

## MÓDULO 6 | ARCHIVOS - ALGORITMOS DE PROCESAMIENTO

### 1. CORTE DE CONTROL

#### EL CORTE DE CONTROL

es una técnica de procesamiento de datos ordenados por diversos criterios, que permite agruparlos para obtener subtotales.

El corte de control es un proceso en el cual los registros del archivo se encuentran ordenados por el valor de uno o más campos, denominados campos **clave** o **llave** (1), el ordenamiento puede ser ascendente –lo más común– o descendente.

Este tipo de proceso generalmente es utilizado para realizar informes o reportes, en el cual se deba emitir de cada grupo –formado por el mismo valor del campo clave– totales, promedios, máximos o mínimos, etc.

En líneas generales este proceso posee una estructura que es bastante característica y que pasaremos a detallar a continuación.

Cada corte de control establece un **nivel**, así si existen **n** cortes de control, habrá **n** niveles (2) (3), y cada uno de estos **n** cortes o niveles están contenidos dentro de un ciclo indefinido, a estos ciclos se le suma uno más, y representa el fin del proceso (4), siendo este ciclo el de mayor nivel y el más externo (2), luego en forma anidada se van desarrollando los ciclos internos (3) en un orden que va de mayor nivel de corte hasta llegar al menor nivel de corte.

El ciclo de mayor nivel establece el fin del proceso, siendo su condición, el fin del archivo que se está procesando, por medio de una variable de tipo boolean. (2)

Las condiciones de los ciclos internos van **arrastrando** las condiciones de los ciclos anteriores o más externos a la que se le suma la propia condición del ciclo, es decir, la que producirá el corte de control de ese nivel, por lo tanto, cuanto más anidado sea el ciclo, más condiciones contendrá.

El ciclo externo contiene una condición, y el ciclo más interno posee  $n + 1$  condiciones. Si un proceso requiere **n** cortes, entonces la cantidad de ciclos será de  **$n + 1$** . Además **cada corte de control obliga a que los datos se encuentren ordenados por el valor del campo clave comenzando desde el de mayor nivel hasta el de menor nivel**, por ejemplo, si se requieren obtener totales de las horas trabajadas por diferentes empleados, los datos deberían estar organizados por el código de empleado; un segundo ejemplo, si los empleados realizan sus actuaciones en una filial y el proceso requiere informar además totales por filial, en este caso los datos deben encontrarse ordenados primero por código de filial y luego por código de empleado, debido a que es de esperar que primero cambien los empleados y luego cambien las filiales.

**El corte de control requiere de una lectura anticipada** (1). Debido a que las próximas lecturas se realizan en otro punto del programa (5), la lectura especial va a ser un módulo con tres parámetros, a saber: el archivo, el registro a devolver y el estado de la operación de lectura, de tipo boolean, si es falso indicará que no fue fin de archivo y se leyó un registro, caso contrario, indicará que es fin de archivo (4). Las próximas lecturas del archivo se realizarán en el ciclo más interno y como última acción. (5)

A continuación presentamos el diagrama de flujo de lo expuesto hasta aquí. Note las referencias numéricas en color rojo en el DFD.

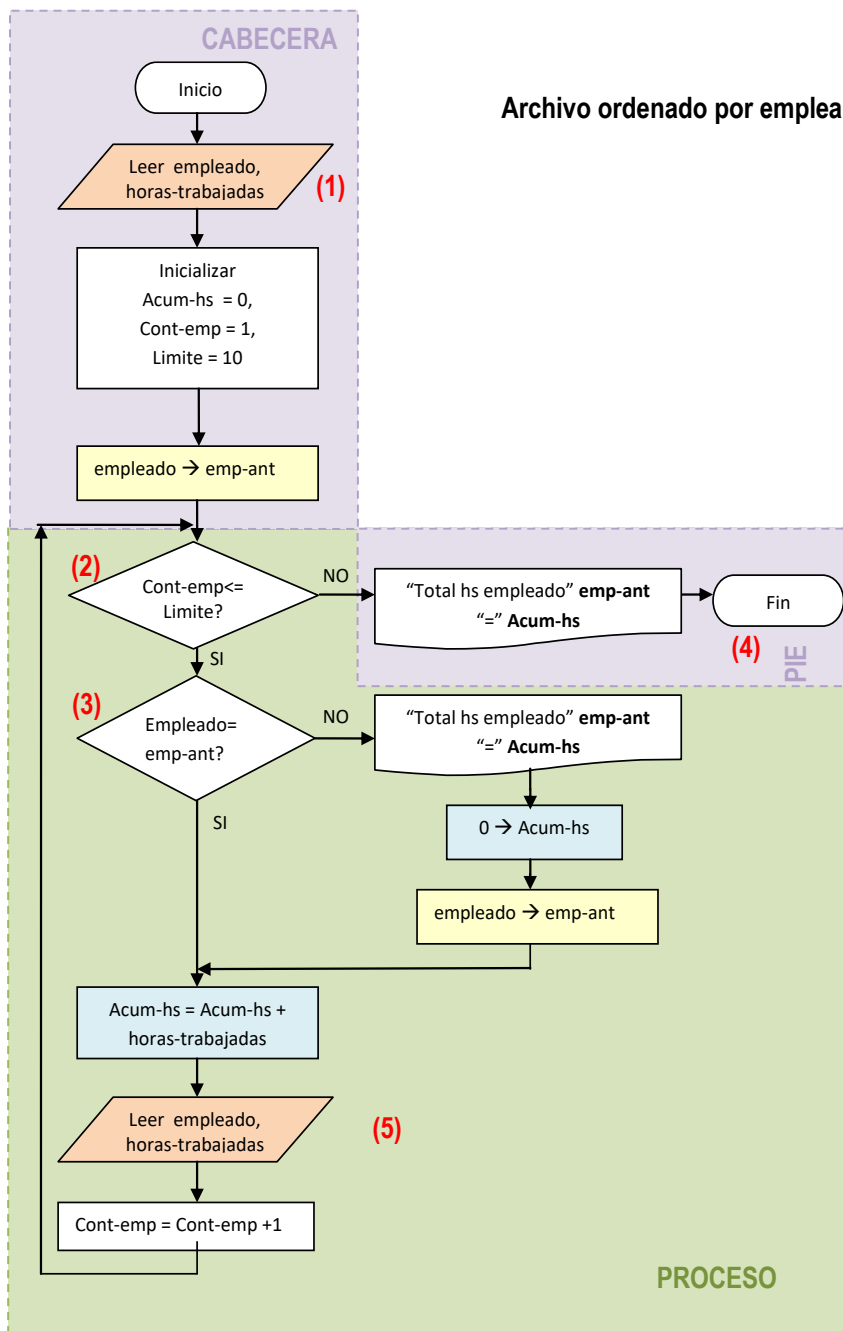
### Ejemplo:

“En una empresa que tiene 10 empleados, se desea saber cuántas horas trabajó cada empleado la última semana”

**Entrada:** empleado, horas-trabajadas

**Salida:** Acum-hs =Total por empleado

**Variables:** emp-ant =empleado anterior; Cont-emp=contador empleados



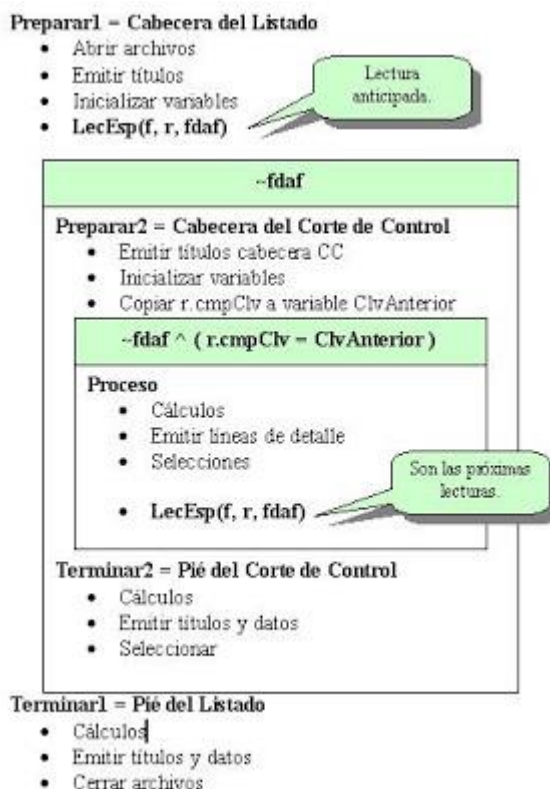
Archivo ordenado por empleado

Archivo: Empleados			
Cant. de empleados: 10			
Cant. de registros: 20			
#	Empleado	Horas trabajadas	...
1	Alvarez	5	
2	Alvarez	8	
3	Durand	4	
4	Ferreyra	7	
5	Ferreyra	7	
6	Kara	4	
7	Kara	6	
8	Morel	5	
9	Moretti	8	
10	Moretti	8	
11	Moretti	8	
12	Ostertag	6	
13	Ostertag	6	
14	Ostertag	8	
15	Pani	5	
16	Salvat	2	
17	Salvat	7	
18	Salvat	5	
19	Zavaleta	3	
20	Zavaleta	3	

Antes de ingresar a un ciclo denominamos a esa región del algoritmo **cabecera**. Al salir de un ciclo, denominamos a esa región del algoritmo **pie**. Dentro de un ciclo, denominamos a esa región del algoritmo **proceso**.

- En la **cabecera** generalmente realizamos las siguientes acciones: **Inicializar, Emitir títulos y datos.**
- En el **pie** generalmente realizamos las siguientes acciones: **Cálculos, Emitir totales, promedios, máximos o mínimos, tomar alguna decisión.**
- En el **proceso** generalmente realizamos las siguientes acciones: **Cálculos, Emitir líneas de detalle, tomar alguna decisión.**

A continuación se presenta un modelo de algoritmo de Corte de Control en un formato general.



#### IMPORTANTE:

Es necesario aplicar este módulo cuando un proceso requiera una lectura anticipada. Este método funciona siempre, es decir, aún para aquellos procesos en que no requiera lectura anticipada, pero en este último caso debemos ubicar el módulo en los dos puntos dentro del algoritmo como fuera indicado anteriormente.

## 2. CORTE DE CONTROL SIMPLE

El algoritmo de **Corte de Control Simple** es aquel que tiene **un solo corte**.

Para la presentación del algoritmo es más sencillo hacerlo mediante un ejemplo:

*“Se ingresan los datos de alumnos: su código y sus notas. Se desea conocer de cada uno el promedio de las notas ingresadas. Se desconoce el número de notas por cada alumno. Se termina la entrada de los datos cuando el número de alumno ingresado es el ‘0’”*

Aclaración: Esta última oración que menciona que se desconoce la cantidad de notas por alumno indica que debe utilizarse la técnica de Corte de Control; ya que si por ejemplo se dijera que “se ingresar 5 notas por alumno”, el ejercicio puede realizarse fácilmente con un ciclo “for” o por contador.

#### Lote de Prueba:

Alumno	Nota
120	4
120	7
120	7
130	6
139	4
139	8

Como fue indicado anteriormente, los datos se encuentran ordenados según la jerarquía correspondiente, en este caso los alumnos.

El algoritmo de Corte de Control no es más que un simple “**método**” de ciclos, que en este caso se colocan de la siguiente manera:

#### **Pseudocódigo**

Proceso Corte\_Control

Definir nota Como Entero

Definir aux Como Entero

Definir alumno Como Entero

Definir suma Como Entero

Definir cant Como Entero

Leer alumno

aux = alumno

cant = 0

Mientras alumno = 0 Hacer

    Mientras aux = alumno Hacer

        Leer nota

        suma = suma + nota

        cant = cant + 1

        Leer alumno

    Fin Mientras

    Escribir suma/cant

    cant = 0

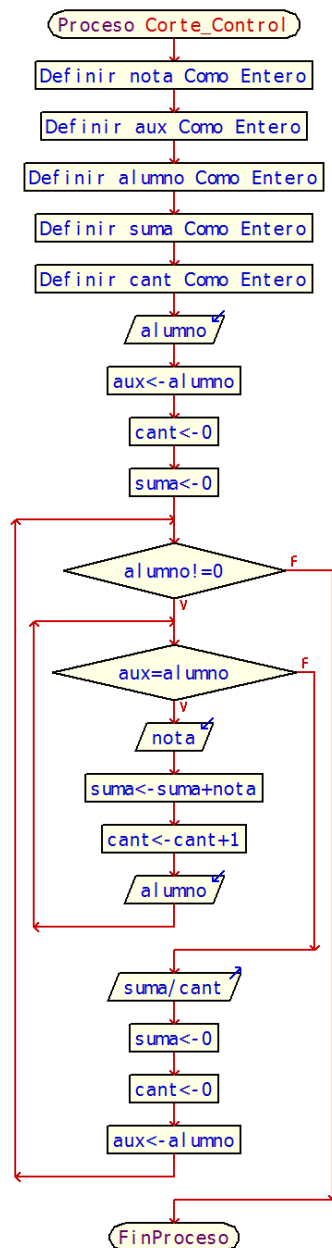
        suma = 0

    aux = alumno

Fin Mientras

FinProceso

**Diagrama de Flujo** (en base al pseudocódigo anterior)



### 3. CORTE DE CONTROL MULTIPLE

El algoritmo de Corte de Control Múltiple es aquel que tiene **n+1 cortes**.

Haremos la presentación del algoritmo mediante un ejemplo:

Una empresa de ventas tiene la información de sus vendedores clasificada por sucursal y dentro de cada sucursal, por vendedor. Cada registro de un vendedor posee los siguientes datos: código de la sucursal, código del vendedor, número de la factura y valor de la venta.

Hacer un algoritmo que muestre la relación de ventas hechas por cada vendedor, dando subtotales por vendedor, por sucursal y el gran total de ventas de la empresa.

## Análisis

### Datos de entrada

- Código de la sucursal.
- Código del vendedor.
- Número de la factura.
- Valor de la venta.

### Datos de salida

- Código de la sucursal.
- Código del vendedor.
- Número de la factura.
- Valor de la venta.
- Total de ventas por cada vendedor.
- Total de ventas por sucursal.
- Total de ventas de la empresa.

### Proceso

Como no se conoce la cantidad de sucursales que tiene la empresa es necesario implementar un esquema cualitativo. Para el control del ciclo de entrada de datos se escoge el campo código de la sucursal y, el ciclo se repetirá hasta que se encuentre un valor de código de sucursal igual a cero.

Para dar totales por vendedor es necesario hacer un rompimiento de control de ejecución cada que, dentro de una sucursal, se cambie de un vendedor a otro. Para ello se escoge una variable auxiliar que almacene el código del vendedor anterior y se actualiza cuando entre información de un vendedor distinto. Como hay que dar totales por sucursal hay que hacer otro rompimiento de control cuando se cambie de una sucursal a otra. Esta clase de problema se denomina rompimiento de control de segundo nivel; pueden existir rompimientos de más niveles. La salida de información estará de acuerdo con el siguiente formato:

VENTAS DEL MES SUCURSAL XX		
VENDEDOR	NRO. FACTURA	VALOR VENTA
XXX	XXX	XXX.xx
XXX	XXX	XXX.xx
TOTAL VENDEDOR	XXX	XXX XXX.xx
TOTAL VENDEDOR	XXX	XXX XXX.xx
TOTAL SUCURSAL	XXX	
TOTAL EMPRESA	XXX	

### Definición de variables

COSUC:	Código de la sucursal
COSUCA:	Código de la sucursal anterior
COVEN:	Código del vendedor
COVENA:	Código del vendedor anterior
NF:	Número de la factura
VV:	Valor de la venta
TOMEN:	Total de ventas de un vendedor
TOTSUC:	Total de ventas de una sucursal
TOTE:	Total de ventas de la empresa

### Algoritmo

```

INICIO
    TOTE = 0
    LEA: COSUC, COVEN, NF, VV
    MIENTRAS COSUC <> 0 HAGA
        COSUCA = COSUC
        TOTSUC = 0
        ESCRIBA: "VENTAS DEL MES SUCURSAL:", COSUC
        ESCRIBA: "VENDEDOR    NRO. FACTURA    VALOR VENTA"
        MIENTRAS (COSUC = COSUCA)
    ^
    (COSUC <> 0) HAGA
        COVENA = COVEN
        TOTVEN = 0
        MIENTRAS (COVEN = COVENA)
    ^
    (COSUC <> 0)
    ^
    (COSUC = COSUCA)
        ESCRIBA: COVEN, NF, VV
        TOTVEN = TOTVEN + VV
        LEA: COSUC
        SI COSUC <> 0 ENTONCES
            LEA: COVEN, NF, VV
        FIN_SI
    FIN_MIENTRAS
    ESCRIBA: "TOTAL VENDEDOR:$", TOTVEN
    TOTSUC = TOTSUC + TOTVEN
    FIN_MIENTRAS
    TOTE = TOTE + TOTSUC
    ESCRIBA: "TOTAL. SUCURSAL:$", TOTSUC
    FIN_MIENTRAS
    ESCRIBA: "TOTAL EMPRESA:$", TOTE
FIN_INICIO

```

### Prueba de escritorio

0			
02	17	19	300000.0
02	16	15	395.80
02	16	11	800.00
01	92	31	489.50
01	92	30	3000.00
01	89	25	586.00
01	89	23	2243

## Salida

Ventas del mes sucursal 01		
Vendedor	Nro. factura	Valor venta
89	23	2243.50
89	25	586.00
Total vendedor		2829.50
92	30	3000.00
92	31	489.50
Total vendedor		3489.50
Total sucursal		6319.00
Ventas del mes sucursal 02		
Vendedor	Nro. factura	Valor venta
16	11	800.00
16	15	395.80
Total vendedor		1195.80
17	19	300000.00
Total vendedor		300000.00
Total sucursal		301195.80
Total empresa		307514.80

## Aspectos a tener en cuenta:

- El esquema cuantitativo se implementa a través de un contador, la expresión lógica que gobierna el ciclo se establece de acuerdo al valor inicial que se le asigne a éste.
- Los contadores y acumuladores deben partir de un valor inicial.
- Una vez que se entra a un ciclo, éste debe ejecutar todo el conjunto de instrucciones que lo componen.
- En la implementación de cualquiera de los dos esquemas, las variables que están involucradas en la expresión lógica deben tener almacenado un valor.
- Cuando existen cálculos totales, algunos se hacen dentro del ciclo y otros pueden hacerse fuera del ciclo.
- El uso de una bandera exige que ésta tenga asignado un valor en el momento de evaluarse.
- Si un ciclo ya cumplió su misión, para que este termine antes de que la condición en forma normal sea falsa, es necesario alterar la expresión lógica