then do
    say 'error al alocar la tabla de salida rc alloc:' rc
    "free FI(TablaI)"
    pull seguir
    return
end

i = 0
do while i < 8
    I = I + 1
    'EXECIO 'i' DISKR TablaI(FINIS STEM linea.'
    if rc ->= 0 then linea.i = insert(' ', ' ', 0, 80, ' ')
    /* IF LENGTH(strip(linea.i)) > 0 then do */
        ddname.i = SUBSTR(linea.i, 1, 8)
        dsn.i = SUBSTR(linea.i, 9, 40)
        disp.i = SUBSTR(linea.i, 49, 4)
        lrecl.i = SUBSTR(linea.i, 53, 4)
        recfm.i = SUBSTR(linea.i, 57, 4)

        if length(ddname.i) = 0 then ddname.i = " "
        if length(dsn.i) = 0 then dsn.i = " "
        if length(disp.i) = 0 then disp.i = " "
        if length(lrecl.i) = 0 then lrecl.i = " "
        if length(recfm.i) = 0 then recfm.i = " "

        interpret 'ddname'i '= ddname.'i
        interpret 'filein'i '= dsn.'i
        interpret 'dis'i ' = disp.'i
        interpret 'lre'i ' = lrecl.'i
        interpret 'rf'i ' = recfm.'i
    /*
        say 'i' i ':' linea.i
        say ddname.i' ' dsn.i ' ' disp.i ' ' lrecl.i ' ' recfm.i
        pull seguir
    */

```

```

/*end*/
end
'FREE FI(tablaI)'
/*
say ddname1' ' filein1 ' ' dis1 ' ' lre1 ' ' rf1
say ddname2' ' filein2 ' ' dis2 ' ' lre2 ' ' rf2
say ddname3' ' filein3 ' ' dis3 ' ' lre3 ' ' rf3
say ddname4' ' filein4 ' ' dis4 ' ' lre4 ' ' rf4
say ddname5' ' filein5 ' ' dis5 ' ' lre5 ' ' rf5
say ddname6' ' filein6 ' ' dis6 ' ' lre6 ' ' rf6
say ddname7' ' filein7 ' ' dis7 ' ' lre7 ' ' rf7
say ddname8' ' filein8 ' ' dis8 ' ' lre8 ' ' rf8
*/
ADDRESS ISPEXEC "VPUT (ddname1,filein1,dis1,lre1,rf1)"
ADDRESS ISPEXEC "VPUT (ddname2,filein2,dis2,lre2,rf2)"
ADDRESS ISPEXEC "VPUT (ddname3,filein3,dis3,lre3,rf3)"
ADDRESS ISPEXEC "VPUT (ddname4,filein4,dis4,lre4,rf4)"
ADDRESS ISPEXEC "VPUT (ddname5,filein5,dis5,lre5,rf5)"
ADDRESS ISPEXEC "VPUT (ddname6,filein6,dis6,lre6,rf6)"
ADDRESS ISPEXEC "VPUT (ddname7,filein7,dis7,lre7,rf7)"
ADDRESS ISPEXEC "VPUT (ddname8,filein8,dis8,lre8,rf8)"
/*
say ' acabo de MEMORIZAR tabla' pgmdb2 ' ispf. rc: 'rc
pull seguir
*/

return
/* ***** */
mostrar_variables_memorizadas:

say ddname1' ' filein1 ' ' dis1 ' ' lre1 ' ' rf1
say ddname2' ' filein2 ' ' dis2 ' ' lre2 ' ' rf2
say ddname3' ' filein3 ' ' dis3 ' ' lre3 ' ' rf3
say ddname4' ' filein4 ' ' dis4 ' ' lre4 ' ' rf4
say ddname5' ' filein5 ' ' dis5 ' ' lre5 ' ' rf5
say ddname6' ' filein6 ' ' dis6 ' ' lre6 ' ' rf6
say ddname7' ' filein7 ' ' dis7 ' ' lre7 ' ' rf7
say ddname8' ' filein8 ' ' dis8 ' ' lre8 ' ' rf8

say variables memorizadas
pull seguir

return
/* ***** */
cargar_tabla_con_variables:

index = 0
do i = 1 to 8
interpret 'filein.'i ' = filein'i
w_aux = left(filein.i,6)
upper w_aux
if w_aux ^= "FILEIN" & length(filein.i) ^= 0 then do
if debug = "SI" then do
SAY i ': dsn:' filein.i
end
index = index + 1
interpret 'ddname.'index '= ddname'i
interpret 'dsn.'index '= filein'i
interpret 'disp.'index '= dis'i

```

```

/*
interpret say length('lre'i)
*/

interpret 'if length(lre'i) = 0 then lre'i' = " "
interpret 'lrecl.'index  '= lre'i

/*
interpret say length('rf'i)
*/

interpret 'if length(rf'i) = 0 then rf'i' = " "
interpret 'recfm.'index  '= rf'i
interpret 'cmdlin.'index  '= c'i

end
end /* ----- fin de cargar en una tabla cada variable */

return
/* ***** */
mostrar_tabla:

    SAY 'index :' index
    do i = 1 to index
        SAY ddname.i dsn.i disp.i lrecl.i recfm.i
    end

return
/* ***** */
examinar_comandos_linea:

comando_ejecutado = 'no'
do i = 1 to index
    if (cmdlin.i ^= " ") ,
        & (dsn.i ^= " ") then do
            file_a_examinar = dsn.i
            call existe_fichero file_a_examinar
            if result ^= 98 then do
                if cmdlin.i = "B" then "ispexec browse dataset('"dsn.i"')"
                if cmdlin.i = "E" then "ispexec edit  dataset('"dsn.i"')"
                if cmdlin.i = "V" then "ispexec view  dataset('"dsn.i"')"
            end
        end
    end
end

c1 = ""
c2 = ""
c3 = ""
c4 = ""
c5 = ""
c6 = ""

return
/* ***** */
editar_fecha:

dt    = date('s')
dtc   = substr(dt,1,4) '-' substr(dt,5,2) '-' substr(dt,7,2)

times = time('l')
tms   = substr(times,1,2) ':' substr(times,4,2) '.' ,
        substr(times,7,2) '.' substr(times,10,6)

```

```

fecha_mod = substr(dtc' 'tms,1,16)

return
/* ***** */
existe_fichero:

    x = outtrap('off')
    x = outtrap('on')
    address tso "listd " file_a_examinar

    if rc == 12 | rc == 20 then do
        say ' No existe en host el fichero:' file_a_examinar
        return 98
    end

return

```

9 Panel: DB2BATCH

```

)ATTR
  _ TYPE(INPUT) INTENS(HIGH) PAD('_')
  ~ TYPE(INPUT) INTENS(HIGH) PAD('_') CAPS(ON)
  | TYPE(&TIPOS) INTENS(&VISIBLE)
  $ TYPE(OUTPUT) INTENS(HIGH)
  { TYPE(TEXT) INTENS(LOW) COLOR(TURQ) HILITE(REVERSE) SKIP(ON)
  ] TYPE(TEXT) INTENS(HIGH) COLOR(GREEN) SKIP(ON)
  [ TYPE(TEXT) INTENS(HIGH) COLOR(BLUE) SKIP(ON)
  @ TYPE(TEXT) INTENS(HIGH) COLOR(yellow) SKIP(ON)
  ? TYPE(&TIPO) INTENS(HIGH)
  } TYPE(OUTPUT) INTENS(NON ) COLOR(RED )
)BODY WIDTH(80)
%
%-----
% [JMP @Ejecutar pgm con BD2 en Batch [USUARIO: &ZUSER %
%-----
%Command ==> ZCMD
%
% [Este procedimiento submite un Job, para ejecutar un programa cobol-DB2
%
] ]Programa.....:-pgmdb2 [ Cargar ddnames memorizadas.]-Z@ (S/N)
] ]File del JCL a submitir:-jclende
] [Submitir JCL generado...-Z@(S/N) [ Editar Jcl:-Z@(S/N) [ Editar lib: -Z@(S/N)
%
% DDname DSN Disp Lrecl RecFM
% -----
] -z-ddname1 ]-filein1 ]-dis1-lre1] -rf1]
] -z-ddname2 ]-filein2 ]-dis2-lre2] -rf2]
] -z-ddname3 ]-filein3 ]-dis3-lre3] -rf3]
] -z-ddname4 ]-filein4 ]-dis4-lre4] -rf4]
] -z-ddname5 ]-filein5 ]-dis5-lre5] -rf5]
] -z-ddname6 ]-filein6 ]-dis6-lre6] -rf6]
] [Parametros DCBs nuevas:-Med@ [Cantidad Primaria:-pri@[ Secundaria: -sec@
%
% Comandos de linea:@E@V@B% Pulse @F3 %para salir o@INTRO %para continuar
)INIT
.ZVARS = '(cargar,submitir,editar,editlib,c1,c2,c3,c4,c5,c6)'
Vget (aplica,cadena)
Vget (ddname1,filein1,dis1,lre1,rf1)
Vget (ddname2,filein2,dis2,lre2,rf2)
Vget (ddname3,filein3,dis3,lre3,rf3)
Vget (ddname4,filein4,dis4,lre4,rf4)

```

```

Vget (ddname5,filein5,dis5,lre5,rf5)
Vget (ddname6,filein6,dis6,lre6,rf6)
Vget (ddname7,filein7,dis7,lre7,rf7)
Vget (ddname8,filein8,dis8,lre8,rf8)
&ZCMD = ' '
&TIPOS = 'OUTPUT'
&VISIBLE = 'NON'
&SUBSYS = 'SYSD'
.HELP = ' '
.CURSOR = pgmdb2
IF (&ZUSER = 'USRJMPP','USRJMP')
    &TIPOS = 'INPUT'
    &VISIBLE = 'HIGH'
)PROC
VER (&med,LIST,TRK,CYL,msg=jmpp001F)
VER (&jclende,NB,MSG=JMPP002)
Vput (ddname1,filein1,dis1,lre1,rf1)
Vput (ddname2,filein2,dis2,lre2,rf2)
Vput (ddname3,filein3,dis3,lre3,rf3)
Vput (ddname4,filein4,dis4,lre4,rf4)
Vput (ddname5,filein5,dis5,lre5,rf5)
Vput (ddname6,filein6,dis6,lre6,rf6)
Vput (ddname7,filein7,dis7,lre7,rf7)
Vput (ddname8,filein8,dis8,lre8,rf8)
Vput (jclende,pgmdb2) shared
Vput (med,pri,sec) shared
)END

```