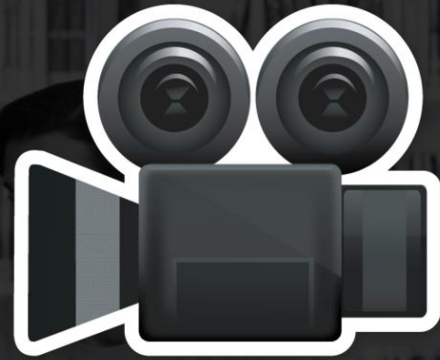


Bootcamp Cobol

Introducción a la programación cobol - Parte 1

**Recuerda
poner a grabar
la clase.**



Confección de un programa COBOL

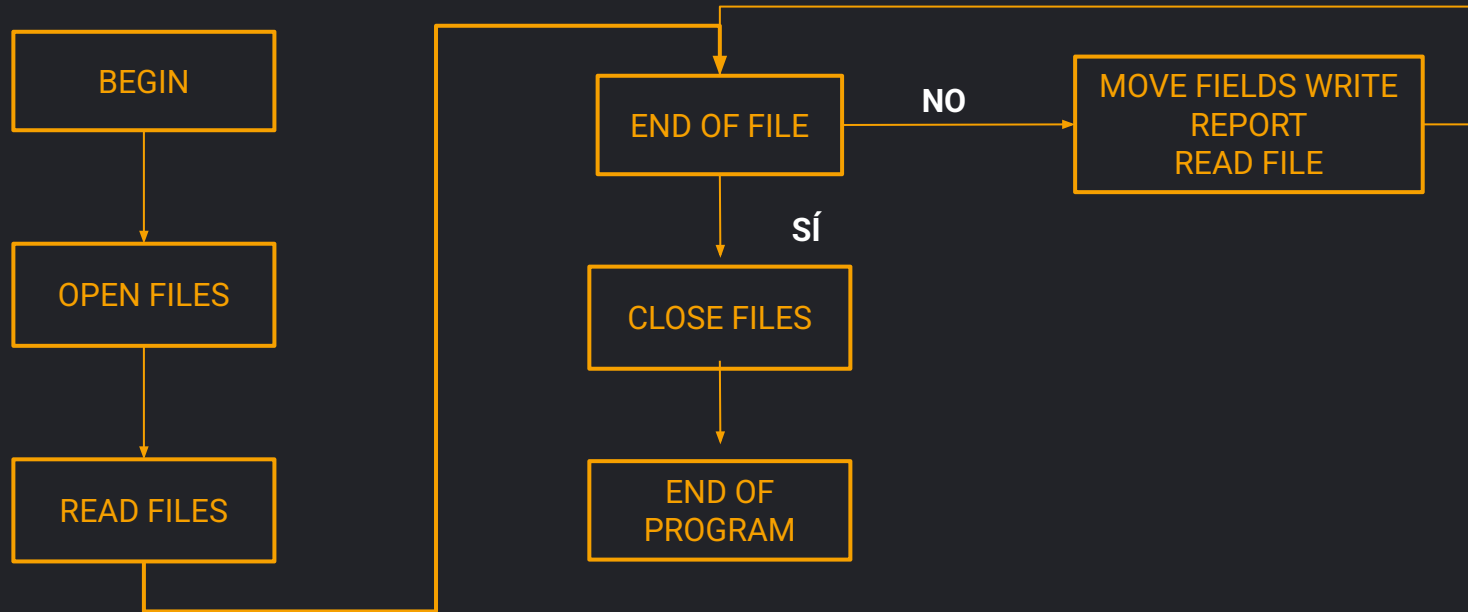


... ¿Que es Cobol?

COBOL (COMMON BUSINESS ORIENTED LANGUAGE)

- Es uno de los lenguajes más usados en el mundo en los negocios comerciales y financieros.
- Es considerado el primer lenguaje orientado a este tipo de negocios, y en la actualidad es el lenguaje más usado en el MAINFRAME.
- Su objetivo es resolver problemas de negocio en forma fácil, soportando las normas ISO (International Standards Organization)

... Ejemplo de cómo pasamos de un PSEUDOCÓDIGO a COBOL





... Ejemplo de un PSEUDOCÓDIGO como solución.

Open entrada archivo personal y salida archivo impresión

Read archivo personal, al finalizar, coloque una marca de fin

Perform until fin

- Move campos de entrada a campos de salida
- Write registro salida from campos de salida
- Read archivo personal at end coloque marca de fin

End-perform

Close entrada archivo personal y salida archivo impreso

... Ejemplo del programa COBOL desde el PSEUDOCÓDIGO como solución.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. PERSRPT.  
ENVIRONMENT DIVISION.  
INPUT-OUTPUT SECTION.  
FILE CONTROL.  
        SELECT      PERSONAL  ASSIGN      DDPERS.  
        SELECT      REPORTE   ASSIGN      DDRPT.  
DATA DIVISION.  
FILE SECTION  
FD      PERSONAL  
        RECORDING MODE IS F  
        BLOCK CONTAINS 0 RECORDS.  
01      REGISTRO-PERSONAL                                PIC X(80).  
FD      REPORTE  
        RECORDING MODE IS F  
        BLOCK CONTAINS 0 RECORDS.  
01      REGISTRO-REPORTE                                PIC X(80).
```

... Ejemplo del programa COBOL desde el PSEUDOCÓDIGO como solución.

WORKING STORAGE SECTION.

```
01  PERSONAL-LAYOUT
    02  NUM-EMPLEADO          PIC 9(5).
    02  FECHA-ALTA
        03  AA-ALTA          PIC 9(4)  VALUE ZERO.
        03  MM-ALTA          PIC 99    VALUE ZERO.
        03  DD-ALTA          PIC 99    VALUE ZERO.
    02  FILLER                PIC X(67).
01  FLAG-FIN                  PIC XX    VALUE SPACES.
    88  END-OF-FIN            VALUE 'SI'.
    88  NOT-END-OF-FIN        VALUE 'NO'.
```




... Ejemplo del programa COBOL desde el PSEUDOCÓDIGO como solución.

SOLUCIÓN 1 DE CÓMO ENFOCAR LA PROCEDURE DIVISION

PROCEDURE DIVISION.

OPEN INPUT PERSONAL.

OPEN OUTPUT REPORTE.

READ PERSONAL INTO PERSONAL-LAYOUT AT END SET END-OF-FIN TO TRUE

PERFORM UNTIL END-OF-FIN

MOVE NUM-EMPLEADO TO REP-NUM-EMPLEADO

MOVE AA-ALTA TO REP-AA-ALTA

MOVE MM-ALTA TO REP-MM-ALTA

MOVE DD-ALTA TO REP-DD-ALTA

MOVE REPORTE-LAYOUT TO REGISTRO-REPORTE

WRITE REGISTRO-REPORTE

READ PERSONAL INTO PERSONAL-LAYOUT AT END SET END-OF-FIN TO TRUE

END-PERFORM.

CLOSE PERSONAL REPORTE.

GOBACK.



... Ejemplo del programa COBOL desde el PSEUDOCÓDIGO como solución.

SOLUCIÓN 2 DE CÓMO ENFOCAR LA PROCEDURE DIVISION

```
PROCEDURE DIVISION
    OPEN INPUT PERSONAL.
    OPEN OUTPUT REPORTE.
    READ PERSONAL INTO PERSONAL-LAYOUT    AT END SET END-OF-FIN TO TRUE
    PERFORM UNTIL END-OF-FIN
        PERFORM      ARMO-REPORTE
        PERFORM      LEO-PERSONAL
    END-PERFORM
    CLOSE PERSONAL REPORTE
    GOBACK.

ARMO-REPORTE
    MOVE  NUM-EMPLEADO      TO REP-NUM-EMPLEADO
    MOVE  AA-ALTA           TO REP-AA-ALTA
    MOVE  MM-ALTA           TO REP-MM-ALTA
    MOVE  DD-ALTA           TO REP-DD-ALTA
    MOVE  REPORTE-LAYOUT TO REGISTRO-REPORTE
    WRITE REGISTRO-REPORTE

LEO-PERSONAL
    READ  PERSONAL INTO PERSONAL-LAYOUT    AT END SET END-OF-FIN TO TRUE.
```



... Formato de la sentencia COBOL

COLUMNAS

01-06

07

08-11

12-72

73-80

USO COBOL

Número de secuencia de área

Indicador de área

Área A

Área B

Designación del programa

El indicador de área, si lleva un '*', indica que la línea es un comentario.

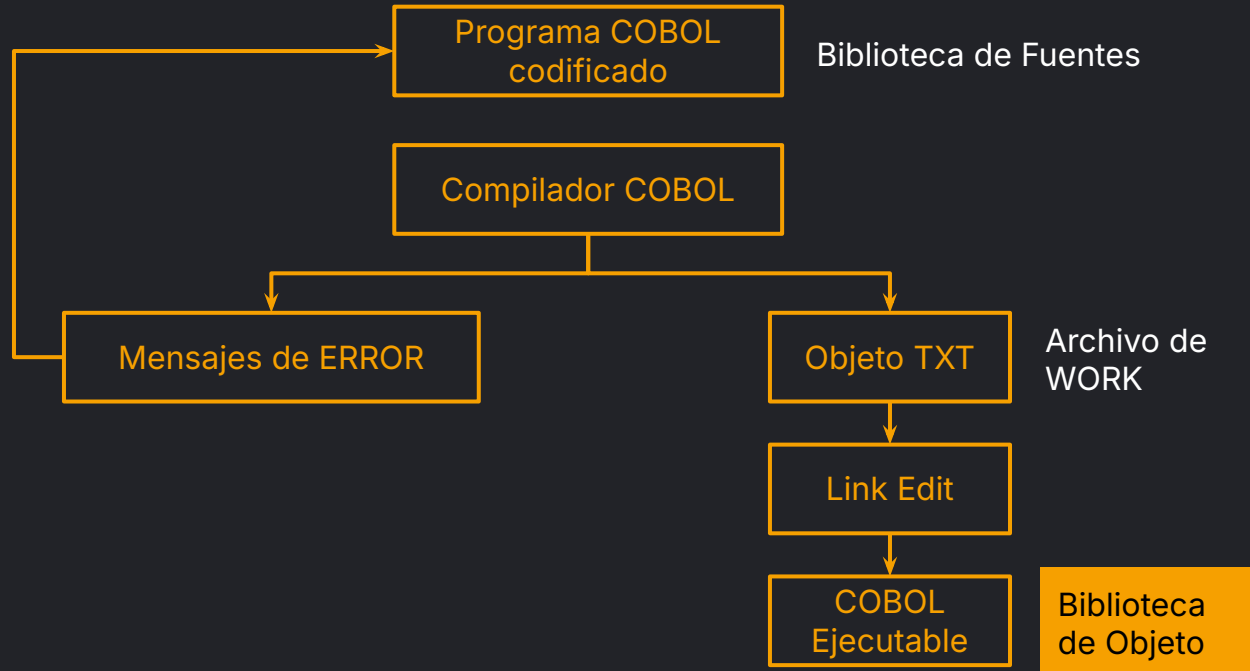
En el Área A, se describen

- Las DIVISION / SECTION / Nombres de párrafos / Nivel de indicador (FS o SD) / Nivel de número (01 o 77)

En el Área B, describen

- Entradas, sentencias, cláusulas y continuación de línea.

... Ahora con un programa COBOL codificado, cómo lo puedo ejecutar





... Descripción general

Un programa fuente COBOL es un conjunto de sentencias COBOL sintácticamente correctas.

Las sentencias, entradas, párrafos y secciones de un programa COBOL son agrupadas en cuatro divisiones que forman la estructura general de dicho programa.

El formato general y el orden de presentación de las entradas y sentencias que constituyen un programa COBOL es:

1. IDENTIFICATION DIVISION.
2. ENVIRONMENT DIVISION.
3. DATA DIVISION.
4. PROCEDURE DIVISION.

Estos nombres representan las cabeceras de las cuatro divisiones mencionadas



IDENTIFICATION DIVISION



... IDENTIFICATION DIVISION.

La división de IDENTIFICATION, sirve para proporcionar un nombre para identificar el programa. Opcionalmente se puede especificar información acerca del autor, fecha en que fue escrito, etc..

El formato general para la división es el siguiente.

IDENTIFICATION DIVISION.

PROGRAM-ID. Nombre del programa.

[AUTHOR. Nombre del programador.]

[INSTALLATION. Lugar de instalación.]

[DATE-WRITTEN. Fecha en que fue escrito.]

[DATE-COMPILED. Fecha en que fue compilado.]

[SECURITY. Entrada de comentarios.]

Solo las dos primeras líneas, IDENTIFICATION DIVISION y PROGRAM-ID, son obligatorias.



... **Program-id.**

Este párrafo sirve para especificar el nombre del programa. Este nombre será utilizado por el compilador o durante la ejecución para indicar en qué programa ocurrió algún error.

El nombre del programa debe ajustarse a las reglas de formación de un identificador COBOL, vistas en el capítulo anterior.

El resto de los párrafos son opcionales, pueden aparecer en cualquier orden, y su utilización queda perfectamente definida en el formato descrito.

Si un comentario ocupa más de una línea, no está permitido utilizar el guión como carácter de continuación, sin embargo puede ser continuado en las líneas siguientes, comenzando en el margen B.




ESTRUCTURA DE UN PROGRAMA COBOL

IDENTIFICATION DIVISION.

PROGRAM-ID. PROGCONS.

ENVIRONMENT DIVISION.

.



ENVIRONMENT DIVISION



... ENVIRONMENT DIVISION.

La división de ENTORNO permite definir el tipo de ordenador para el que fue escrito el programa, así como los dispositivos periféricos necesarios para soportar los ficheros utilizados en el programa. El formato general de la sección es el siguiente:

ENVIRONMENT DIVISION.

[CONFIGURATION SECTION.

SOURCE-COMPUTER. Modelo de ordenador.

OBJECT-COMPUTER. Modelo de ordenador.

SPECIAL-NAMES.

Nombres especiales.]

[INPUT-OUTPUT SECTION.

FILE-CONTROL.

Control de ficheros.

[I-O CONTROL.

Control de entrada / Salida.]]



... CONFIGURATION SECTION

Esta sección es opcional. Su utilidad es documentar el programa COBOL, indicando el modelo de ordenador utilizado y asociar los nombres mnemónicos especificados por el programador a nombres utilizados por el compilador. A continuación se describen los párrafos que la componen.

- **Párrafo SOURCE:** Sirve para especificar el modelo de ordenador que va a compilar el programa. Su formato es el siguiente:
- **SOURCE COMPUTER.** Modelo de ordenador.
[WITH DEBUGGING MODE.]

La opción WITH DEBUGGING MODE permite introducir en el fichero fuente COBOL, código que nos servirá de ayuda para depurar el programa. Párrafo OBJECT: Sirve para especificar el modelo de ordenador que va a ejecutar el programa.



... CONFIGURATION SECTION

Párrafo SPECIAL-NAMES:

El formato de este párrafo es el siguiente:

SPECIAL-NAMES

[Nombre de función IS nombre mnemónico]

[ALPHABET]

[DECIMAL-POINT IS COMMA]

[CURRENCY SIGN IS literal]

Nombre de función es un nombre reconocido por el compilador, por ejemplo PRINTER y nombre mnemotécnico es un nombre creado por el programador.



... CONFIGURATION SECTION

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SPECIAL-NAMES.

DECIMAL-POINT IS COMMA

CURRENCY SIGN IS '\$US' WITH PICTURE SYMBOL '\$'.

DATA DIVISION.

WORKING-STORAGE SECTION.

*

01 WS-SALDO-AUX	PIC	9.999.999,99	VALUE ZEROS.
-----------------	-----	--------------	--------------

01 WS-SALDO-AUX2	PIC	\$\$.\$9,99	VALUE ZEROS
------------------	-----	-------------	-------------



... SECCIÓN DE ENTRADA Y SALIDA

Esta sección es opcional. El nombre de la misma, INPUT-OUTPUT SECTION, deberá figurar cuando se especifique alguno de sus párrafos.

Su Formato es el siguiente:

- INPUT-OUTPUT SECTION.
- FILE-CONTROL.
- { entrada de control de los ficheros } ...



... ENVIRONMENT DIVISION

INPUT-OUTPUT SECTION.

- Se definen con la sentencia SELECT que se escribe en margen B.
- Definición de archivos secuenciales de entrada y salida.

SELECT PERSONAL ASSIGN TO DDPERS.

FILE NAME en FILE SECTION
programa.

DDNAME en confección en JCL de ejecución



... ENVIRONMENT DIVISION

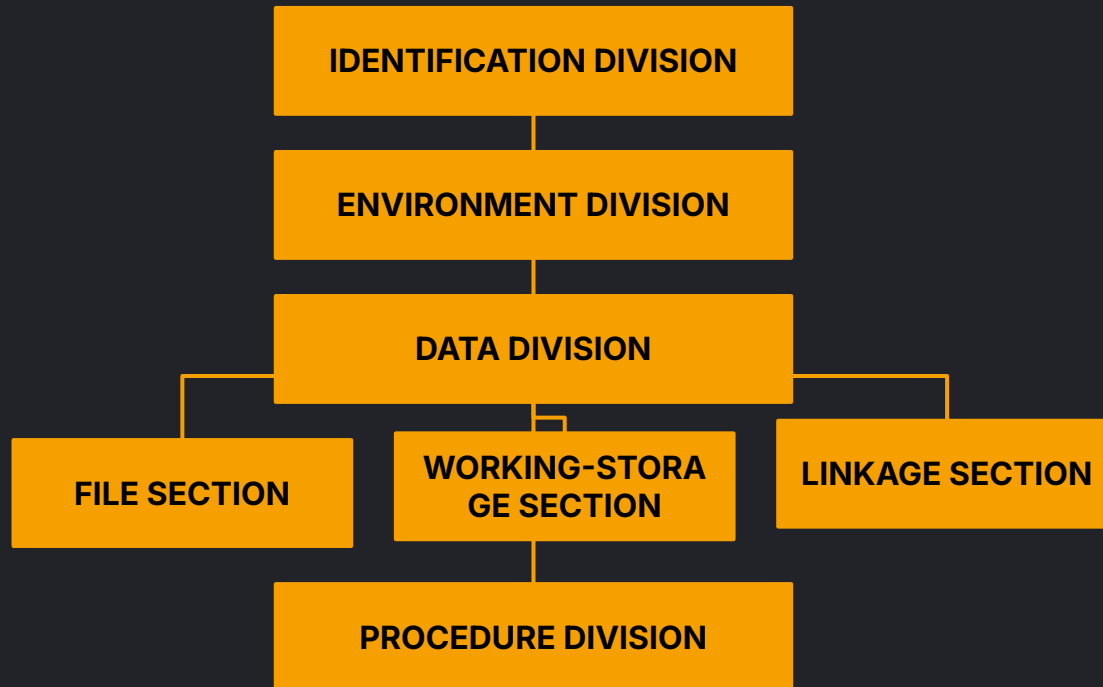
INPUT-OUTPUT SECTION.

- Definición de archivos secuenciales.
SELECT PERSONAL ASSIGN TO DDPERSO
FILE STATUS IS WS-PER-CODE.



DATA DIVISION

... Data Division





... DATA DIVISION

FILE SECTION

- Define las estructuras de archivos. Las áreas de datos en la FILE SECTION no están disponibles hasta que los archivos no son abiertos en PROCEDURE DIVISION. Esta SECTION solo COBOL BATCH.
 - FD (File Definition) descripción de entradas
 - Descripción del registro

WORKING-STORAGE SECTION

- Entrada descripción de registros
- Entrada descripción de ítems.

LINKAGE SECTION

- Esta SECTION es opcional, para describir registros e ítems que intervienen en la comunicación entre programas.



... DATA DIVISION

FILE SECTION

- Ejemplo de FD para archivo secuencial, con longitud variable
- DATA DIVISION.
- FILE SECTION.
- FD PERSONAL
- DATA RECORD IS REGISTRO-PERSONAL
- RECORDING MODE IS V.
- 01 REGISTRO-PERSONAL PIC X(4092).
- La longitud del registro lleva 4 BYTES para la longitud del registro.
- La lectura debe realizarse con READ PERSONAL INTO WS-PERSONAL (WS-PERSONAL definido en WS).

**¡Muchas
gracias!**

