



Corte de Control

Un problema se identifica como de corte de control cuando se establece una jerarquía sobre los datos a procesar.

Esta jerarquía, con respecto a uno o más campos, requiere, necesariamente, que los datos se presenten agrupados con respecto a dicho/s campo/s y que, además, se repita/n en posiciones consecutivas.

Por ejemplo: si una empresa cuenta con un archivo de ventas con datos de la Provincia, la Localidad, el nombre del Cliente y el monto de cada venta, se puede construir un listado con tres niveles donde el primer nivel de la jerarquía sea la Provincia, luego la Localidad y por último el Cliente, siendo posible mostrar totales por cada uno de estos niveles.

Para ello, es necesario que los datos se ingresen agrupados por Provincia, dentro de una misma Provincia por Localidad y dentro de una misma Localidad por Cliente. Además, se producen repeticiones consecutivas de estos tres campos ya que por Provincia, Localidad y Cliente pueden producirse varias ventas.

Este tipo de proceso permite visualizar fácilmente la composición de la jerarquía y obtener, por cada uno de los niveles de jerarquía o cortes de control, resultados como, por ejemplo, totales, promedios, porcentajes, máximos, mínimos, etc.

En general, los datos se presentan ordenados con respecto al campo o los campos que producen el o los cortes de control ya que al ordenar, los datos quedan agrupados según se requiera.

También, el corte de control es una forma de tratar los datos que tiene por objetivo no mostrar información repetida de manera innecesaria cuando se genera una salida impresa, por pantalla o a archivo.

Por ejemplo: Una empresa dispone de un archivo con los datos de sus empleados **ordenados por categoría** con la siguiente estructura:

NOMBRE	CATEGORÍA
20 caracteres	Byte

y se requiere un listado con el siguiente diseño:

LISTA DE PERSONAL	
CATEGORÍA 1	MARTINEZ PEREZ RODRÍGUEZ
CATEGORÍA 2	GIMENEZ SUAREZ

Único Nivel en la
Jerarquía

Es necesario que en el archivo contemos con la información organizada de esta forma:



Nombre	Categoría
MARTINEZ	1
PEREZ	1
RODRIGUEZ	1
JIMENEZ	2
SUAREZ	2

Es requisito **indispensable** que los datos que contiene el archivo estén agrupados por la condición de **Corte de Control** que se solicita, en este caso por categoría.

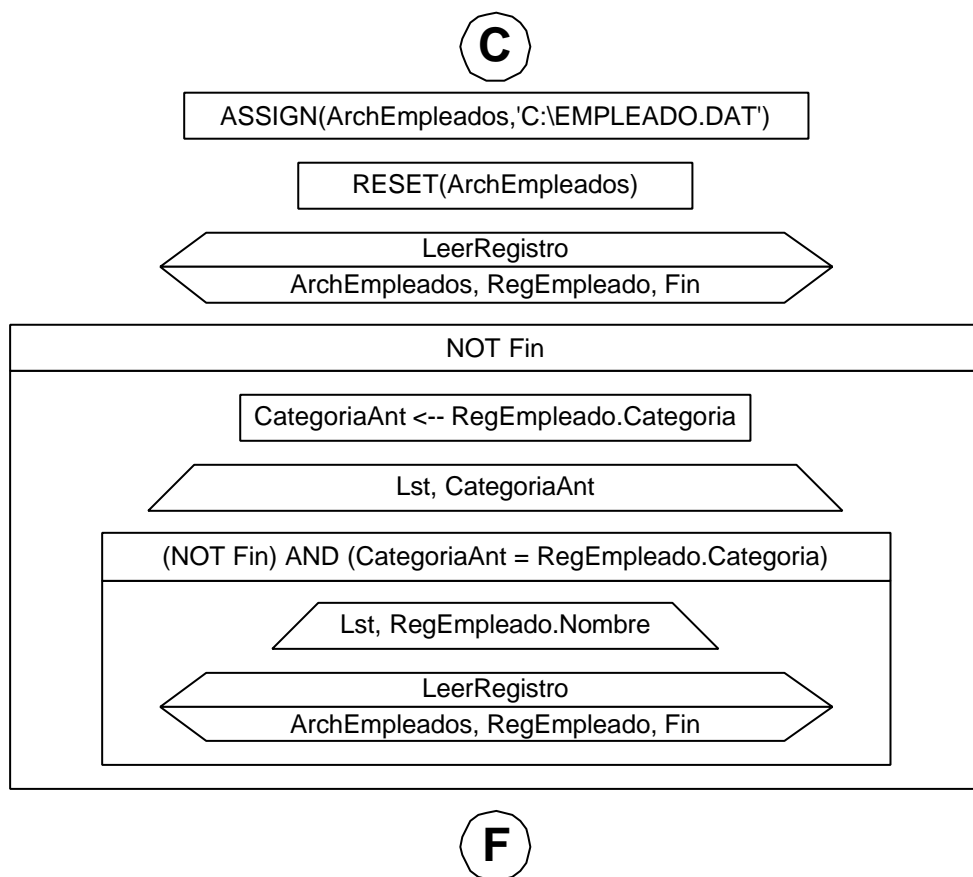
Una característica de los datos, para aplicar la técnica de corte de control, es que se repitan, en posiciones consecutivas, valores de aquellos campos sobre los que se pretende realizar el corte, en este caso la categoría.

Siguiendo con el ejemplo, dado que para una misma categoría existen muchos empleados, se necesita mantener de alguna manera el valor de la categoría del último registro leído para ir comparándolo con el valor del actual registro leído.

Dicho valor se mantiene en una variable auxiliar.

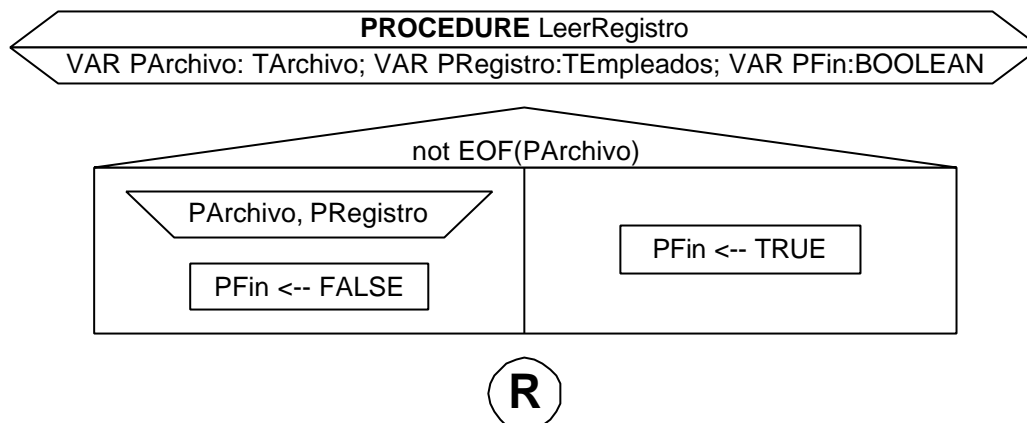
Si en la comparación el valor de la variable auxiliar y la categoría del registro actual son iguales se continúa mostrando los datos de los empleados de esa categoría y, si son diferentes, se cambia el valor de la variable que mantiene la categoría anterior y se continúa mostrando los datos de la siguiente categoría.

Algoritmo del Programa Principal





Algoritmo del módulo LeerRegistro



Se puede observar que la cantidad de estructuras repetitivas anidadas es igual a la cantidad de niveles de la jerarquía más uno (en este caso hay un nivel, por lo tanto, dos ciclos) . Un ciclo controla el fin de datos y otro ciclo procesa cada corte o nivel de agrupamiento. Otro aspecto importante es que, a medida que se anidan las repeticiones, el ciclo más interno arrastra la condición de corte del/ de los niveles anteriores más su propia condición de corte, que será la del nivel siguiente en la jerarquía.

Definición de Tipos y Procedimientos y Declaración de variables

TYPE

TNombre = **STRING**[20];

TEmpleados = **RECORD**

Nombre: TNombre;

Categoria: **BYTE**;

END;

TArchivo = **FILE OF** TEmpleados;

VAR

ArchEmpleados: TArchivo;

RegEmpleado: TEmpleados;

CategoriaAnt: Byte;

Fin : boolean;

PROCEDURE LeerRegistro (**VAR** PArchivo: TArchivo; **VAR** PRegistro: TEmpleados; **VAR** PFin: **BOOLEAN**);

BEGIN

IF NOT EOF(PArchivo) **THEN**

BEGIN

READ (PArchivo,PRegistro);

PFin := **FALSE**

END

ELSE PFin := **TRUE**

END;



Codificación en Pascal del Programa Principal

BEGIN

ASSIGN (ArchEmpleados,'C:\EMPLEADO.DAT');

RESET (ArchEmpleados);

LeerRegistro (ArchEmpleados, RegEmpleado, Fin);

WHILE NOT(Fin) DO

BEGIN

CategoriaAnt := RegEmpleado.Categoria;

WRITE (Lst, CategoriaAnt);

WHILE ((NOT(Fin)) AND (CategoriaAnt = RegEmpleado.Categoria)) DO

BEGIN

WRITE (Lst, RegEmpleado.Nombre);

LeerRegistro (ArchEmpleados, RegEmpleado, Fin)

END

END

END.