SilverTech

+talento +tecnología +empleabilidad

Socio estratégico

diagonal
asociación civil

Impulsan

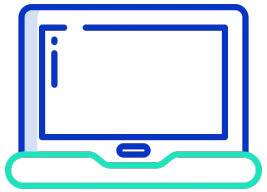
EidoS
ALWAYS LEARNING

BID | LAB

SilverTech
+talento +tecnología +empleabilidad

Cobol – Clase 28

Job Control Language - JOB



Reglas de la clase

- Micrófonos apagados
- Consultas al final de la clase
- Consultas por chat

SilverTech
+talento +tecnología +empleabilidad

Cronograma

Primera Parte

18:30
a
19:25

Break

19:25
a
19:35

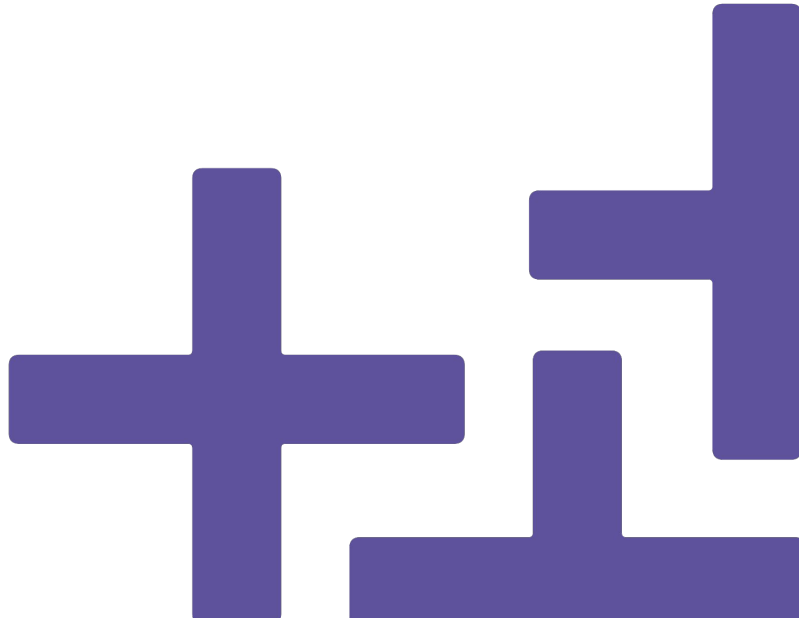
Segunda Parte

19:35
a
20:30

SilverTech
+talento +tecnología +empleabilidad

¿Qué veremos hoy?

- JOB
- Estructura del JOB
- Sentencias



SilverTech
+talento +tecnología +empleabilidad

Job control language

- **JES** (Job execution subsystem –Subsistema de ejecución de trabajos) conjunto de rutinas responsables de manejar las entradas y salidas a los Jobs es decir de todos los procesos batch.

- **SPOOL** Es el archivo que contiene las entradas y salidas de los jobs. Es un archivo que reside en varios volúmenes.

- **JOB** define las características de la ejecución, la clase de salida de los mensajes (MSGCLASS), el límite del tiempo de proceso y la opción para procesamiento diferido (HOLD).

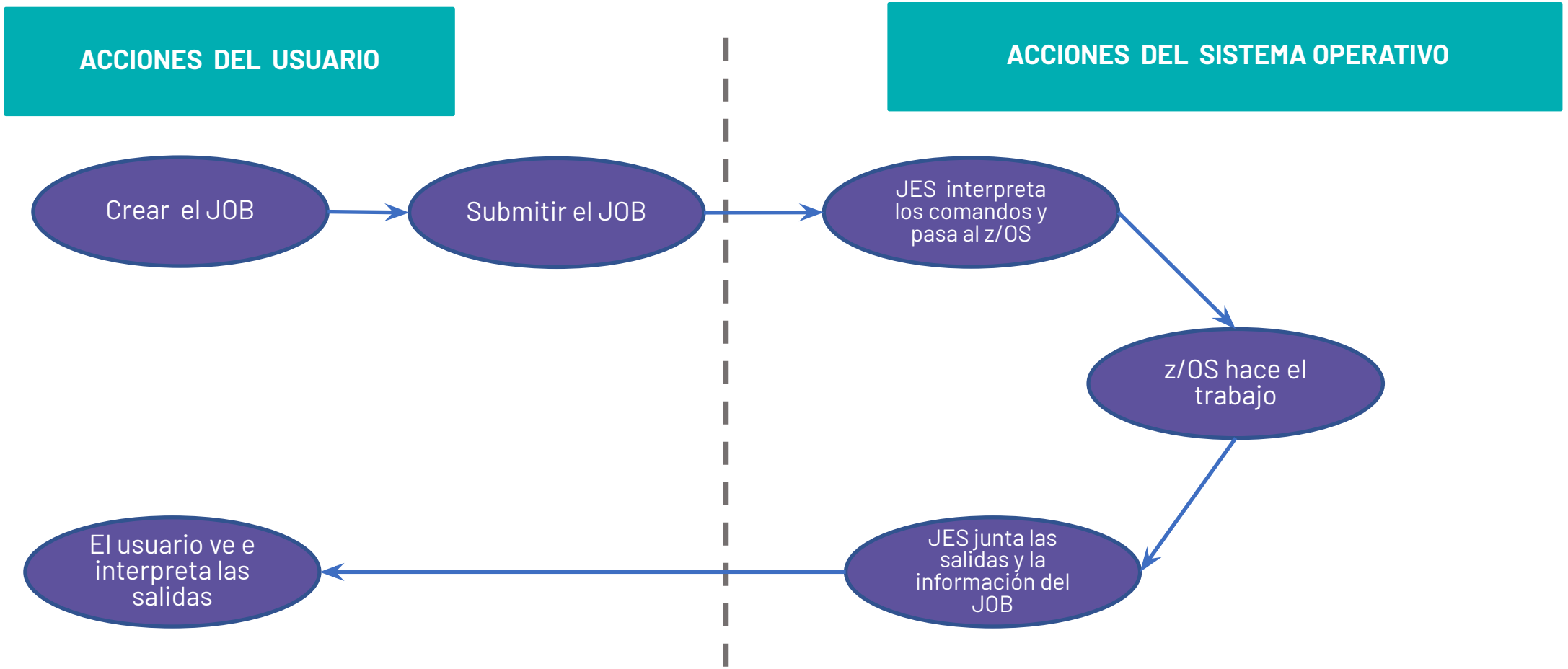
Para conseguir que el Sistema Operativo trabaje para nosotros debemos indicarle qué tiene que hacer y cuales son los recursos que necesita.

JCL es un lenguaje que comunica al Sistema Operativo :

- los programas que debe procesar
- Orden de ejecución
- Archivo de entrada y salida que se van a utilizar

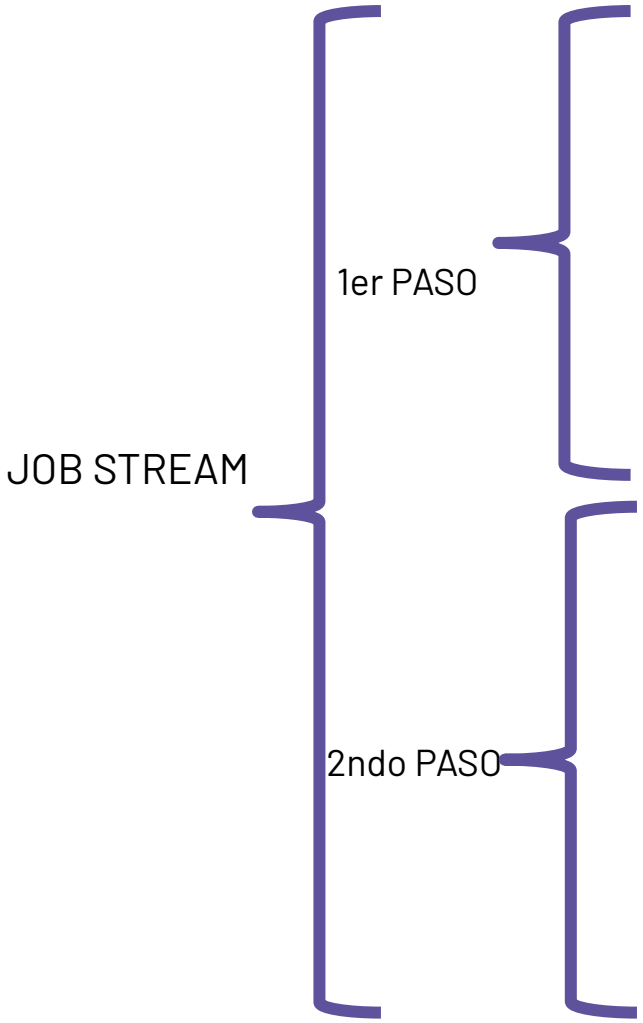
Está compuesto por sentencias que se ingresan en forma de registros de 80 posiciones.

Característica de algoritmo Corte de control



Estructura de un job

Nombre del Job: **EJEMPLO**




```
//EJEMPLO JOB
//PASO1 EXEC
//* primer paso
//ARCHIVO DD
// PASO2  EXEC
//ARCHIVO DD
//SALIDA DD
//
```

SilverTech
+talento +tecnología +empleabilidad

Características

SilverTech
+talento +tecnología +empleabilidad

- Las sentencias JCL comienzan con **//**
- Línea de comentario **// ***
- Delimitador **/***
- Las sentencias JCL finalizan cuando encuentran una línea cuyo único contenido es **//**
- A partir de la columna **72** se considera comentario.
- Todas las sentencias se deben escribir con **MAYÚSCULAS**.
- La primera sentencia es **JOB**.
- Todo Job debe tener como mínimo 1 Paso (sentencia **EXEC**).
- Todo paso debe incluir como mínimo una definición de datos (sentencia **DD**)

Tipos de Declaraciones de JCL

- **JOB** - Identifica el comienzo de un trabajo
- **EXEC** - Indica qué tipo de trabajo que se va a realizar
- **DD** - Significa Definición de Datos, es decir, identifica qué recursos se necesitan y dónde encontrarlos

SilverTech
+talento +tecnología +empleabilidad

Sentencia JOB

Coding the JOB Statement

```
// JOBNAME    JOB  [RESPONSABLE]
                   [CLASS = clase]
                   [PRTY = prioridad]
                   [TYPERUN = SCAN/HOLD]
                   [REGION = xxxx]
                   [MSGCLASS = clase]
      [MSGLEVEL = (X,X)]
```

```
//MFDEM001  JOB     'RESPONSABEL',  CLASS=A,     MSGCLASS=A
```

- Job Name
- Job Statement
- Job Info/ Programmer's Name
- Class Parameter
- Message Class Parameter

Sentencia: JOB

[RESPONSABLE]

- Posicional y opcional
- Puede tener hasta 20 caracteres
- Si se omite, no se codifica la coma indicando la ausencia
- Se utiliza para identificar a la persona o equipo responsable de la corrida

Sentencia: JOB
[CLASS = clase]

- Clase en la cual va a entrar el JOB en cola de ejecución cuando se submite
- El menor valor de clase indica mayor prioridad.
- Puede ser numérica (0 a 9) o alfanumérica (A a Z)

Aclaración:

La clase A se cancela automáticamente cuando consume 1 minuto de CPU
La clase J se cancela automáticamente cuando consume 4 minutos de CPU
La clase F no tiene restricciones de tiempo.

Sentencia: JOB

[PRTY = prioridad]

- Dentro de una misma clase se puede definir la prioridad en que se va a ejecutar el JOB
- Se ejecuta primero el de mayor prioridad
- Puede ser numérica (0 a 15)

Sentencia: JOB

[TYPERUN = SCAN/HOLD/COPY]

Puede tener 3 parámetros

- SCAN: indica que sólo se ejecutará el proceso de control de sintaxis de JCL
- HOLD: el JOB queda suspendido hasta que el operador lo desholdea.
- COPY: envía el contenido del JOB a la impresora sin ejecutar el trabajo.

Sentencia: JOB

[REGION = XXX]

- Indica la cantidad de memoria virtual que podrá utilizar el job para su ejecución. Es la memoria utilizada en todos los pasos

[NOTIFY = usuario]

- Indica a quien se debe enviar el mensaje de cómo finalizó el proceso.

Sentencia: JOB

[COND = (cod,oper)]

- Permite establecer condiciones para la no ejecución del paso, si alguna de las condiciones se cumple el paso no se ejecuta
- El parámetro tiene dos COND subparámetros: Código y el operador
- El parámetro tiene dos COND subparámetros: Código y el operador.

Código: Se trata de un decimal entre 0 y 4095

Operador: Los valores posibles son:

GT - Mayor que el código de retorno.

LT - Menos que el código de retorno.

EQ - el equivalente al código de retorno.

GE - Mayor o igual que el código de retorno.

LE - Menor o igual al código de retorno.

NE - no igual al código de retorno.

Ej: //PASO1 EXEC PGM=PRUEBA,COND=(7,LT)

Sentencia: JOB

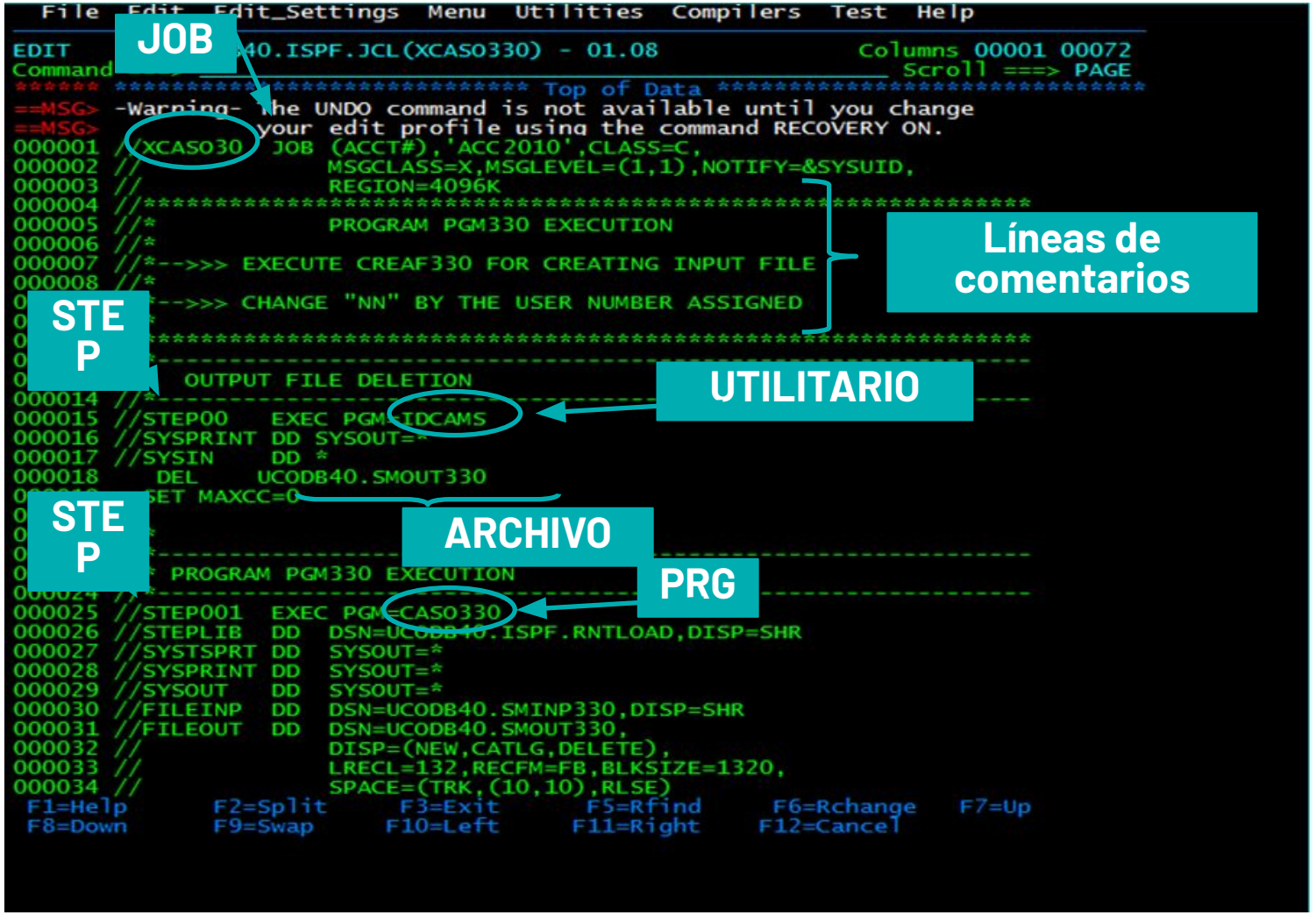
[TIME]

- Opcional. Indica el tiempo neto del procesador expresado en minutos que podrá usar el Job para su ejecución. Cuando el tiempo termina el Job cancela.
  El intervalo de tiempo permitido va entre 1 y 1440 minutos.

[RESTART]

- Se usa para re arrancar un JOB comenzando desde un paso especificado y saltando los pasos anteriores.

Ejemplo



SilverTech
+talento +tecnología +empleabilidad

Sentencia EXEC

Sentencia: EXEC

Define:

- La iniciación de un paso
- Nombre del programa o procedimiento el cual debe estar compilado y linkeado (módulo de carga ejecutable)

Permite:

- Fijar condiciones para la ejecución del paso
- Limitar el tiempo de procesador
- Limitar la memoria virtual a usar
- Pasar información al programa

Sentencia: EXEC

```
// STEPNAME   EXEC [PGM=programa] o [PROC=procedimiento ]
                [PARM='parametros', ]
                    [REGION=nnnnK, ]
                [COND=((comp,oper,step),.....)]
                        [TIME=mm,ss]
```

Se usa para pasar información variable al programa que se está ejecutando en ese job step. Para poder usar esa información el programa debe especificar instrucciones para recibirlos

EJEMPLO:

```
//PASO1    EXEC   PGM=PRGLIST1
//PASO2    EXEC   PGM=PRGLIST2,REGION=200K, PARM='A128,P=50',
//                                  COND=(4,EQ, PASO1),TIME=(20,30)
```

Sentencia: EXEC

[PRG=XXXXXXXX]

- Puede tener hasta 8 caracteres
- Indica el nombre del programa a ejecutar

- Ej: //PASO1 EXEC PGM=PRGLIST1

[PARM = 'parámetros']

- Se utiliza para pasar información al programa que se procesa (hasta 100 posiciones).

- Ej: //PASO1 EXEC PGM= PRGLIST1,
  // PARM=("027PEREZ 1993")

Sentencia: EXEC

[TIME=mm,ss]

- Indica el tiempo máximo permitido para que un paso se ejecute en minutos y segundos

- Ej: //PASO1 EXEC PGM=LISTA,TIME(20,30)

SilverTech
+talento +tecnología +empleabilidad

Sentencia DD

Sentencia : DD

SilverTech
+talento +tecnología +empleabilidad

Establece el enlace entre un nombre de DD lógico y un fichero o spool físico.

```
//DDNAME   DD      [DSN=XXXXXXXX.XXXXXXXX.XXXX]
                   [UNIT=XXXXXXXX]
                   [VOL=XXXXXXXX]
                   [DCB=XXXXXXXX]
                   [SPACE=XXXXXXXX]
                   [DISP=XXXXXXXX]
```

```
EJEMPLO:    // ARCHIVO1   DD   DSN= ARCHIVO.MAESTRO.EMPLEADO
            //                  DISP=(NEW,CATLG,DELETE),
            //                 UNIT=DISK1,
            //                 DCB=(RECFM=FB,LRECL=70,BLKSIZE=700)
```

Sentencia: DD

[DDNAME]

Nombre lógico del archivo, debe coincidir con el de la SELECT.

- Determina el nombre del DATA SET que puede ser de hasta 44 caracteres de longitud.
- Puede conformarse por calificadores de hasta 8 caracteres, separados por "."
- Si se omite, el sistema generará un nombre para ese DATA SET

Ejemplo:

```
//ARCHIVO1    DD  DSN=MAESTRO
//ARCHIVO2    DD  DSN=MAESTRO.SUELDOS
//ARCHIVO3    DD  DSN=PROD.SUE.MAESTRO.MAYO
```

Sentencia: DD

[UNIT=XXXXXX]

- Determina el tipo de dispositivo que aloja al DATA SET
- SYSDA - TAPE - DISCO

[DISP=XX,XX,XX]

- Determina cuál es la disposición actual del DATA SET y cual será al finalizar el PASO
- Se conforma con tres subparámetros posicionales:
  1 - Disposición Inicio
  2 - Disposición por terminación normal
  3 - Disposición por cancelación del paso

Sentencia: DD

[DISP=XXX,XXX,XXX]

NEW	Se creará el archivo en el paso.
OLD	El DATA SET existe al comenzar el paso
SHR	El DATA SET ya existe y podrá ser compartido por otros trabajos
MOD	El DATA SET ya existe y se grabará a continuación del último registro
DELETE	El DATA SET será eliminado al terminar el paso.
KEEP	El DATA SET será conservado al terminar el paso.
PASS	El DATA SET permanece vigente hasta la terminación del JOB, guardando sus referencias para pasos posteriores.
CATLG	El DATA SET se conserva y se cataloga
UNCATLG	El DATA SET se borra del Catálogo pero permanece en el disco

OMISION DEL PRIMER SUBPARÁMETRO → Asumirá NEW
OMISION DEL SEGUNDO SUBPARÁMETRO → Dejará el archivo en el mismo estado que al comenzar el paso
Si NEW → 'DELETE' Si OLD → 'KEEP'
OMISION DEL TERCER SUBPARÁMETRO → Asumirá el mismo valor que el 2do subparámetro.

Sentencia: DD

DISP = (NEW / OLD / SHR / MOD , DELETE / KEEP / PASS / CATLG , DELETE / KEEP / CATLG / UNCATLG)

Si no se especifica DISP asume (NEW,DELETE,DELETE)
Para crear (, CTLG.DELETE) asume NEW
Para Borrar (OLD,DELETE,KEEP)

Sentencia: DD
[DCBP=XXXXXX]

+ Define las características físicas de grabación del DATA SET.

LRECL Define la longitud del registro lógico
BLKSIZE Define la longitud del registro físico (bloque)

RECFM Especifica el formato de grabación del DATA SET.
Puede ser: 'F' Longitud fija
'FB' Longitud fija bloqueado
'V' Longitud variable
'VB' Longitud variable bloqueado
'U' Longitud indefinida

EJEMPLO:

```
//ARCH  DD DSN=FILE.UNO,DCB=(LRECL=200,BLKSIZE=2000,RECFM=FB)
//VARI  DD  DSN=DISCO,DCB=(LRECL=140,BLKSIZE=3000,RECFM=VB)
```

Sentencia: DD

[SCAPCE=XXXXXX]

+ Especifica el espacio en disco y en que forma se alocará

  - TRK → Será alocada en pistas
  - CYL → Será alocado en cilindros
  - LB → Será alocada en bloque

EJEMPLO:

```
//SORT   DD  UNIT=WORKDISK,SPACE=(CYL,(2,1))
```

[DUMMY]

+ DUMMY or DSN=NULLFILE simula la presencia de un archivo para cuando un programa lo necesita sin utilizar un Dataset

EJEMPLO:

```
//LIST       EXEC PGM=AGP0034P
//PASO1    DD  DSN=MAESTRO.CLIENTES, DISP=SHR
//LISTADO  DD DUMMY
```

Sentencia: DD

Sentencias DD especiales - JCLLIB

- Indica dónde se encuentra el programa que será ejecutado en la sentencia EXEC.

  Debe codificarse antes de la primera EXEC del JOB
  - Es aplicable a todos los pasos de un JOB
  - Se define inmediatamente después de la sentencia JOB
  - No puede ser utilizada en procedimientos

- Ej: //JOB1 JOB (E343),'ACC'
  //JOBLIB DD DSN=UCODBNN.CURSO.LOAD,DISP=SHR
  //STEP1 EXEC PGM=PROG1

Utilitarios

SilverTech
+talento +tecnología +empleabilidad

Son programas del Sistema Operativo que ayudan a la organización, seguimiento y mantenimiento de datos.

DFSORT Es un utilitario usado principalmente para ordenar registros en un archivo.

MERGE Intercala registros de dos o más archivos en orden.

IEBGENER Programa utilitario de IBM que se usa para crear,copiar o imprimir archivos secuenciales y particionados-

IDCAMS Utilitario para manejar archivos VSAM pero que se puede usar también con archivos secuenciales.

Repro Copia de secuencial a secuencial

```
Command ===> _____ Scroll ===> PAGE
=COLS> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
****** ***************************** Top of Data ******************************
==CHG> //UCODB40R JOB (101,'REPRO    '),CLASS=B,MSGCLASS=X,MSGLEVEL=(2,1),
000002 //             NOTIFY=&SYSUID
000003 /*
000004 //*******************************************************************
000005 //* DELETEA Y COPIA EL ARCHIVO DE ENTRADA AL CASO 350 (SMINP350)
000006 //*
000007 //*                            REPRO
000008 //*
==CHG> //*-->>> CHANGE "40" BY THE USER NUMBER ASSIGNED  (ALL)
000010 //*
000011 //*******************************************************************
000012 //*-----------------------------------------------------------------
000013 //*  OUTPUT FILE DELETION
000014 //*-----------------------------------------------------------------
000015 //STEP00   EXEC PGM=IDCAMS
000016 //SYSPRINT DD SYSOUT=*
000017 //SYSIN    DD *
==CHG>    DEL    UCODB40.SMINP350
000019   SET MAXCC=0
000020 /*
000021 //*-----------------------------------------------------------------
000022 //*  COPY MASTER FILE TO USER FILE
000023 //*-----------------------------------------------------------------
000024 //STEP01   EXEC PGM=IDCAMS
000025 //SYSPRINT DD SYSOUT=*
000026 //FILE1    DD DSN=MCOD.PBM1.BAT1CONS.SMINP350,DISP=SHR
==CHG> //FILE2    DD DSN=UCODB40.SMINP350,
000028 //            DISP=(NEW,CATLG,DELETE),
000029 //            LRECL=94,RECFM=FB,BLKSIZE=9400,
000030 //            SPACE=(TRK,(5,5),RLSE)
000031 //SYSIN    DD *
000032    REPRO   INFILE  (FILE1)   -
000033            OUTFILE (FILE2)
****** ***************************** Bottom of Data ***************************

 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

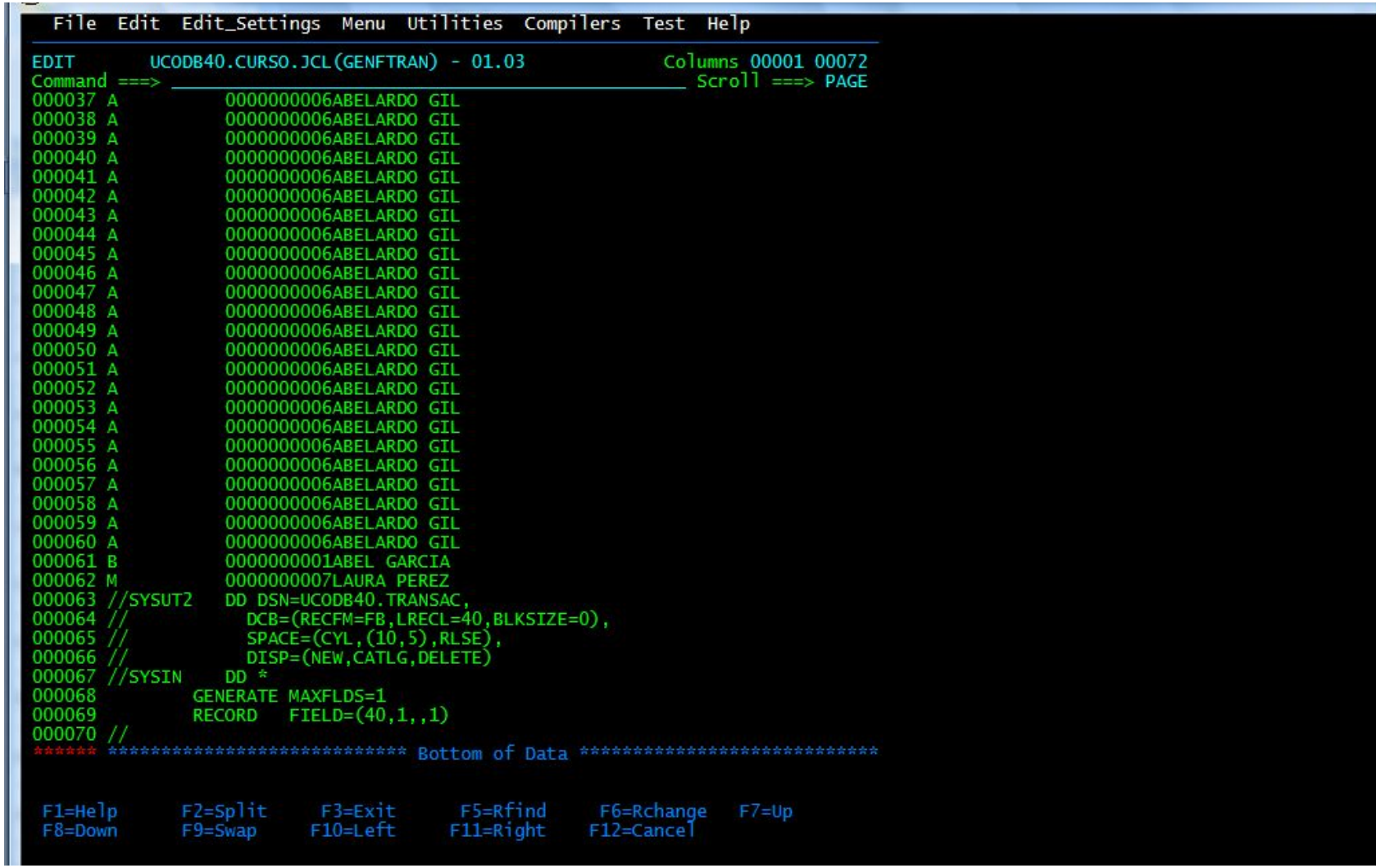
Carga Datos por parámetro en un Secuencial (1 de 2)

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       UCODB40.CURSO.JCL(GENFTRAN) - 01.03            Member GENFTRAN saved
Command ===>                                               Scroll ===> PAGE
****** ***************************** Top of Data ******************************
000001 //UCODB40T JOB (101,'GENERATE'),CLASS=B,MSGCLASS=X,MSGLEVEL=(2,1),
000002 //             NOTIFY=&SYSUID
000003 /*
000004 //*****************************************************************
000005 /*
000006 //*   CARGA LOS DATOS DE ENTRADA AL SORT EN UCODB40.TRANSAC
000007 /*
000008 //*-->>> CHANGE "40" BY THE USER NUMBER ASSIGNED  (ALL)
000009 //*****************************************************************
000010 //*---------------------------------------------------------------
000011 //*  OUTPUT FILE DELETION
000012 //*---------------------------------------------------------------
000013 //P0001 EXEC PGM=IDCAMS
000014 //SYSPRINT DD SYSOUT=*
000015 //SYSIN DD *
000016    DEL    UCODB40.TRANSAC
000017   SET MAXCC=0
000018 /*
000019 //*---------------------------------------------------------------
000020 //*  TRANSACTIONS FILE GENERATION
000021 //*---------------------------------------------------------------
000022 //STEP1    EXEC PGM=IEBGENER,
000023 //SYSPRINT DD SYSOUT=*
000024 //SYSUT1   DD *
000025 A          0000000001ABEL GARCIA
000026 A          0000000002ZACARIAS LOPEZ
000027 A          0000000003BALDOMERO GOMEZ
000028 A          0000000004EDUARDO ALVAREZ
000029 M          0000000001LUCIA ANTON
000030 M          0000000006ABELARDO GIL
000031 B          0000000006ABELARDO GIL
000032 B          0000000001ABEL GARCIA
000033 A          0000000006ABELARDO GIL
000034 J          0000000008VALENTIN MARIN
000035 A          0000000006ABELARDO GIL
000036 A          0000000006ABELARDO GIL
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Se fuerza el código 0 por si el archivo que voy a borrar no existe y el código de error resultante es distinto de 0

SilverTech
+talento +tecnología +empleabilidad

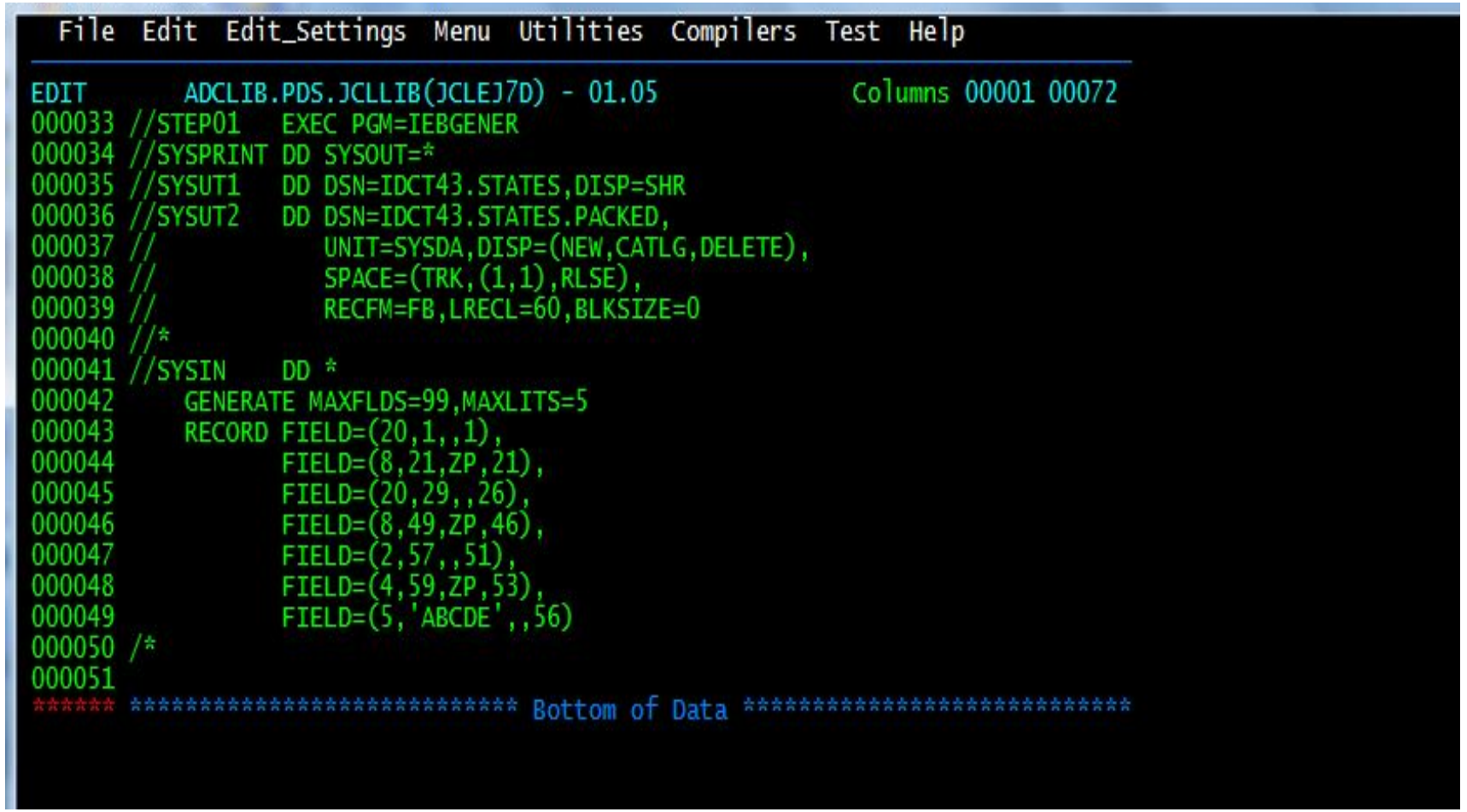
Carga Datos por parámetro en un Secuencial (2 de 2)



Ejemplo de SORT

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       UCODB40.CURSO.JCL(SORTTRAN) - 01.01            Columns 00001 00072
Command ===> _                                             Scroll ===> PAGE
****** ******************************** Top of Data ********************************
000001 //UCODB40S JOB (101,'SORT    '),CLASS=B,MSGCLASS=X,MSGLEVEL=(2,1),
000002 //             NOTIFY=&SYSUID
000003 /*
000004 //*********************************************************************
000005 //*                SORT DE UCODB40.TRANSAC
000006 /*
000007 //*             ORDENA POR INDICATOR Y KEY-TRAN
000008 //*********************************************************************
000009 //* IT WILL BE INPUT FOR ACTUALNN PROGRAM
000010 //*-->>> CHANGE "NN" BY THE USER NUMBER ASSIGNED  (ALL)
000011 //*********************************************************************
000012 //*-------------------------------------------------------------------
000013 //*  SORT TRANSAC FILE GENERATED IN JOB GENFTRAN
000014 //*-------------------------------------------------------------------
000015 //STEP1    EXEC PGM=SORT
000016 //SYSOUT   DD SYSOUT=*
000017 //SORTIN   DD DSN=UCODB40.TRANSAC,
000018 //            DISP=OLD
000019 //SORTOUT  DD DSN=UCODB40.TRANSAC,DISP=OLD
000020 //SYSIN    DD *
000021          SORT FIELDS=(1,1,CH,A,2,20,CH,A)
000022 //SORWK01 DD UNIT=SYSALLDA,SPACE=(CYL,(10,5),RLSE)
000023 //
****** ***************************** Bottom of Data ******************************
```

Ejemplo de IEBGENER



Concatenación de Archivos

- Cuando se desea procesar varios Data Set en una misma DD se utiliza la concatenación de DDs y permite tratar varios archivos físicos como si fuera un único archivo lógico.

```
//ARCHIVO  DD  DSN=ARC.UNO,UNIT=DISCO1
//                 DISP=OLD
//             DD  DSN=ARC.DOS,UNIT=DISCO2
//          DISP=SHR, VOL=SECDISCO2
//          DD   *
/*
```

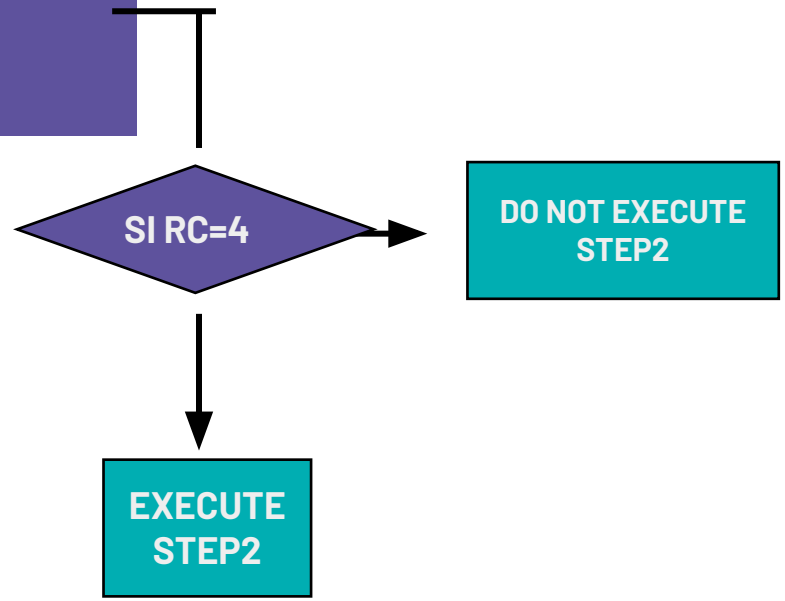
Parámetro de condición

- COND O IF/ELSE permite establecer condiciones para que un PASO se ejecute o no.

run STEP01
IF STEP01's **return** code **is greater than or equal to** 4 **THEN**
run STEP02
END IF
IF any previous return code **is less than** 8 **THEN**
run STEP03
END IF
IF STEP01 abnormally ended **THEN**
run STEP04
END IF

CONDitional Execution

```
//STEP1 EXEC PGM=PROGRAM1
//STEP2 EXEC PGM=PROGRAM2,COND=(4,EQ,PASO1)
//STEP3 EXEC PGM=PROGRAM3
```



IF/THEN/ELSE/ENDIF Statements

- Ejemplo:

```
// IF RC >= 8 THEN
//PASO1   EXEC PGM=AR5340
//SYSOUT   DD   SYSOUT=*
//ERRLOG   DD   DSNAME=MMA2.ERRLOG,DISP=MOD
// ELSE
// PASO2   EXEC PGM=AR5350
//SYSOUT    DD   SYSOUT=*
//TRANFILE DD   DSNAME=MMA2.TRANFILE,DISP=SHR
//ENDIF
```

SilverTech
+talento +tecnología +empleabilidad

Comunicación

Foro de consultas TEC:
https://campus.soysilvertech.org

Mails de consulta TEC:
consultasCOBOL@soysilvertech.org

GRACIAS

SilverTech
+talento +tecnología +empleabilidad

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

SilverTech

+talento +tecnología +empleabilidad