

# CORTE DE CONTROL Y APAREO

<https://www.udocz.com/apuntes/42866/ejercicios-resueltos-corte-de-control-programacion>

```
/*1. Se tiene un archivo ventas.dat que contiene la info de ventas realizadas por la empresa en el mes. El archivo está ordenado por número de vendedor. Estructura: num factura (entero), numero vendedor (entero), importe de la factura (real) Se quiere que leyendo el archivo calcule la cantidad de ventas y el importe total facturado por cada vendedor. Al finalizar mostrar el importe total facturado por la empresa. */
```

```
struct info
{ int numfactura;
  int numvendedor;
  float importefactura;} //en ventas.dat

//se pasa el archivo abierto f=fopen("ventas.dat", "rb")
void calculos (FILE *f)
{ float importetotal=0;
  float importexvendedor=0;
  int ant=-1;
  info r;
  while (fread(&r, sizeof(info), 1, f))
  {importetotal= importetotal+ r.importefactura;
   while ( ant == r.numvendedor)
   {importexvendedor= importexvendedor + r.importexvendedor;
    ant=r.numvendedor}
   cout << "Importe del vendedor nro." << r.numvendedor << ":" << importexvendedor;}
   cout << "Importe total:" << importetotal << endl;}
```

```
//PERSONAL.dat con 175 operativos
```

```
struct personal
{ int nrolegajo;
  char apeynom[80];
  int valorhora;
  int totalhorastrabajadas; //actualizar sumando
  int totalsueldosacobrar; //actualizar
  char fechadelultimoproceso; }
```

1. Generar un vector llamado ArrayPerso con los datos existentes en el archivo PERSONAL.dat

```
//f=fopen("Personal.dat", "rb")
void lecturapersonal (FILE* f, personal p[], int tam )
{ int i;
  personal p;
  for (i=0; i<175; i++)
  {while (fread(&p, sizeof(personal), 1, f))
```

```

    v[i] = p;}
    fclose(f);}

```

3. Para el control de la existencia del Nro. de Legajo confeccion la función búsqueda con los argumentos necesarios. Si no existe, grabar la información ingresada (nro. legajo - cantidad de horas trabajadas) en el archivo ERROR.dat

```

struct horas
{ int nrolegajo;
  int canthoras;} //Horas.dat

//f=fopen("Horas.dat", "rb")
void actualizacion (personal p[], FILE* f, int tam)
{ FILE *f2 = fopen ("Error.dat", "rb");
  horas r;
  int pos;
  while (fread(&r, sizeof(horas), 1, f))
  { pos=busquedaSecuencial (r.numlegajo, p, tam)
    if (pos>=0)
    {p[pos].totalhorastrabajadas+= r.canthoras;
      p[pos].totalsueldoacobrar= p[pos].totalhorastrabajadas * p[pos].valorhora;}
    else
      fwrite (r, sizeof(horas), 1, f2);}
  f2=close(f);}}

```

5. actualizar PERSONAL.DAT mediante la función GRABAR

```

void actualizar (FILE* f, personal v[], int tam)
{personal p;
  int i;
  { for (i=0; i<tam; i++)
    {p= v[i];
      fwrite (&p, sizeof(personal), 1, f);}
    f=fclose(f);}

```

6. Informar los datos de los legajos que tienen sueldo mayor a 1500 y las horas no son mas de 140. Función listar.

```

void listar (personal p[], int tam)
{ int i;
  for (i=0; i<tam; i++)
  {if (p[i].totalsueldosacobrar>1500 && p[i].totalhorastrabajadas < 140)
    {cout << p[i].nrolegajo << endl;}}}

```