

# Tecnologías Mainframe

Cobol-JCL-Utilitarios

# JCL (Job Control Language)

- **Función:** Es un lenguaje utilizado para definir y controlar trabajos (jobs) en entornos mainframe. Uno de ellos es COBOL.
- **Uso:** Se utiliza para especificar qué programas y utilitarios se ejecutarán, así como para definir los archivos de entrada/salida y los parámetros de ejecución.
- **Características:** Controla el flujo de ejecución, permite la ejecución secuencial de pasos y proporciona opciones de control y notificación.

# IMPLEMENTACION COBOL CON JCL

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. InputCOBOL.  
  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 NAME-IN PIC X(20).  
01 AGE-IN PIC 99.  
  
PROCEDURE DIVISION.  
    DISPLAY 'Enter your name: '  
    ACCEPT NAME-IN  
    DISPLAY 'Enter your age: '  
    ACCEPT AGE-IN  
  
    DISPLAY 'Hello, ' NAME-IN '!'  
    DISPLAY 'Your age is: ' AGE-IN  
  
    STOP RUN.
```

```
//JOBNAME    JOB (ACCT), 'Job description',CLASS=class,MSGCLASS=class  
//STEP1      EXEC PGM=YOURCOBOLCOMPILER  
//STEPLIB    DD DSN=YOUR.COMPIER.LIBRARY,DISP=SHR  
//SYSOUT     DD SYSOUT=*  
//SYSUDUMP   DD SYSOUT=*  
//SYSIN      DD *  
             //COBOL.SYSIN DD *  
IDENTIFICATION DIVISION.  
PROGRAM-ID. InputCOBOL.  
  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 NAME-IN PIC X(20).  
01 AGE-IN PIC 99.  
  
PROCEDURE DIVISION.  
    DISPLAY 'Enter your name: '  
    ACCEPT NAME-IN  
    DISPLAY 'Enter your age: '  
    ACCEPT AGE-IN  
  
    DISPLAY 'Hello, ' NAME-IN '!'  
    DISPLAY 'Your age is: ' AGE-IN  
  
    STOP RUN.  
/*
```

# COBOL

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. InputCOBOL.  
  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 NAME-IN PIC X(20).  
01 AGE-IN PIC 99.  
  
PROCEDURE DIVISION.  
    DISPLAY 'Enter your name: '  
    ACCEPT NAME-IN  
    DISPLAY 'Enter your age: '  
    ACCEPT AGE-IN  
  
    DISPLAY 'Hello, ' NAME-IN '!'  
    DISPLAY 'Your age is: ' AGE-IN  
  
    STOP RUN.
```

- **IDENTIFICATION DIVISION.:** Esta línea indica el comienzo de la sección de identificación del programa COBOL.
- **PROGRAM-ID. InputCOBOL.:** Define el nombre del programa COBOL como "InputCOBOL".
- **DATA DIVISION.:** Esta línea indica el comienzo de la sección de datos del programa COBOL.
- **WORKING-STORAGE SECTION.:** Define la sección de almacenamiento de trabajo, donde se declaran las variables utilizadas en el programa.
- **01 NAME-IN PIC X(20).:** Declara la variable NAME-IN como un campo alfanumérico de longitud máxima de 20 caracteres. Esta variable se utilizará para almacenar el nombre ingresado por el usuario.
- **01 AGE-IN PIC 99.:** Declara la variable AGE-IN como un campo numérico de dos dígitos. Esta variable se utilizará para almacenar la edad ingresada por el usuario.
- **PROCEDURE DIVISION.:** Esta línea indica el comienzo de la sección de procedimientos del programa COBOL.
- **DISPLAY 'Enter your name: ':** Muestra un mensaje en la salida estándar solicitando al usuario que ingrese su nombre.
- **ACCEPT NAME-IN:** Lee el valor ingresado por el usuario y lo asigna a la variable NAME-IN.
- **DISPLAY 'Enter your age: ':** Muestra un mensaje en la salida estándar solicitando al usuario que ingrese su edad.
- **ACCEPT AGE-IN:** Lee el valor ingresado por el usuario y lo asigna a la variable AGE-IN.
- **DISPLAY 'Hello, ' NAME-IN '!':** Muestra un mensaje de saludo que incluye el nombre ingresado por el usuario.
- **DISPLAY 'Your age is: ' AGE-IN:** Muestra un mensaje que indica la edad ingresada por el usuario.
- **STOP RUN.:** Finaliza la ejecución del programa.

# JCL

```
//JOBNAME JOB (ACCT), 'Job description',CLASS=class,MSGCLASS=class
```

**JOBNAME:** Es el nombre del trabajo JCL. Puedes reemplazarlo con un nombre significativo para tu trabajo.

**(ACCT):** Especifica la cuenta asociada al trabajo.

**'Job description':** Es una descripción opcional del trabajo.

**CLASS=class:** Especifica la clase de prioridad del trabajo.

**MSGCLASS=class:** Especifica la clase de salida para los mensajes del trabajo.

```
//STEP1 EXEC PGM=YOURCOBOLCOMPILER  
//STEPLIB DD DSN=YOUR.COMPIER.LIBRARY,DISP=SHR
```

**STEP1:** Es el nombre del paso en el trabajo JCL. Puedes asignarle un nombre significativo.

**EXEC PGM=YOURCOBOLCOMPILER:** Especifica el programa ejecutable que se debe ejecutar en este paso. Debes reemplazar "YOURCOBOLCOMPILER" con el nombre del compilador COBOL que estés utilizando en tu entorno.

**STEPLIB DD DSN=YOUR.COMPIER.LIBRARY,DISP=SHR:** Especifica la biblioteca donde se encuentra el compilador COBOL. Debes reemplazar "YOUR.COMPIER.LIBRARY" con la ubicación de la biblioteca en tu entorno.

# JCL

```
//SYSOUT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*
```

```
//SYSIN DD *  
//COBOL.SYSIN DD *  
IDENTIFICATION DIVISION.  
PROGRAM-ID. InputCOBOL.  
  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 NAME-IN PIC X(20).  
01 AGE-IN PIC 99.  
  
PROCEDURE DIVISION.  
    DISPLAY 'Enter your name: '  
    ACCEPT NAME-IN  
    DISPLAY 'Enter your age: '  
    ACCEPT AGE-IN  
  
    DISPLAY 'Hello, ' NAME-IN '!'  
    DISPLAY 'Your age is: ' AGE-IN  
  
    STOP RUN.  
/*
```

**SYSOUT DD SYSOUT=\***: Define la salida del sistema para los registros y mensajes generados durante la ejecución del trabajo JCL. En este caso, se envía a la salida estándar del sistema.

**SYSUDUMP DD SYSOUT=\***: Define la salida del sistema para los registros de volcado de error durante la ejecución del trabajo JCL. En este caso, también se envía a la salida estándar del sistema.

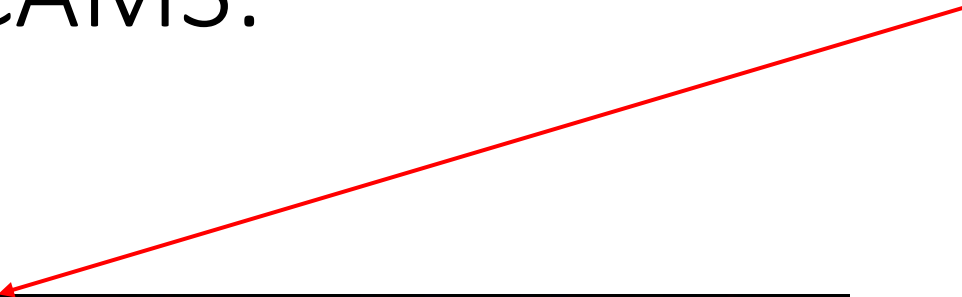
**SYSIN DD \***: Define la entrada de datos para el programa COBOL. En este caso, el código COBOL se proporciona directamente en el JCL utilizando el bloque `/* ... */`.

**//COBOL.SYSIN DD \***: Especifica un nombre lógico para el archivo de entrada del programa COBOL. Puede ser utilizado por el programa COBOL para acceder a los datos.

# IDCAMS:

- **Función:** Es un utilitario utilizado para el manejo y administración de archivos y conjuntos de datos.
- **Uso:** Se utiliza para realizar tareas como creación, eliminación, copia, renombrado, reorganización y consulta de archivos y conjuntos de datos.
- **Características:** Proporciona comandos para realizar operaciones específicas en conjuntos de datos, como eliminación, creación, copia y modificación.

# IDCAMS:



```
//IDCAMSJOB JOB (ACCTINFO), 'Job Name', NOTIFY=&SYSUID  
//IDCAMSSTEP EXEC PGM=IDCAMS  
//SYSOUT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
DELETE YOUR.DATASET.NAME  
SET MAXCC=0  
/*
```

En este ejemplo de JCL, se define un trabajo (JOB) llamado **IDCAMSJOB**. El paso de ejecución (STEP) específico para el utilitario IDCAMS se llama IDCAMSSTEP.

- La instrucción **EXEC PGM=IDCAMS** indica que se debe ejecutar el utilitario IDCAMS.
- El **DD statement SYSOUT** se utiliza para dirigir la salida del sistema.
- El **DD statement SYSPRINT** se utiliza para dirigir la salida específica del utilitario IDCAMS.
- El **DD statement SYSIN** se utiliza para proporcionar los comandos y parámetros específicos del IDCAMS. En este caso, se utilizan los comandos DELETE y SET MAXCC.
- **DELETE YOUR.DATASET.NAME** indica que se debe eliminar el conjunto de datos especificado. Debes reemplazar "YOUR.DATASET.NAME" con el nombre del conjunto de datos que deseas eliminar.
- **SET MAXCC=0** se utiliza para establecer el código de retorno máximo como 0, lo que significa que se debe ignorar cualquier error durante la ejecución del IDCAMS.

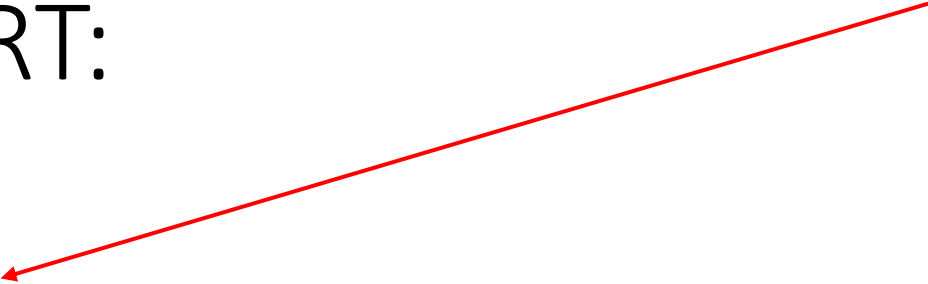
Después de definir el JCL con las instrucciones y parámetros necesarios, puedes enviar el trabajo al sistema operativo para su ejecución.



# SORT:

- **Función:** Es un utilitario utilizado para ordenar, combinar y manipular datos de archivos.
- **Uso:** Se utiliza para realizar operaciones como ordenar datos, fusionar registros, realizar cálculos y resúmenes, filtrar registros y copiar/reorganizar registros.
- **Características:** Permite especificar campos clave para ordenar, realizar operaciones de clasificación y manipulación de datos eficientes, y proporciona una amplia gama de opciones y parámetros.

# SORT:



```
//SORTJOB JOB (ACCTINFO), 'Job Name', NOTIFY=&SYSUID
//SORTSTEP EXEC PGM=SORT
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=your.input.file,DISP=SHR
//SORTOUT DD DSN=your.output.file,DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(10,10),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//SYSIN DD *
        SORT FIELDS=(1,10,CH,A)
/*
```

En este ejemplo de JCL, se define un trabajo (JOB) llamado **SORTJOB**. El paso de ejecución (STEP) específico para el utilitario SORT se llama **SORTSTEP**.

- La instrucción **EXEC PGM=SORT** indica que se debe ejecutar el utilitario SORT.
- El **DD statement SYSOUT** se utiliza para dirigir la salida del sistema.
- El **DD statement SORTIN** especifica el archivo de entrada utilizado por el utilitario SORT. Debes reemplazar "your.input.file" con el nombre del archivo de entrada que desees utilizar.
- El **DD statement SORTOUT** especifica el archivo de salida donde se guardarán los resultados del SORT. Debes reemplazar "your.output.file" con el nombre del archivo de salida deseado. En este ejemplo, se crea un nuevo archivo de salida con opciones de espacio y bloqueo definidas.
- El **DD statement SYSIN** se utiliza para proporcionar las instrucciones y parámetros específicos del SORT. En este caso, se utiliza la opción SORT FIELDS para especificar los campos que se utilizarán para ordenar los datos. En el ejemplo, se ordenan los primeros 10 caracteres de cada registro en orden ascendente.

Después de definir el JCL con las instrucciones y parámetros necesarios, puedes enviar el trabajo al sistema operativo para su ejecución.

# EIBGENER:

- **Función:** Es un utilitario utilizado para generar y manipular conjuntos de datos de Entrada/Salida en JCL.
- **Uso:** Se utiliza para realizar tareas como la creación de conjuntos de datos, la copia de conjuntos de datos, la conversión de formatos y la generación de datos de prueba o muestra.
- **Características:** Permite crear, copiar y manipular conjuntos de datos, así como convertir entre diferentes formatos de registro y generar datos de prueba o muestra.

# EIBGENER:

```
//JOBNAME    JOB (ACCT), 'Job description',CLASS=class,MSGCLASS=class
//STEP1      EXEC PGM=EIBGENER
//SYSPRINT   DD SYSOUT=*
//SYSUT1     DD DSN=input.dataset,DISP=SHR
//SYSUT2     DD DSN=output.dataset,
//           DISP=(NEW,CATLG,DELETE),
//           SPACE=(CYL,(5,5),RLSE),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//SYSIN      DD *
            COPY INDD=SYSUT1,OUTDD=SYSUT2
/*
```

- **JOBNAME** es el nombre del trabajo JCL.
- **STEP1** es el nombre del paso.
- **PGM=EIBGENER** especifica que se debe utilizar el programa EIBGENER.
- **SYSPRINT es el DD (Data Definition)** para la salida del registro de mensajes del programa.
- **SYSUT1 es el DD** que especifica el archivo de entrada (en este caso, "input.dataset").
- **SYSUT2 es el DD** que especifica el archivo de salida (en este caso, "output.dataset"). El archivo se crea como nuevo, se cataloga y se eliminará después de su uso.
- **SPACE=(CYL,(5,5),RLSE)** especifica el espacio en cilindros que se asigna para el archivo de salida.
- **DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)** define las características del registro del archivo de salida, en este caso, un formato de registro fijo (FB) con una longitud de registro de 80 bytes.
- **SYSIN es el DD** que contiene las instrucciones para el programa EIBGENER. En este caso, se utiliza la instrucción COPY para copiar los datos de entrada al archivo de salida.

# VSAM (Virtual Storage Access Method)

- **Funciones:**

- Organización y acceso eficiente de datos.
- Gestión de registros dentro de los conjuntos de datos.
- Control de concurrencia para mantener la integridad de los datos.

- **Usos:**

- Almacenamiento de datos transaccionales en entornos mainframe.
- Creación de archivos de índices para búsquedas rápidas.
- Subyacente en sistemas de bases de datos mainframe.

- **Características:**

- Alto rendimiento para un acceso rápido a los datos.
- Control de integridad y administración de bloqueos.
- Opciones de recuperación y disponibilidad.
- Utilidades específicas como IDCAMS para tareas de administración.

# VSAM

- **Definición del clúster VSAM:** Se define un clúster VSAM llamado 'CLUSTER.NAME' con organización INDEXED (KSDS). El clúster tiene claves primarias de 10 bytes de longitud y no tiene claves secundarias. Cada registro tiene una longitud de 80 bytes. Se asigna espacio libre de 10 cilindros al clúster y se especifica un tamaño de CI (control interval) de 4096 bytes y un tamaño de buffer de 4096 bytes.
- **Creación del clúster VSAM:** Se utiliza el comando REPRO de IDCAMS para copiar los datos del archivo de entrada 'INPUT.FILE' al clúster VSAM 'CLUSTER.NAME'.

```
DEFINE CLUSTER (NAME('CLUSTER.NAME') -  
              INDEXED -  
              KEYS(10 0) -  
              RECORDSIZE(80 80) -  
              FREESPACE(10 10) -  
              CISZ(4096) -  
              BUFFERSPACE(4096))
```

```
IDCAMS SYSIN  
REPRO -  
  INFILE('INPUT.FILE') -  
  OUTFILE('CLUSTER.NAME')
```

# VSAM

- **Acceso y manipulación de datos VSAM:** Una vez que el clúster VSAM ha sido creado y cargado con datos, puedes acceder y manipular los datos utilizando programas COBOL u otros lenguajes de programación compatibles con VSAM.
- **Se declara un archivo VSAM llamado VSAM-FILE** y un registro **VSAM-RECORD** con los campos correspondientes. Se abre el archivo VSAM en modo de lectura y se realiza una operación de lectura. Si hay registros disponibles, se muestra el contenido de los campos FIELD1 y FIELD2. Cuando no hay más registros, se muestra el mensaje **"No more records"**. Luego, se cierra el archivo VSAM y el programa termina.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. READ_VSAM.  
  
DATA DIVISION.  
FILE SECTION.  
FD VSAM-FILE.  
01 VSAM-RECORD.  
   05 FIELD1 PIC X(10).  
   05 FIELD2 PIC X(20).  
   ...  
  
WORKING-STORAGE SECTION.  
01 WS-VSAM-STATUS PIC X(2).  
...  
PROCEDURE DIVISION.  
MAIN.  
   OPEN INPUT VSAM-FILE.  
   READ VSAM-FILE  
   AT END DISPLAY "No more records"  
   NOT AT END DISPLAY "Record: " FIELD1 ", " FIELD2  
   END-READ.  
   CLOSE VSAM-FILE.  
   STOP RUN.
```