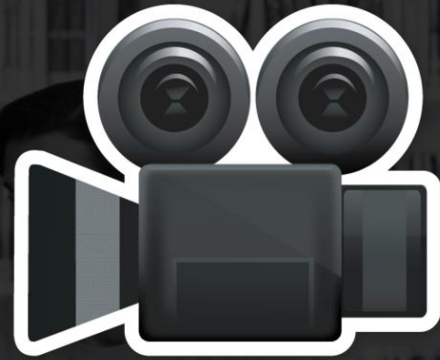


Programador Cobol Mainframe

Introducción a la programación Cobol - Parte 2

**Recuerda
poner a grabar
la clase.**



Confección de un programa Cobol



Data Division Copy



... Copy

Un programa COBOL puede y es aconsejable, re-utilizar el código programado, recurriendo a la inclusión de estos por medio de COPY

- Declaración de archivos (Select)
- Definición de archivos y sus registros (FD)
- Variables propias de cada archivo (Ubicación, File Status)
- Definición de variables comunes para todos los programas Working-Storage
- Definición de Tablas de Bases de Datos
- Rutinas comunes (Cálculos, Mensajes, Errores, LOG, etc)
- Esqueletos pre-definidos con pseudo-texto



... Formato de un REPLACE

COPYS

- Se proveen los copys que usan los programas
- Archivos y Tablas de la Base de Datos
- Ejemplos de Procedimientos comunes a todos los programas



COPY DE DATA DIVISION - EJEMPLO

Vamos a crear el COPY FILEA, como miembro FILEA en SYS1.COPYLIB, que contendrá la siguiente estructura:

```
01    FILEA
      02    NOMBRE-EMPL          PIC X(40).
      02    DOMICILIO            PIC X(40).
      02    CIUDAD               PIC X(20).
      02    COD-POSTAL           PIC X(4).
```

..... Fuente del programa

WORKING-STORAGE SECTION.

COPY FILEA.

..... Posterior a la compilación

WORKING-STORAGE SECTION.

```
01    FILEA
      02    NOMBRE-EMPL          PIC X(40).
      02    DOMICILIO            PIC X(40).
      02    CIUDAD               PIC X(20).
      02    COD-POSTAL           PIC X(4).
```



COPY DE DATA DIVISION - EJEMPLO COPY REPLACING

Sirve para realizar cambios en el texto del COPY previamente generado en COPYLIB, por necesidad del programador.

Ejemplo de cómo es reemplazado en el programa el contenido de un COPY
COPY PAYLIB REPLACING

FLDA BY PAY-RECORD
FLDB BY HRLY-RATE
FLDC BY HRS-WORKD.

TEXTO DEL COPY EN LA BIBLIOTECA

01 FLDA.
02 FLDB PIC 999V99.
02 FLDC PIC 999V99

TEXTO DEL COPY EN EL PROGRAMA

01 PAY-RECORD.
02 HRLY-RATE PIC 999V99.
02 HRS-WORKD PIC 999V99.



COPY DE DATA DIVISION - EJEMPLO COPY REPLACING

Con Pseudo texto, ejemplo. Los delimitadores son (==)

COPY PAYLIB REPLACING

```
==:PFFX:==          BY          ==PAY==.
```

TEXTO DEL COPY EN LA BIBLIOTECA

```
01  :PFFX:.
    02  :PFFX:-RATE    PIC 999V99
    02  :PFFX:-WORD    PIC 999V99
```

TEXTO DEL COPY EN EL PROGRAMA

```
01  PAY.
    02  PAY-RATE    PIC 999V99.
    02  PAY-WORD    PIC 999V99.
```



DATA DIVISION

Definición Tablas

... MANIPULACIÓN DE TABLAS



Una tabla es un conjunto de elementos del mismo tipo, que comparten en común un nombre común pero que son distinguibles por la posición que ocupa cada uno de ellos en la tabla.

Cada elemento de la tabla puede contener un dato numérico o una cadena de caracteres o una combinación de ambos. La representación de una tabla se hace mediante variables suscritas o de subíndices y puede tener una o varias dimensiones.

Descripción de una tabla.

La descripción de una tabla se hace en la sección FILE y en la sección WORKING-STORAGE utilizando la cláusula OCCURS.



... Cláusula OCCURS.

Declara una estructura de datos denominada tabla. Su formato es el siguiente:

1. No puede especificarse en una descripción de nivel 01, 77, 88 o 66. Entero indica el número de veces que se repite un campo con la misma descripción.
2. Se definen solamente en los niveles 02 a 49 ambos inclusive



... Sentencia SET.

Permite transferir el valor de un literal, nombre de datos o de un índice a uno o más identificadores.

Formato 1:

Cuando se ejecuta esta sentencia el valor del operando que sigue a TO es transferido a los operandos que siguen a SET.

SET I J TO 1.

Formato 2:

Este formato permite ir incrementando (UP BY) o decrementando (DOWN BY) el índice o índices en el valor especificado por nombre dato o entero.

SET I J UP BY 1.



OCCURS:

- Esta cláusula es para definir tablas de 'n' dimensiones, y pueden ser referenciadas por índices o suscriptores
- Esta cláusula no se puede usar en los niveles 01/66/77/88

OCCURS entero TIMES ASCENDING KEY IS data-name-2 INDEXED BY
index-name-2
DESCENDING

Opcional

OCCURS entero-1 TO entero-2 TIMES DEPENDING ON data-name-1
01 TABLA-MESES.

02 MES OCCURS 12 TIMES PIC X(10) INDEXED X.

01 TABLA-ARBOL.

02 ELEM-1 OCCURS nn TIMES.

03 ELEM-2 OCCURS nn TIMES.

04 ELEM-3 PIC X(8) OCCURS nn TIMES.

MOVE ELEM-3 (1,2,1) TO



OCCURS- EJEMPLO CON SUSCRIPTORES

01 TABLA-ELEM PIC X(3) OCCURS 31 TIMES.

01 FERIADO PIC XX.

 88 SI-FER VALUE 'SI'

 88 NO-FER VALUE 'NO'

77 CANTIDAD PIC 9(3) BINARY VALUE 1.

.....

 SET NO-FER TO TRUE.

 PERFORM UNTIL CANTIDAD GREATER THAN 31

 IF TABLA-ELEM (CANTIDAD) = 'FER'

 SET SI-FER TO TRUE

 ADD 31 TO CANTIDAD

 ELSE

 ADD 1 TO CANTIDAD

 END-IF

 END-PERFORM.



... COPY DE PROCEDURE DIVISION

Este tipo de COPY se invoca en la PROCEDURE DIVISION y contiene un conjunto de sentencias, de uso común para los programas, por ejemplo tratamiento de errores.

COPY text-name OF nombre-biblioteca SUPRESS REPLACING operando-1 BY operando-2

literal-1 IN literal-2

Opcional

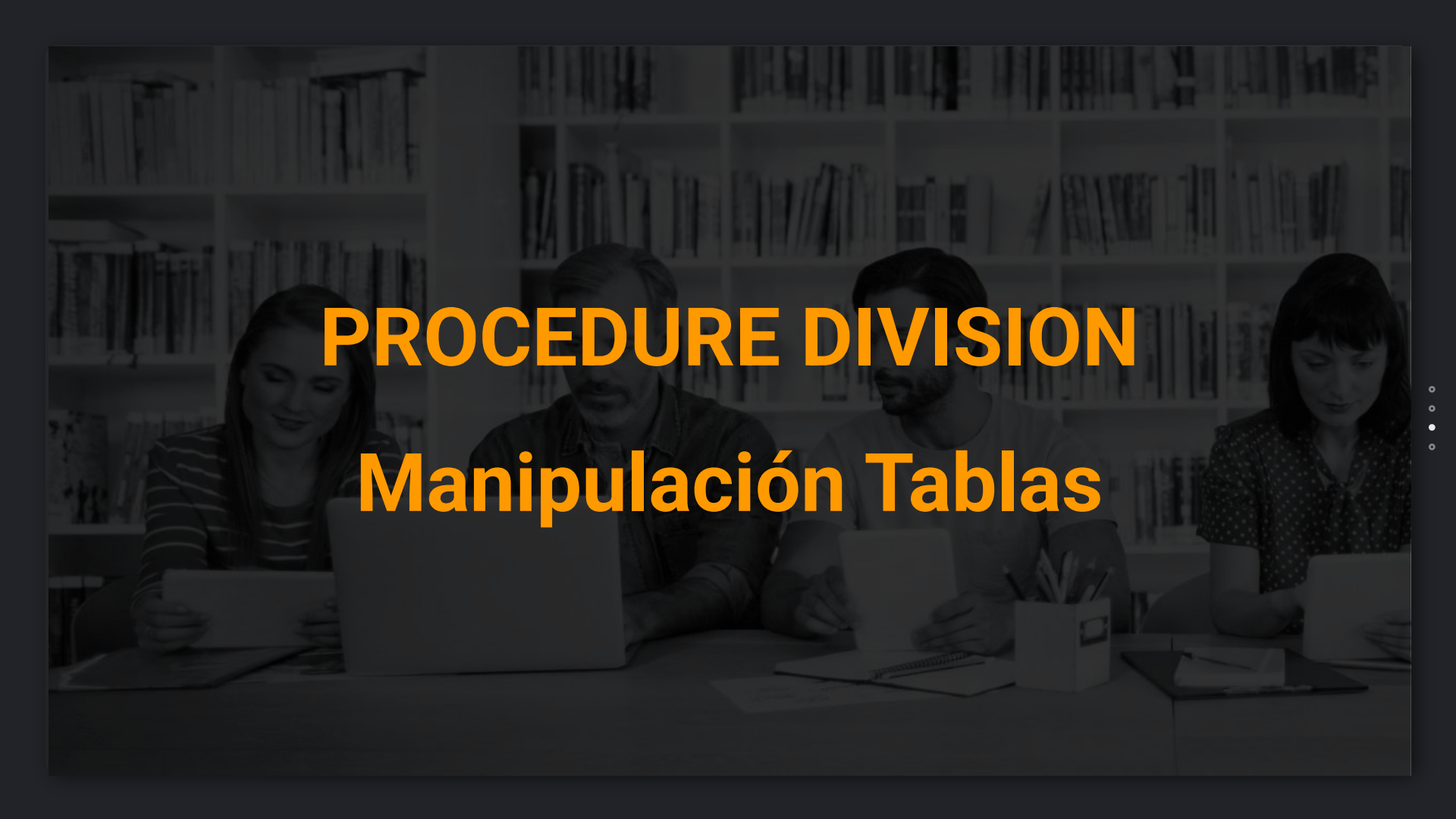


COPY DE PROCEDURE DIVISION EJEMPLO

```
Invocando rutina error  
PROCEDURE DIVISION.  
    READ FILEA  
    IF RETORNO IS NOT EQUAL 'OK'  
        PERFORM INVOCO-ERROR  
    ELSE .....  
INVOCO-ERROR  
    COPY RUT-ERROR
```

TEXTO **DEL** **COPY** **EN** **LA** **BIBLIOTECA**

```
MOVE RETORNO TO COD-ERROR.  
CALL 'RUTERROR' USING COD-ERROR.
```



PROCEDURE DIVISION

Manipulación Tablas

... Manipulación de tablas



Cuando se especifica nombre dato-2 debe ser descrito por medio de la cláusula `USAGE IS INDEX` o como un campo elemental entero. El valor de este campo evoluciona paralelamente al índice de la tabla y se emplea cuando:

- A) Aparte de realizar la búsqueda de un determinado elemento de la tabla se desea conocer su posición.
- B) Se desea incrementar simultáneamente el índice de otra tabla diferente a la tabla en la que se está realizando la búsqueda. Las condiciones de las cláusulas `WHEN` se evalúan en el orden en que están descritas.



PROCEDURE DIVISION


Entry Point /

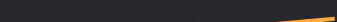
Terminación





... Formato

- Es el punto de entrada de ejecución de un programa (ENTRY POINT), con el nombre definido en PROGRAM-ID

PROCEDURE DIVISION.  Obligatorio

LINKAGE SECTION.  Datos de Invocación por programa llamador

01 PARAM.  Datos de PARM de EXEC en JCL

02 LONG-PARAM PIC XX. 

02 DATO-1 PIC X(8).

02 DATO-2 PIC X(6).

PROCEDURE DIVISION USING PARAM. Código de retorno al programa llamador debe

PROCEDURE DIVISION USING DATO-RUTINA RETURNING CODE-RUTINA. Estar definida en WS



La **DIVISIÓN DE PROCEDIMIENTOS** indica la secuencia de operaciones necesarias, sentencias COBOL, para resolver un problema dado. El formato general para esta división es el siguiente:

```
{Nombre de Sección SECTION [numero-segmento].  
[Nombre de párrafo. [Sentencia] ...]...}...
```

La cláusula USING se especifica solamente en programas que luego son llamados (CALL) o enlazados (CHAIN).

Puede ser dividida en secciones y estas a su vez en párrafos. Cada párrafo generalmente agrupa varias sentencias. La división en párrafos es frecuente.

Opcionalmente la PROCEDURE DIVISION puede contener secciones DECLARATIVAS. En caso de utilizarlas, estas deben ser las primeras en definirse.



... TERMINACIÓN DE UN PROGRAMA

- EXIT PROGRAM
 - Especifica el fin de una rutina y retorna el control al programa llamador
- STOP RUN
 - Al fin de ejecución retorna el control al sistema operativo
- GOBACK
 - Cumple las dos funciones la de EXIT PROGRAM, si es una rutina y la STOP RUN si es un programa MAIN
 - EL GOBACK es el recomendado.



TERMINACIÓN DE SENTENCIAS

END-ADD	END-INVOKE	END-SEARCH	END-CALL	END-MULTIPLY
END-START	END-COMPUTE	END-PERFORM	END-STRING	END-DELETE
END-READ	END-SUBSTRING	END-DIVIDE	END-RETURN	END-UNSTRING
END-EVALUATE	END-REWRITE	END-WRITE	END-IF	



¿Consultas?





**¡Muchas
gracias!**



Codeki™