#### PARA COMPTLAR gcc DatosPersonalesB1.c comunes.c principal.c -o compara.exe gcc DatosPersonalesB2.c comunes.c principal.c -o compara.exe gcc DatosPersonalesT1.c comunes.c principal.c -o compara.exe gcc DatosPersonalesB2.c comunes.c principal.c -o compara.exe qcc DatosPersonalesB2.c comunes.c borradoFisico.c -o borradoFisico.exe ESTUCTURA DE LOS DATOS EN MEMORIA struct DatosPersonales char nombre[15]; char apellido[15]; long dni; float salario; }; #DEFINE #define MAX LINEA 256 #define NUM CAMPOS 4 #define BORRADO -1 т1 Т2 B1 B2 ESTUCTURA DE LOS DATOS EN DISCO: T1 ESTRUCTURA DE LOS DATOS EN DISCO: T2 ESTRUCTURA DE LOS DATOS EN DISCO: B1 y B2 Presupone que la información de cada Presupone que la información de cada struct DatosPersonales persona está en una única línea. persona ocupa tres líneas, una por char nombre[15]; NO se permiten nombres ni apellidos char apellido[15]; compuestos (con espacios en blanco) SI se permiten nombres y apellidos long dni; compuestos (con espacios en blanco) }; antonio gomez 1111 1234.56 еj. NOTAS: francisco perez 2222 987.56 antonio jose No es obligatorio utilizar una marca de borrado, eso dependerá del lucia lozano 3333 1350.9 gomez enunciado del problema. 1111 • En el caso B2 la marca de borrado consistirá en poner el campo dni a -1 1234.56 francisco del monte 2222 987.56 Lucia lozano 3333 1350.9 void verFichero(char \*fichero) FILE \*pFichero; FILE \*pFichero; FILE \*pFichero; FILE \*pFichero; struct DatosPersonales persona; struct DatosPersonales persona; struct DatosPersonales persona; struct DatosPersonales persona; char linea[MAX LINEA];

pFichero = fopen(fichero, "rb");

pFichero = fopen(fichero, "rb"); /\*

pFichero = fopen(fichero, "r"); /\*

```
abre fichero paa lectura */
                                          pFichero = fopen(fichero, "r"); /*
                                                                                   /* Comienza a leer datos del fichero
                                                                                                                           abre fichero para lectura */
/* Comienza a leer datos del fichero
                                                                                                                            /* Comienza a leer datos del fichero
                                         abre fichero para lectura */
                                                                                  hasta que llega al final */
hasta que llega al final */
                                          /* Comienza a leer datos del fichero
                                                                                   while (fread (&persona, sizeof (struct
                                                                                                                           hasta que llega al final */
//Tambien serviría un while
                                         hasta que llega al final */
                                                                                  DatosPersonales), 1, pFichero) == 1)
                                                                                                                            while (fread (&persona, size of (struct
(fscanf( )!=EOF)
                                         while (fgets(linea, MAX LINEA,
                                                                                   /* Comprueba que no ha llegado al
                                                                                                                           DatosPersonales),1,pFichero) == 1)
                                         pFichero) !=NULL)
while (fscanf(pFichero, "%s %s %ld
                                                                                  final del fichero */
%f", persona.nombre, persona.apellido,
                                                                                                                              /* Hav que comprobar que no esté
&persona.dni.
                                             //Procesamos nombre
                                                                                      escribirDatosPersonales(persona):
                                                                                                                           marcado como borrado */
&persona.salario) == NUM CAMPOS)
                                             if(linea[strlen(linea)-1]=='\n')
                                                                                  /* Muestra el registro por pantalla */
                                                                                                                              if (persona.dni != BORRADO)
                                               linea[strlen(linea)-11='\0':
                                                                                                                               escribirDatosPersonales(persona);
                                                                                                                           /* Muestra el registro por pantalla */
   escribirDatosPersonales(persona);
                                             strcpv(persona.nombre, linea);
                                                                                   fclose(pFichero);
                                                                                                                            fclose(pFichero);
fclose(pFichero);
                                             //Procesamos apellidos - se
                                         admiten espacios en blanco
                                             fgets(linea, MAX LINEA, pFichero);
                                              if(linea[strlen(linea)-1]=='\n')
                                              linea[strlen(linea)-1]='\0';
                                             strcpy(persona.apellido, linea);
                                             //Procesamos dni
                                             fgets(linea, MAX LINEA, pFichero);
                                             sscanf(linea, "%ld",
                                         &persona.dni);
                                                 //Procesamos salario
                                             fgets(linea, MAX LINEA, pFichero);
                                             sscanf(linea, "%f",
                                         &persona.salario);
                                             escribirDatosPersonales(persona);
                                          fclose(pFichero);}
                                                            long contarRegistros(char *fichero)
                                         long contarRegistros(char *fichero)
long contarRegistros(char *fichero)
                                                                                  long contarRegistros(char *fichero)
                                                                                                                           long contarRegistros(char *fichero)
FILE *pFichero;
                                          FILE *pFichero;
                                                                                   FILE *pFichero;
                                                                                                                            FILE *pFichero;
long nRegistros=0;
                                          long nRegistros=0;
                                                                                   long numeroRegistros;
                                                                                                                            struct DatosPersonales persona;
struct DatosPersonales persona;
                                          struct DatosPersonales persona;
                                                                                                                            long numeroRegistros=0;
                                          char linea[MAX LINEA];
                                                                                   /* Se abre para lectura el fichero */
pFichero = fopen(fichero, "r"); /*
                                                                                   pFichero = fopen(fichero, "rb");
                                                                                                                            /* Se abre para lectura el fichero */
abre fichero paa lectura */
                                          pFichero = fopen(fichero, "r"); /*
                                                                                                                            pFichero = fopen(fichero, "rb");
/* Comienza a leer datos del fichero
                                         abre fichero paa lectura */
                                                                                   /* Nos situamos al final del mismo */
                                          /* Comienza a leer datos del fichero
                                                                                                                            /* Nos situamos al final del mismo */
hasta que llega al final */
                                                                                   fseek(pFichero, 0, SEEK END);
 //Tambien serviría un while
                                         hasta que llega al final */
                                                                                                                            while (fread (&persona, size of (struct
                                         while (fgets (linea, MAX LINEA,
(fscanf(...)!=EOF)
                                                                                   /* ftell devuelve el numero de bytes
                                                                                                                           DatosPersonales),1,pFichero) == 1)
 while (fscanf(pFichero, "%s %s %ld
                                         pFichero) !=NULL)
                                                                                  desde el principio del fichero
%f", persona.nombre, persona.apellido,
                                                                                     hasta la posicion actual que es el
                                                                                                                              /*Usar fseek v ftell contaria
                   &persona.dni,
                                             nRegistros++;
                                                                                  final del fichero. */
                                                                                                                           tambien los marcados para borrar*/
                                                                                   numeroRegistros =
&persona.salario) == NUM CAMPOS)
                                                                                                                              /*solo contabilizamos si el
                                          fclose(pFichero);
                                                                                  ftell(pFichero)/sizeof(struct
                                                                                                                           registro no esta marcado para borrar*/
                                                                                                                              if(persona.dni !=BORRADO)
                                         return(nRegistros/NUM CAMPOS);
                                                                                  DatosPersonales);
   nRegistros++;
                                                                                   fclose(pFichero);
                                                                                                                                 numeroRegistros++;
                                                                                   return numeroRegistros;
                                                                                                                            fclose(pFichero);
 fclose(pFichero);
                                                                                                                            return numeroRegistros;
 return (nRegistros);
```

#### void anadirRegistro(char \*fichero, struct DatosPersonales persona) void anadirRegistro(char \*fichero, void anadirRegistro(char \*fichero, void anadirRegistro(char \*fichero, struct DatosPersonales persona) struct DatosPersonales persona) struct DatosPersonales persona) FILE \*pFichero: FILE \*pFichero; FILE \*pFichero; /\* abre el fichero para anadir \*/ pFichero = fopen(fichero, "ab"): pFichero = fopen(fichero, "a"); /\* pFichero = fopen(fichero, "a"); /\* /\* guarda los datos en el fichero \*/ abre el fichero para anadir \*/ abre el fichero para anadir \*/ fwrite (&persona, sizeof (struct DatosPersonales), 1, pFichero): fprintf(pFichero, "%s %s %ld %.3f\n", fprintf(pFichero. fclose(pFichero): persona.nombre, persona.apellido, "%s\n%s\n%ld\n%.3f\n", persona.nombre, persona dni. persona.apellido. persona.salario); /\* guarda los datos persona.dni. en el fichero \*/ persona.salario); /\* guarda los datos fclose(pFichero); en el fichero \*/ fclose(pFichero); struct DatosPersonales registro i(char \*fichero, int i) struct DatosPersonales registro i(char struct DatosPersonales registro i(char struct DatosPersonales registro i(char struct DatosPersonales registro i(char \*fichero, int i) \*fichero, int i) \*fichero, long i) \*fichero, long i) FILE \*pFichero; FILE \*pFichero; FILE \*pFichero; FILE \*pFichero; struct DatosPersonales aux: long cont=0: long i: long i: struct DatosPersonales persona; struct DatosPersonales persona; struct DatosPersonales persona; char nombre[MAX LINEAl: /\* Se abre para lectura el fichero \*/ pFichero = fopen(fichero, "r"); char apellidos[MAX LINEA]; pFichero = fopen(fichero, "rb"); /\* Se abre para lectura el fichero \*/ char dni[MAX LINEA]; pFichero = fopen(fichero, "rb"); for(j=0; j<i; j++) char salario[MAX LINEA]; /\* Nos desplazamos al final del registro i-1, desde el principio while(cont<i) fscanf(pFichero, "%s %s %ld %f". pFichero = fopen(fichero, "r"); del fichero con la funcion fseek\*/ persona.nombre, persona.apellido, fseek(pFichero, (i-1)\*sizeof(struct /\*Usar fseek y ftell contaria &persona.dni, &persona.salario); for(j=0; j<i; j++) DatosPersonales), SEEK SET); tambien los marcados para borrar\*/ fread(&persona, sizeof(struct fclose(pFichero); faets (nombre, MAX LINEA, /\* Se lee el siguiente registro \*/ DatosPersonales), 1, pFichero); pFichero): fread(&aux.sizeof(struct if (persona.dni !=BORRADO) /\* devuelve el registro leido \*/ fgets(apellidos, MAX LINEA, DatosPersonales), 1, pFichero); cont++; return persona; fgets (dni, MAX LINEA, pFichero); fclose(pFichero); fgets(salario, MAX LINEA, /\* devuelve el registro leido \*/ printf("\ncont: %d", cont); pFichero); fclose(pFichero): return aux: /\* devuelve el registro leido \*/ //Procesamos nombre return persona: if (nombre[strlen(nombre)-1]=='\n') nombre[strlen(nombre)-1]='\0'; strcpy(persona.nombre, nombre); //Procesamos apellidos - se admiten espacios en blanco if (apellidos[strlen(apellidos)-1]=='\n') apellidos[strlen(apellidos)strcpy(persona.apellido, apellidos); //Procesamos dni

sscanf(dni, "%ld", &persona.dni);

```
sscanf(salario, "%f",
&persona.salario);

fclose(pFichero);

/* devuelve el registro leido */
return persona;
```

## int buscarporDni(char \*fichero, long dni buscar, struct DatosPersonales \*persona)

```
int buscarporDni(char *fichero, long
dni, struct DatosPersonales *persona)
FILE *pFichero;
int encontrado = 0; /* variables
auxiliares para la busqueda */
int cont:
struct DatosPersonales auxiliar:
/* abre fichero para lectura */
pFichero = fopen(fichero, "r");
while (fscanf(pFichero, "%s %s %ld
%f", auxiliar.nombre,
auxiliar.apellido.
          &auxiliar.dni.
&auxiliar.salario) == NUM CAMPOS)
   if (auxiliar.dni == dni) /* ha
encontrado el registro */
     encontrado = 1:
      *persona = auxiliar; /* almacena
en persona el registro encontrado */
fclose(pFichero); /* se cierra el
fichero */
return encontrado;
```

```
int buscarporDni(char *fichero, long
dni buscar, struct DatosPersonales
*persona)
FILE *pFichero;
int salir = 0:
int encontrado = 0; /* variables
auxiliares para la busqueda */
int cont:
struct DatosPersonales auxiliar;
char nombre[MAX LINEA];
char apellidos[MAX LINEA];
char dni[MAX LINEA];
char salario[MAX LINEA];
pFichero = fopen(fichero, "r"); /*
abre fichero para lectura */
while ((fgets(nombre, MAX LINEA,
pFichero)!=NULL) &&!encontrado)
    //Leemos apellidos - se admiten
espacios en blanco
   fgets(apellidos, MAX LINEA,
pFichero):
    //Leemos dni
    fgets(dni, MAX_LINEA, pFichero);
   sscanf(dni, "%ld", &auxiliar.dni);
   //Leemos salario
   fgets(salario, MAX LINEA,
pFichero):
   if (auxiliar.dni == dni buscar) /*
ha encontrado el registro *7
     encontrado = 1:
     //Procesamos nombre
     if (nombre[strlen(nombre)-
1]=='\n')
       nombre[strlen(nombre)-1]='\0';
     strcpy(auxiliar.nombre, nombre);
     //Procesamos apellidos - se
admiten espacios en blanco
     if (apellidos[strlen(apellidos)-
1]=='\n')
       apellidos[strlen(apellidos)-
```

```
La función es la misma para B1 y B2.Al ser el dni el campo que se marca para
borrar, no haría falta comprobar si dni==BORRADO en el caso de ficheros B2
int buscarporDni(char *fichero, long dni, struct DatosPersonales *persona)
 FILE *pFichero;
 int encontrado = 0:
 struct DatosPersonales auxiliar;
 /* abre fichero para lectura */
 pFichero = fopen(fichero, "rb");
 while (fread(&auxiliar, sizeof(struct DatosPersonales), 1, pFichero) == 1)
    if (auxiliar.dni == dni)
    /* ha encontrado el registro*/
      encont.rado = 1:
      *persona = auxiliar; /* almacena en persona el registro encontrado */
 fclose(pFichero); /* se cierra el fichero */
 return encontrado;
```

```
1]='(0';
    strcpy(auxiliar.apellido,
    apellidos);

    //Procesamos el salario
    sscanf(salario, "%f",
    &auxiliar.salario);

    //almacena en persona el
    registro encontrado
        *persona = auxiliar;
    }
}

fclose(pFichero); /* se cierra el
fichero */
    return encontrado;
}

int mostrarporNombre(che
```

### int mostrarporNombre(char \*fichero, char \*auxNombre)

```
int mostrarporNombre(char *fichero,
char *auxNombre)
FILE *pFichero;
struct DatosPersonales auxiliar:
int encontrado = 0:
pFichero = fopen(fichero, "r"); /*
abre fichero para lectura */
//Tambien serviría un while
(fscanf(...)!=EOF),
while (fscanf(pFichero, "%s %s %ld
%f", auxiliar.nombre,
auxiliar.apellido,
          &auxiliar.dni,
&auxiliar.salario) == NUM CAMPOS)
  if (strcmp(auxiliar.nombre,
auxNombre) == 0)
  /* se ha encontrado un registro con
ese nombre */
escribirDatosPersonales(auxiliar); /*
se escriben sus datos */
    encontrado = 1;
fclose(pFichero); /* se cierra el
fichero */
return encontrado;
```

```
int mostrarporNombre(char *fichero,
char *auxNombre)
FILE *pFichero;
struct DatosPersonales auxiliar:
int encontrado = 0:
char linea[MAX LINEA];
pFichero = fopen(fichero, "r"); /*
abre fichero para lectura */
while (fgets(linea, MAX LINEA,
pFichero)!=NULT)
     //Procesamos nombre
   if(linea[strlen(linea)-1]=='\n')
     linea[strlen(linea)-1]='\0';
    strcpy(auxiliar.nombre, linea);
    //Procesamos apellidos - se
admiten espacios en blanco
   faets(linea, MAX LINEA, pFichero);
    if(linea[strlen(linea)-1]=='\n')
     linea[strlen(linea)-1]='\0';
    strcpy(auxiliar.apellido, linea);
    //Procesamos dni
    fgets(linea, MAX LINEA, pFichero);
   sscanf(linea, "%ld",
&auxiliar.dni);
    //Procesamos el salario
    fgets(linea, MAX LINEA, pFichero);
   sscanf(linea, "%f",
&auxiliar.salario):
    //se ha encontrado un registro con
```

```
int mostrarporNombre(char *fichero,
char *auxNombre)
FILE *pFichero;
struct DatosPersonales auxiliar:
int encontrado = 0:
 /* abre fichero para lectura */
pFichero = fopen(fichero, "rb");
while (fread (&auxiliar, size of (struct
DatosPersonales), 1, pFichero) == 1)
  if (strcmp(auxiliar.nombre,
auxNombre) == 0)
  /* se ha encontrado un registro con
ese nombre */
escribirDatosPersonales(auxiliar); /*
se escriben sus datos */
    encontrado = 1;
fclose(pFichero); /* se cierra el
fichero */
return encontrado;
```

```
int mostrarporNombre(char *fichero,
char *auxNombre)
FILE *pFichero;
struct DatosPersonales auxiliar:
int encontrado = 0:
/* abre fichero para lectura */
pFichero = fopen(fichero, "rb");
while (fread (&auxiliar, size of (struct
DatosPersonales).1.pFichero) == 1)
/* se leen todos los registros */
  if (strcmp(auxiliar.nombre,
auxNombre) == 0
      && auxiliar.dni!=BORRADO)
  /* se ha encontrado un registro con
ese nombre v no está marcado */
escribirDatosPersonales(auxiliar); /*
se escriben sus datos */
    encont.rado = 1:
fclose(pFichero); /* se cierra el
fichero */
return encontrado;
```

```
ese nombre
                                           if (strcmp(auxiliar.nombre,
                                        auxNombre) == 0)
                                        escribirDatosPersonales(auxiliar): /*
                                        se escriben sus datos */
                                                         encontrado = 1:
                                         fclose(pFichero); /* se cierra el
                                        fichero */
                                         return encontrado:
                                                      int actualizarporDni(char* fichero, long dni)
int actualizarporDni(char* fichero.
                                        int mostrarporNombre(char *fichero.
                                                                                 int actualizarporDni(char* fichero, long dni)
long dni)
                                        char *auxNombre)
                                                                                  FILE *pFichero:
 FILE *pFicherol, *pFichero2;
                                         FILE *pFichero:
                                                                                  struct DatosPersonales aux;
 struct DatosPersonales aux;
                                         struct DatosPersonales auxiliar:
                                                                                  int encontrado =0;
 int encontrado = 0:
                                         int encontrado = 0:
                                         char linea[MAX LINEA];
                                                                                  /* Se abre para lectura y escritura */
 /* Se abre para lectura el fichero
                                                                                  pFichero = fopen(fichero, "r+b"):
original */
                                         pFichero = fopen(fichero, "r"); /*
pFicherol = fopen(fichero, "r");
                                        abre fichero para lectura */
                                                                                  /* Se recorre el fichero */
                                                                                  while(fread(&aux, sizeof(struct DatosPersonales),1,pFichero) == 1 )
/* Fichero temporal para volcar los
                                         while (fgets(linea, MAX LINEA,
                                        pFichero)!=NULL)
registros que no se borran */
                                                                                    /* Se comprueba si el registro tiene ese dni*/
pFichero2 = fopen("temporal", "w");
                                                                                    if (aux dni == dni)
                                              //Procesamos nombre
 /* Se recorre el fichero original v
                                             if(linea[strlen(linea)-1]=='\n')
                                                                                      aux = introducirDatosPersonales();
                                              linea[strlen(linea)-1]='\0';
los registros que no hav que borrar
                                                                                      encontrado = 1;
   se pasan al fichero temporal */
                                             strcpy(auxiliar.nombre, linea);
                                                                                      /* Nos situamos al principio del registro encontrado */
 //Tambien serviría un while
                                             //Procesamos apellidos - se
                                                                                      fseek (pFichero, -(int) sizeof (struct DatosPersonales), SEEK CUR);
(fscanf(...)!=EOF)
while (fscanf (pFicherol, "%s %s %ld
                                        admiten espacios en blanco
                                                                                      /* Se reescribe el registro */
                                            fgets(linea, MAX LINEA, pFichero);
                                                                                      fwrite(&aux, sizeof(struct DatosPersonales), 1, pFichero);
       aux.nombre, aux.apellido,
                                             if(linea[strlen(linea)-1]=='\n')
                                                                                      //La siguiente orden es necesario para que en windows
&aux.dni, &aux.salario) == NUM CAMPOS)
                                              linea[strlen(linea)-1]='\0';
                                                                                      //se actualicen los flags de FILE*
                                             strcpy(auxiliar.apellido, linea);
                                                                                      //En Linux, no es necesario, pero se puede dejar
                                                                                      //El estándar requiere fflush para hacer fread despues de fwrite
  if (aux.dni == dni)
                                             //Procesamos dni
                                                                                      fflush (pFichero);
                                             fgets(linea, MAX LINEA, pFichero);
         allx =
                                            sscanf(linea, "%ld",
introducirDatosPersonales();
         encontrado = 1;
                                         &auxiliar.dni);
                                                                                    /* Se cierra el fichero*/
                                                                                   fclose(pFichero);
  fprintf(pFichero2, "%s %s %ld
                                             //Procesamos el salario
%.3f\n", aux.nombre, aux.apellido,
                                             fgets(linea, MAX LINEA, pFichero);
                                                                                   return(encontrado);
                                            sscanf(linea, "%f",
aux.dni, aux.salario);
                                        &auxiliar.salario);
                                             //se ha encontrado un registro con
 /* Se cierran los ficheros */
```

fclose(pFicherol):

fclose(pFichero2);

ese nombre

auxNombre) == 0)

if (strcmp(auxiliar.nombre,

```
/* Se borra el fichero original */
 remove(fichero):
                                         escribirDatosPersonales(auxiliar): /*
/* Se renombra el temporal con el
                                         se escriben sus datos */
nombre del original */
                                                          encontrado = 1;
rename ("temporal", fichero):
return (encontrado):
                                          fclose(pFichero); /* se cierra el
                                         fichero */
                                         return encontrado;
                                                         int borrarporDni(char *fichero, long dni)
int borrarporDni(char *fichero, long
                                         int borrarporDni(char *fichero, long
                                                                                  int borrarporDni(char *fichero, long
                                                                                                                           int borrarporDni(char *fichero, long
dni)
                                         dni buscar)
                                                                                  dni buscar)
                                                                                                                           dni)
FILE *pFichero1, *pFichero2;
                                         FILE *pFichero1, *pFichero2;
                                                                                   FILE *pFichero1. *pFichero2:
                                                                                                                            FILE *pFichero:
struct DatosPersonales aux;
                                          struct DatosPersonales persona;
                                                                                   struct DatosPersonales aux;
                                                                                                                            struct DatosPersonales aux;
int borrado = 0:
                                          char linea[MAX LINEA];
                                                                                   int borrado=0:
                                                                                                                            int encontrado = 0:
                                          int borrado = \overline{0}:
 /* Se abre para lectura el fichero
                                                                                   /* Se abre para lectura el fichero
                                                                                                                            /* Se abre para lectura v escritura
original */
                                          /* Se abre para lectura el fichero
                                                                                  original */
                                                                                                                           el fichero */
pFicherol = fopen(fichero, "r");
                                                                                   pFicherol = fopen(fichero, "rb");
                                         original */
                                                                                                                            pFichero = fopen(fichero, "r+b");
                                         pFicherol = fopen(fichero, "r");
 /* Fichero temporal para volcar los
                                                                                   /* Fichero temporal para volcar los
                                                                                                                            /* Se recorre el fichero original v
registros que no se borran */
                                          /* Fichero temporal para volcar los
                                                                                  registros que no se borran */
                                                                                                                           los registros a borrar se
pFichero2 = fopen("temporal", "w");
                                         registros que no se borran */
                                                                                   pFichero2 = fopen("temporal", "wb");
                                                                                                                               reescriben marcandolos como
                                         pFichero2 = fopen("temporal", "w");
                                                                                                                           horrados */
 /* Se recorre el fichero original y
                                                                                   /* Se recorre el fichero original v
                                                                                                                            while ((fread (&aux, size of (struct
los registros que no hay que borrar
                                          /* Se recorre el fichero original v
                                                                                  los registros que no hay que borrar
                                                                                                                           DatosPersonales),1,pFichero) == 1
   se pasan al fichero temporal */
                                                                                      se pasan al fichero temporal */
                                         los registros que no hay que borrar
                                                                                                                           ) &&!encontrado)
 //Tambien serviría un while
                                             se pasan al fichero temporal */
(fscanf( )!=EOF)
                                                                                   while (fread (&aux, size of (struct
                                                                                                                              /* Se comprueba si el registro
while (fscanf (pFicherol, "%s %s %ld
                                           while (fgets(linea, MAX LINEA,
                                                                                  DatosPersonales), 1, pFichero1) == 1)
                                                                                                                           tiene ese nombre v no está marcado */
                                                                                                                              if (aux dni == dni)
                                         pFichero1)!=NULL)
       aux.nombre, aux.apellido,
                                                                                     /* Se comprueba si el registro
&aux.dni, &aux.salario) == NUM CAMPOS)
                                             //Procesamos nombre
                                                                                  tiene ese dni*/
                                                                                                                                /* Se marca como borrado */
                                             if(linea[strlen(linea)-1]=='\n')
                                                                                     if (dni buscar!=aux.dni)
                                                                                                                                aux.dni = BORRADO:
   if (aux.dni != dni)
                                               linea[strlen(linea)-1]='\0';
                                                                                       fwrite(&aux, sizeof(struct
                                                                                                                                /* Nos situamos al principio del
   fprintf(pFichero2, "%s %s %ld
                                             strcpy(persona.nombre, linea);
                                                                                  DatosPersonales), 1, pFichero2);
                                                                                                                           registro encontrado */
%.3f\n", aux.nombre, aux.apellido,
                                                                                     else
                                                                                                                                fseek (pFichero. -
                                             //Procesamos apellidos - se
                                                                                                                           (int) sizeof (struct DatosPersonales).
aux.dni, aux.salario);
                                                                                       horrado = 1:
  else
                                         admiten espacios en blanco
                                                                                                                           SEEK CUR):
   borrado = 1;
                                             fgets(linea, MAX LINEA,
                                                                                    /* Se cierran los ficheros */
                                                                                                                                /* Se reescribe el registro */
                                                                                    fclose(pFicherol);
                                                                                                                                fwrite(&aux, sizeof(struct
                                         pFicherol):
                                              if(linea[strlen(linea)-1]=='\n')
                                                                                    fclose(pFichero2);
                                                                                                                           DatosPersonales), 1, pFichero);
 /* Se cierran los ficheros */
                                               linea[strlen(linea)-1]='\0';
                                                                                                                                //La siguiente orden es necesario
                                                                                                                           para que en windows
 fclose(pFicherol);
                                             strcpy(persona.apellido, linea);
                                                                                   /* Se borra el fichero original */
 fclose(pFichero2);
                                                                                   remove(fichero):
                                                                                                                                //se actualicen los flags de
                                             //Procesamos dni
 /* Se borra el fichero original */
                                             fgets(linea, MAX LINEA,
                                                                                   /* Se renombra el temporal con el
                                                                                                                                //En Linux, no es necesario, pero
 remove(fichero):
                                                                                  nombre del original */
                                         pFicherol):
                                                                                                                           se puede dejar
                                             sscanf(linea, "%ld",
                                                                                   rename ("temporal", fichero);
                                                                                                                                fflush (pFichero);
 /* Se renombra el temporal con el
                                         &persona.dni):
                                                                                   return (borrado):
                                                                                                                                /* Se ha borrado al menos un
nombre del original */
                                                                                                                           registro */
```

encontrado = 1;

//Procesamos salario

rename("temporal", fichero);

```
return (borrado):
                                            fgets(linea, MAX LINEA,
                                        pFicherol);
                                            sscanf(linea, "%f",
                                                                                                                            /* Se cierra el fichero */
                                        &persona.salario);
                                                                                                                            fclose(pFichero):
                                                                                                                            return encontrado;
                                                //el registro no contiene
                                        dni buscar
                                            if (persona.dni != dni buscar)
                                               fprintf(pFichero2.
                                        "%s\n%s\n%ld\n%.3f\n", persona.nombre,
                                        persona.apellido, persona.dni,
                                        persona salario):
                                            else
                                              borrado = 1;
                                          /* Se cierran los ficheros */
                                          fclose(pFicherol):
                                          fclose(pFichero2);
                                          /* Se borra el fichero original */
                                          remove(fichero):
                                          /* Se renombra el temporal con el
                                        nombre del original */
                                          rename("temporal", fichero);
                                          return (borrado):
                                           struct DatosPersonales* ficheroAVector(char* fichero, long* nEle)
struct DatosPersonales*
                                        struct DatosPersonales*
                                                                                 struct DatosPersonales*
                                                                                                                          struct DatosPersonales*
                                        ficheroAVector(char* fichero, long*
                                                                                 ficheroAVector(char* fichero, long*
                                                                                                                          ficheroAVector(char* fichero, long*
ficheroAVector(char* fichero.
long* nEle)
                                           FILE *pFichero;
                                                                                    struct DatosPersonales* V;
                                                                                                                             struct DatosPersonales* V;
  FILE *pFichero;
                                           struct DatosPersonales persona, *V;
                                                                                    FILE* pFichero;
                                                                                                                             FILE* pFichero;
  struct DatosPersonales persona, *V;
                                           long i=0;
                                                                                                                             long registros, i;
                                           char linea[MAX LINEA];
                                                                                    *nEle = contarRegistros(fichero);
   * nEle = contarRegistros(fichero);
                                                                                    V = reservarVector(*nEle);
                                                                                                                             //Contamos registros, incluidos los
  V = reservarVector(*nEle);
                                            *nEle = contarRegistros(fichero);
                                                                                    pFichero = fopen(fichero, "rb");
                                                                                                                          borrados
  pFichero = fopen(fichero, "r");
                                           V = reservarVector(*nEle);
                                                                                                                             //contarRegistros solo cuenta los
      /*Leemos todo el fichero*/
                                           pFichero = fopen(fichero, "r");
                                                                                    /*Leemos todo el fichero (mas
                                                                                                                          activos, por lo que no es valida
   //Tambien serviría un while
                                                                                 rapido que elemento a elemento) */
                                                                                                                            pFichero = fopen(fichero, "rb");
(fscanf(...) !=EOF)
                                           /*Leemos todo el fichero*/
                                                                                    fread(V, sizeof(struct
                                                                                                                             fseek (pFichero, 0, SEEK END);
  while (fscanf(pFichero, "%s %s %ld
                                                                                 DatosPersonales), *nEle, pFichero);
                                           while (fgets(linea, MAX LINEA,
                                                                                                                             registros =
%f", persona.nombre, persona.apellido,
                                        pFichero) !=NULL)
                                                                                                                          ftell(pFichero)/sizeof(struct
                                                                                    fclose(pFichero);
                   &persona.dni.
                                                                                    return(V);
                                                                                                                          DatosPersonales);
&persona.salario) == NUM CAMPOS)
                                            //Procesamos nombre
                                            if(linea[strlen(linea)-1]=='\n')
                                                                                                                            V = reservarVector(registros);
      V[i]=persona;
                                              linea[strlen(linea)-1]='\0';
                                            strcpy(persona.nombre, linea);
                                                                                                                             /*Leemos todo el fichero (mas
                                                                                                                          rapido que elemento a elemento) */
  fclose(pFichero);
                                            //Procesamos apellidos - se
                                                                                                                             fseek(pFichero, 0, SEEK SET);
  return(V);
                                        admiten espacios en blanco
                                                                                                                             fread(V, sizeof(struct
                                            fgets(linea, MAX LINEA, pFichero);
                                                                                                                          DatosPersonales), registros,
                                             if(linea[strlen(linea)-1]=='\n')
                                                                                                                          pFichero);
                                              linea[strlen(linea)-1]='\0';
                                                                                                                            fclose(pFichero);
```

```
strcpy(persona.apellido, linea);
                                                                                                                           //Nos quedamos solamente con los
                                            //Procesamos dni
                                                                                                                         registros no borrados
                                            fgets(linea, MAX LINEA, pFichero);
                                                                                                                           for(i=0, *nEle=0; i<registros; i++)
                                            sscanf(linea, "%ld",
                                        &persona.dni):
                                                                                                                               printf("\ncopiando registro: %ld
                                                                                                                         nEle: %ld", i, *nEle);
                                            //Procesamos salario
                                                                                                                                   if(V[i].dni!=BORRADO)
                                            fgets(linea, MAX LINEA, pFichero);
                                            sscanf(linea, "%f",
                                                                                                                                         V[*nEle] = V[i];
                                        &persona.salario);
                                                                                                                                     (*nEle)++; //Parentesis
                                                                                                                         necesarios
                                            V[i]=persona;
                                           i++;
                                                                                                                           V= (struct DatosPersonales*)
                                                                                                                         realloc(V, (*nEle) *sizeof(struct
                                           fclose(pFichero);
                                                                                                                        DatosPersonales));
                                           return(V);
                                                                                                                           return(V):
                                       void vectorAFichero(char* fichero, struct DatosPersonales* V, long nEle)
void vectorAFichero(char* fichero,
                                        void vectorAFichero(char* fichero,
                                                                                void vectorAFichero(char* fichero, struct DatosPersonales* V, long nEle)
struct DatosPersonales* V, long nEle)
                                        struct DatosPersonales* V, long nEle)
                                                                                   FILE* pFichero;
                                           FILE* pFichero;
  FILE* pFichero;
                                                                                   pFichero = fopen(fichero, "wb");
  long i=0;
                                           long i=0;
                                                                                   fwrite(V, sizeof(struct DatosPersonales), nEle, pFichero);
  pFichero = fopen(fichero, "w");
                                           pFichero = fopen(fichero, "wt");
                                                                                   fclose(pFichero);
  for (i=0; i<nEle; i++)
                                           for (i=0; i<nEle; i++)
  fprintf(pFichero, "%s %s %ld
                                           fprintf(pFichero,
%.3f\n", V[i].nombre, V[i].apellido,
                                        "%s\n%s\n%ld\n%.3f\n", V[i].nombre,
            V[i].dni, V[i].salario);
                                        V[i].apellido,
  fclose(pFichero);
                                                    V[i].dni, V[i].salario);
                                           fclose(pFichero);
                                                             int borrarMarcados(char *fichero)
                                                                                                                         int borrarMarcados (char *fichero)
                                                                                                                         FILE *pFichero1, *pFichero2;
                                                                                                                          struct DatosPersonales aux;
                                                                                                                          int borrados = 0;
                                                                                                                          /* Se abre para lectura el fichero
                                                                                                                         original */
                                                                                                                         pFicherol = fopen(fichero, "rb");
                                                                                                                          /* Se abre para escritura el fichero
                                                                                                                         temporal para volcar los registros
                                                                                                                          que no se borran */
                                                                                                                          pFichero2 = fopen("temporal", "wb");
                                                                                                                          /* Se recorre el fichero original y
                                                                                                                         los registros no marcados
                                                                                                                            se pasan al fichero temporal */
                                                                                                                          while (fread (&aux, sizeof (struct
                                                                                                                         DatosPersonales),1,pFichero1) == 1)
```

```
if(aux.dni != BORRADO)
 /* Si no está marcado el registro
se pasa al temporal */
 fwrite(&aux, sizeof(struct
DatosPersonales), 1, pFichero2);
  /* al menos hay uno marcado */
  borrados++;
 /* Se cierran los ficheros */
 fclose(pFicherol);
 fclose(pFichero2);
 /* Se borra el fichero original */
 remove(fichero);
 /* Se renombra el temporal con el
nombre del original */
 rename ("temporal", fichero);
 return borrados;
```

# struct DatosPersonales introducirDatosPersonales()

```
struct DatosPersonales introducirDatosPersonales()
struct DatosPersonales aux;
/*Introduccion de datos */
printf(" DNI : ");
scanf("%ld", &aux.dni);
getchar();
printf("Nombre : ");
fgets(aux.nombre, MAX LINEA, stdin);
aux.nombre[strlen(aux.nombre)-1]='\0';
printf("Apellido :");
fgets(aux.apellido, MAX LINEA, stdin);
aux.apellido[strlen(aux.apellido)-1]='\0';
printf("Salario:");
scanf("%f", &aux.salario);
getchar();
 /* devolucion de los datos de la persona */
return aux;
```

```
void escribirDatosPersonales(struct DatosPersonales aux)
void escribirDatosPersonales (struct DatosPersonales aux)
printf("DNI: %ld\n\tNombre: %s Apellido: %s Salario: %.3f\n",
        aux.dni, aux.nombre, aux.apellido, aux.salario);
                                                            int existeFichero(char *fichero)
int existeFichero(char *fichero)
FILE *pFichero;
pFichero = fopen(fichero, "r"); /* abre fichero para lectura */
if (pFichero == NULL) /* el fichero no existe */
  return 0;
 else /* el fichero existe */
  fclose (pFichero);
  return 1;
                                                             long contarBytes (char *fichero)
long contarBytes(char *fichero)
 FILE* f;
 long tam;
 if((f=fopen(fichero, "r")) ==NULL)
   fprintf(stderr, "\nError: no puedo abrir <%s>", fichero);exit(-1);
 if(fseek(f, OL, SEEK END))
    fprintf(stderr, "\nError: no puedo usar <%s>", fichero); exit(-1);
 tam = ftell(f);
 fclose(f);
 return(tam);
                                                   struct DatosPersonales* reservarVector(long nEle)
struct DatosPersonales* reservarVector(long nEle)
  struct DatosPersonales* V=NULL;
  if((V=(struct DatosPersonales*)malloc(nEle*sizeof(struct DatosPersonales))) ==NULL)
     printf("\nError: No se ha podido reservar la memoria");
     exit(-1);
  return(V);
```

```
void liberarVector(struct DatosPersonales** V)
void liberarVector(struct DatosPersonales** V)
  free(*V);
  *V=NULL;
                                              void incrementarSalarios(char* fichero, float incremento)
void incrementarSalarios(char* fichero, float incremento)
  struct DatosPersonales * V;
  long nEle;
  FILE* pFichero;
  long i;
  /*Para reducir los accesos a disco trabajamos con un vector*/
  V = ficheroAVector(fichero, &nEle);
  /*Actualizamos los datos en el vector*/
  for(i=0; i<nEle; i++)
    V[i].salario += incremento;
  /*Escribimos el vector en el fichero*/
  vectorAFichero(fichero, V, nEle);
  /*liberamos el vector*/
  liberarVector(&V);
```