

Archivar maestros y detalle
 info q' viene de algo

Maestro: DATO de alumnos y profesor

Detalle: info q' viene en 1 to y nota
 info q' viene en una 11 + nota

1 Maestro } = criterio de ordenamiento
 no detalle } es el detalle ordenado solo empleando q' viene en el Maestro
 Un emp. de Maestro aparece solo 1 vez en detalle (max)

```
program actualizar;
const valoralto='9999';
type str4 = string[4];
prod = record
  cod: str4;
  description: string[30];
  pu: real;
  cant: integer;
end;
v_prod = record
  cod: str4;
  cv: integer; (cantidad vendida)
end;
detalle = file of v_prod;
maestro = file of prod;

var
  regm: prod;
  regd: v_prod;
  mael: maestro;
  detl: detalle;
  total: integer;
```

```
begin
  assign (mael, 'maestro');
  assign (detl, 'detalle');
  {proceso principal}
  reset (mael); reset (detl);
  while (not eof(detl)) do begin
    read (mael, regm);
    read (detl, regd);
    while (regm.cod <> regd.cod) do
      read (mael, regm);
    while (regm.cod = regd.cod) do begin
      regm.cant := regm.cant + regd.cv;
      read (detl, regd);
    end;
    seek (mael, filepos (mael)-1);
    write (mael, regm);
  end;
end.
```

```
procedure leer(var archivo: detalle; var dato: venta_prod);
begin
  if (not(eof(archivo))) then
    read (archivo, dato)
  else dato.cod := valoralto;
end;
begin
  assign(mae, 'maestro'); assign(det, 'detalle');
  reset(mae); reset(det);
  read(mae, regm);
  leer(det, regd);
  while (regd.cod <> valoralto) do begin //se procesan todas las registros del archivo detalle
    aux := regd.cod;
    total := 0;
    while (aux = regd.cod) do begin //se totaliza la cantidad vendida de productos iguales en el archivo de detalle
      total := total + regd.cant_vendida;
      leer(det, regd);
    end;
    while (regm.cod <> aux) do //se busca el producto del detalle en el maestro
      read (mae, regm);
    regm.cant := regm.cant + total; //se modifica el stock del producto con la cantidad total vendida de ese producto
    seek(mae, filepos(mae)-1); //se reubica el puntero en el maestro
    write(mae, regm); // se actualiza el maestro
  end;
  close(det);
  close(mae);
end.
```

En este se suman los vendidos del detalle y luego se actualiza de una el maestro

```
program actualizar;
const valoralto='9999';
type str4 = string[4];
prod = record
  cod: str4;
  description:
string[30];
  pu: real;
  cant: integer;
end;
v_prod = record
  cod: str4;
  cv: integer;
(cantidad vendida)
end;
detalle = file of v_prod;
maestro = file of prod;

var
  regm: prod;
  regd: v_prod;
  mael: maestro;
  detl: detalle;
  total: integer;
```

Deducción de variables

En este se va actualizando el maestro al leer cada detalle

```
procedure leer (var archivo:detalle; var dato:v_prod);
begin
  if (not eof(archivo)) then
    read (archivo,dato)
  else dato.cod := valoralto;
end;
begin
  assign (mael, 'maestro'); assign (detl, 'detalle');
  reset (mael); reset (detl);
  leer (detl,regd); //se procesan todas las registros del archivo detl
  while (regd.cod <> valoralto) do begin
    read (mael, regm);
    while (regm.cod <> regd.cod) do
      read (mael, regm);
    while (regm.cod = regd.cod) do begin //se procesan códigos iguales
      regm.cant := regm.cant + regd.cv;
      leer (detl,regd);
    end;
    seek (mael, filepos(mael)-1); //reubica el puntero
    write (mael,regm);
  end;
end;
```

Corte de control

Rebajar datos ordenados hasta q' cambien.
 Procesar un archivo aprovechando forma en que se ordena.
 No el detalle total de archivo.