Comparativa texto y binario

T1	T2	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 persona está en una única línea. NO se permiten nombres ni apellidos compuestos (con espacios en blanco) ej. 1111 antonio gomez 2231.300 2222 francisco perez 2345.500 3333 lucia lozano 2111.420	Presupone que la información de cada persona ocupa cuatro líneas, una por campo. SI se permiten nombres y apellidos compuestos (con espacios en blanco) ej. 1111 antonio jose gomez 1234.220 2222 francisco del monte 1345.650 3333 lucia lozano 956.800	 char nombre[15]; char apellido[15]; long dni; //clave única para cada float salario; }; NOTAS: No es obligatorio utilizar un dependerá del enunciado de En el caso B1 suponemos que (no hay borrados) 	na marca de borrado, eso

```
PARA COMPILAR
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 DatosPersonalesT2.c comunes.c principal.c -o compara.exe
gcc comunes.c borradoFisico.c DatosPersonalesB2.c –o borradoFisico.exe
                             ESTUCTURA DE LOS DATOS EN MEMORIA
struct DatosPersonales
 long dni; //clave unica para cada registro
 char nombre[15];
 char apellido[15];
 float salario;
                                             #DEFINE
#define MAX LINEA 256
#define NUM CAMPOS 4
#define BORRADO -1
```

Funciones comunes auxiliares

int existeFichero(char *nombreFichero) int existeFichero(char *nombreFichero) FILE *pFichero; pFichero = fopen(nombreFichero, "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 *nombreFichero)

```
long contarBytes(char *nombreFichero)
{ FILE* f;
 long tam;
 if((f=fopen(nombreFichero, "r"))==NULL)
  fprintf(stderr, "\nError: no puedo abrir <%s>", nombreFichero);exit(-1);
 if(fseek(f, OL, SEEK_END))
  fprintf(stderr, "\nError: no puedo usar <%s>", nombreFichero); exit(-1);
 tam = ftell(f);
 fclose(f);
 return(tam);
```

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);
  limpiarLinea(aux.nombre);
  printf("Apellido :");
  fgets(aux.apellido, MAX LINEA, stdin);
  limpiarLinea(aux.apellido);
  printf("Salario:");
  scanf("%f", &aux.salario);
  getchar();
/* devolución de los datos de la persona */
return aux;
```

```
void escribirDatosPersonales(struct DatosPersonales aux)
void escribirDatosPersonales(struct DatosPersonales aux)
{
   printf("DNI: %Id\n\tNombre: <%s> Apellido(s): <%s> Salario: %.3f\n",
   aux.dni, aux.nombre, aux.apellido, aux.salario);
}
```

```
void limpiarLinea(char* linea)
void limpiarLinea(char* linea)
{
  if(linea[strlen(linea)-1]=='\n')
    linea[strlen(linea)-1] = '\0';
}
```

```
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;
```

Funciones de lectura

void verFichero(char *fichero) void verFichero(char void verFichero(char* void verFichero(char* void verFichero(char *nombreFichero) nombreFichero) nombreFichero) *nombreFichero) FILE *pFichero; FILE *pFichero; FILE *pFichero; FILE *pFichero; struct DatosPersonales persona; struct DatosPersonales persona; struct DatosPersonales persona; struct DatosPersonales persona; /* abre fichero para lectura */ char linea[MAX_LINEA]; pFichero = fopen(nombreFichero, /* abre fichero para lectura */ pFichero = fopen(nombreFichero, pFichero = fopen(nombreFichero, "rb"): "rb"); /* abre fichero para lectura */ "r"); pFichero = fopen(nombreFichero, "r"); /* lee datos del fichero de uno en /* lee datos del fichero uno a uno /* Lee datos del fichero hasta que /*Lee datos del fichero hasta que uno hasta que llega al final */ hasta que llega al final */ llega al final llega al final */ while(fread(&persona,sizeof(struct while(fread(&persona,sizeof(struct Tambien serviría un while while (fgets(linea, MAX LINEA, DatosPersonales),1,pFichero) == 1) DatosPersonales),1,pFichero) == 1) (fscanf(...)!=EOF)*/ pFichero)!=NULL) /* Comprueba que no ha llegado al while (fscanf(pFichero, "%ld %s %s final del fichero */ /* Hay que comprobar que no esté %f", &persona.dni, persona.nombre, //Procesamos dni marcado como borrado */ persona.apellido, sscanf(linea, "%ld", &persona.dni); escribirDatosPersonales(persona); if (persona.dni != BORRADO) &persona.salario)==NUM CAMPOS) /* Muestra el registro por pantalla */ escribirDatosPersonales(persona); //Procesamos nombre /* Muestra el registro por pantalla */ escribirDatosPersonales(persona); fgets(linea, MAX_LINEA, pFichero); fclose(pFichero); limpiarLinea(linea); fclose(pFichero); fclose(pFichero); strcpv(persona.nombre, linea); //Procesamos apellidos - se admiten espacios en blanco fgets(linea, MAX LINEA, pFichero); limpiarLinea(linea); strcpy(persona.apellido, linea); //Procesamos salario fgets(linea, MAX LINEA, pFichero); sscanf(linea, "%f", &persona.salario); escribirDatosPersonales(persona); fclose(pFichero);

long contarRegistros(char *fichero) long contarRegistros(char long contarRegistros(char* long contarRegistros(char* long contarRegistros(char *nombreFichero) nombreFichero) nombreFichero) *nombreFichero) FILE *pFichero; FILE *pFichero; FILE *pFichero; FILE *pFichero; long nRegistros=0; long numeroRegistros; struct DatosPersonales persona; long nRegistros=0; struct DatosPersonales persona; struct DatosPersonales persona; long numeroRegistros=0; /* abre fichero paa lectura */ char linea[MAX LINEA]; /* Se abre para lectura el fichero pFichero = fopen(nombreFichero, /* Abre fichero para lectura */ /* Se abre para lectura el fichero pFichero = fopen(nombreFichero, "r"); pFichero = fopen(nombreFichero, pFichero = fopen(nombreFichero, "rb"); /* Lee datos del fichero hasta que /* lee datos del fichero hasta que "rb"); llega al final Ilega al final */ /* Nos situamos al final del mismo while (fgets(linea, MAX LINEA, Tambien serviría un while /* Nos situamos al final del mismo (fscanf(...)!=EOF) */ pFichero)!=NULL) fseek(pFichero, OL, SEEK END); while (fscanf(pFichero, "%ld %s while(fread(&persona,sizeof(struct /* ftell devuelve el numero de DatosPersonales),1,pFichero) == 1) %s %f", &persona.dni, nRegistros++; bytes desde el principio del fichero persona.nombre, fclose(pFichero); hasta la posicion actual que es el /*Usar fseek y ftell contaria persona.apellido, final del fichero. */ &persona.salario)==NUM CAMPO tambien los marcados para borrar*/ return(nRegistros/NUM CAMPOS); numeroRegistros = S) /*solo contabilizamos si el ftell(pFichero)/sizeof(struct DatosPersonales); registro no esta marcado para nRegistros++; fclose(pFichero); borrar*/ if(persona.dni !=BORRADO) return numeroRegistros; fclose(pFichero); numeroRegistros++; return(nRegistros); fclose(pFichero); return numeroRegistros;

int buscarporDni(char *fichero, long dni_buscar, struct DatosPersonales *persona)

```
int buscarporDni(char *nombreFichero, long dni buscar, struct int buscarporDni(char* nombreFichero, long dni, struct
int buscarporDni(char *nombreFichero, long dni,
struct DatosPersonales *persona)
                                                      DatosPersonales *persona)
                                                                                                                       DatosPersonales *persona)
                                                                                                                       { FILE *pFichero;
                                                                                                                       int encontrado = 0; /* variables auxiliares para la busqueda
FILE *pFichero;
                                                      FILE *pFichero;
int encontrado = 0; /* variables auxiliares para la
                                                      int encontrado = 0;
busqueda */
                                                                                                                       struct DatosPersonales auxiliar;
                                                      int cont;
int cont;
                                                       char linea[MAX LINEA];
                                                                                                                       /* abre fichero para lectura */
struct DatosPersonales auxiliar;
                                                       struct DatosPersonales aux;
                                                                                                                       pFichero = fopen(nombreFichero, "rb");
/* abre fichero para lectura */
                                                                                                                       while (fread(&auxiliar, sizeof(struct DatosPersonales), 1,
pFichero = fopen(nombreFichero, "r");
                                                       /* abre fichero para lectura */
                                                                                                                       pFichero)==1)
                                                       pFichero = fopen(nombreFichero, "r");
                                                                                                                        { if (auxiliar.dni == dni)
/*evaluacion en cortocircuito*/
                                                                                                                         /* ha encontrado el registro */
while (!encontrado && (fscanf(pFichero, "%ld %s %s
                                                      //evaluación en corto circuito
                                                                                                                          encontrado = 1;
                                                       while (!encontrado && (fgets(linea, MAX LINEA,
    &auxiliar.dni, auxiliar.nombre, auxiliar.apellido,
                                                      pFichero)!=NULL))
                                                                                                                                            /* almacena en persona el registro
                      &auxiliar.salario) ==
                                                                                                                       encontrado */
NUM CAMPOS))
                                                        //Procesamos dni
                                                                                                                           *persona = auxiliar;
                                                        sscanf(linea, "%ld", &aux.dni);
  if (auxiliar.dni == dni) /* ha encontrado el registro
                                                        //Procesamos nombre
                                                                                                                       fclose(pFichero); /* se cierra el fichero */
                                                        fgets(linea, MAX LINEA, pFichero);
                                                                                                                       return encontrado;
                                                        limpiarLinea(linea);
   encontrado = 1;
                                                        strcpy(aux.nombre, linea);
   *persona = auxiliar; /* almacena en persona el
registro encontrado */
                                                        //Procesamos apellidos - se admiten espacios en blanco
                                                        fgets(linea, MAX LINEA, pFichero);
fclose(pFichero); /* se cierra el fichero */
                                                        limpiarLinea(linea);
return encontrado;
                                                        strcpy(aux.apellido, linea);
                                                        //Procesamos el salario
                                                        fgets(linea, MAX LINEA, pFichero);
                                                        sscanf(linea, "%f", &aux.salario);
                                                                           //almacena en persona el registro
                                                      encontrado
                                                        if (aux.dni == dni buscar) /* ha encontrado el registro */
                                                         encontrado = 1:
                                                          *persona = aux;
                                                       fclose(pFichero); /* se cierra el fichero */
                                                       return encontrado;
```

```
int mostrarporNombre(char *fichero, char *auxNombre)
int mostrarporNombre(char
                                             int mostrarporNombre(char
*nombreFichero, char *auxNombre)
                                             *nombreFichero, char *auxNombre)
FILE *pFichero;
                                             FILE *pFichero;
struct DatosPersonales auxiliar;
                                             struct DatosPersonales persona;
int encontrado = 0;
                                             int encontrado = 0;
/* abre fichero para lectura */
                                             char linea[MAX_LINEA];
pFichero = fopen(nombreFichero, "r");
                                             /* abre fichero para lectura */
                                             pFichero = fopen(nombreFichero, "r");
/*Evaluacion en cortocircuito
/ Tambien serviría un while
                                             while (fgets(linea, MAX LINEA,
(fscanf(...)!=EOF)*/
                                             pFichero)!=NULL)
while (!encontrado && (fscanf(pFichero,
"%ld %s %s %f",
                                               //Procesamos dni
    &auxiliar.dni, auxiliar.nombre,
                                               sscanf(linea, "%ld", &persona.dni);
auxiliar.apellido,
                      &auxiliar.salario) ==
                                               //Procesamos nombre
NUM CAMPOS))
                                               fgets(linea, MAX LINEA, pFichero);
                                               limpiarLinea(linea);
                                               strcpy(persona.nombre, linea);
 if (strcmp(auxiliar.nombre, auxNombre) ==
                                               //Procesamos apellidos - se admiten
 /* se ha encontrado un registro con ese
                                             espacios en blanco
nombre */
                                               fgets(linea, MAX LINEA, pFichero);
                                               limpiarLinea(linea);
  escribirDatosPersonales(auxiliar); /* se
                                               strcpy(persona.apellido, linea);
escriben sus datos */
  encontrado = 1;
                                               //Procesamos el salario
                                               fgets(linea, MAX LINEA, pFichero);
                                               sscanf(linea, "%f", &persona.salario);
fclose(pFichero); /* se cierra el fichero */
return encontrado;
                                              //se ha encontrado un registro con ese
                                             nombre
                                              if (strcmp(persona.nombre, auxNombre)
                                             == 0)
                                                 escribirDatosPersonales(persona); /* se
                                             escriben sus datos */
                                             encontrado = 1;
                                             fclose(pFichero); /* se cierra el fichero */
                                             return encontrado;
```

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

```
struct DatosPersonales registro i(char *fichero, int i) NOTA: i comienza a contar desde 1
struct DatosPersonales registro i(char struct DatosPersonales registro i(char
                                                                               struct DatosPersonales
                                                                                                                       struct DatosPersonales registro i(char
                                        *nombreFichero, long i)
*nombreFichero, long i)
                                                                               registro i(char* nombreFichero, long
                                                                                                                       *nombreFichero, long i)
                                        { FILE *pFichero;
                                        long j;
FILE *pFichero;
                                                                                                                        FILE *pFichero;
                                        struct DatosPersonales persona;
                                                                                FILE *pFichero;
long j;
                                                                                                                        long cont=0;
                                        char linea[MAX LINEA];
                                                                                struct DatosPersonales aux;
struct DatosPersonales persona;
                                                                                                                        struct DatosPersonales persona;
                                        pFichero = fopen(nombreFichero, "r");
                                                                                /* Se abre para lectura el fichero */
                                                                                                                       /* Se abre para lectura el fichero */
pFichero = fopen(nombreFichero,
"r");
                                                                                pFichero = fopen(nombreFichero,
                                                                                                                        pFichero = fopen(nombreFichero,
                                        for(j=0; j<i-1; j++)
                                                                                "rb");
                                                                                                                       "rb");
for(j=0; j<i; j++)
                                          fgets(linea, MAX LINEA, pFichero);
                                                                                /* Nos desplazamos al final del
                                                                                                                        while(cont<i)
                                          fgets(linea, MAX LINEA, pFichero);
  fscanf(pFichero, "%ld %s %s %f",
                                                                               registro i-1, desde el principio
                                          fgets(linea, MAX LINEA, pFichero);
&persona.dni, persona.nombre,
                                                                                del fichero con la funcion fseek*/
                                                                                                                         /*Usar fseek y ftell contaria tambien
                                          fgets(linea, MAX LINEA, pFichero);
                                                                                fseek(pFichero, (i-1)*sizeof(struct
                                                                                                                       los marcados para borrar*/
      persona.apellido,
                                                                               DatosPersonales), SEEK_SET);
&persona.salario);
                                                                                                                         fread(&persona,sizeof(struct
                                        //Procesamos dni
                                                                                                                       DatosPersonales),1,pFichero);
                                        fgets(linea, MAX LINEA, pFichero);
                                                                                /* Se lee el siguiente registro */
fclose(pFichero);
                                                                                                                         if(persona.dni !=BORRADO)
                                        sscanf(linea, "%ld", &persona.dni);
                                                                                fread(&aux,sizeof(struct
                                                                                                                           cont++;
/* devuelve el registro leido */
                                                                               DatosPersonales), 1, pFichero);
                                        //Procesamos nombre
return persona;
                                        fgets(linea, MAX LINEA, pFichero);
                                        limpiarLinea(linea);
                                                                                fclose(pFichero);
                                                                                                                        fclose(pFichero):
                                        strcpy(persona.nombre, linea);
                                                                                /* devuelve el registro leido */
                                                                                                                       /* devuelve el registro leido */
                                                                                return aux;
                                                                                                                        return persona;
                                        //Procesamos apellidos - se admiten
                                        espacios en blanco
                                        fgets(linea, MAX LINEA, pFichero);
                                        limpiarLinea(linea);
                                        strcpy(persona.apellido, linea);
                                        //Procesamos salario
                                        fgets(linea, MAX LINEA, pFichero);
                                        sscanf(linea, "%f", &persona.salario);
                                        fclose(pFichero);
                                        /* devuelve el registro leido */
                                        return persona;
```

```
struct DatosPersonales* ficheroAVector(char* fichero, long* nEle)
struct DatosPersonales*
                                         struct DatosPersonales*
                                                                                   struct DatosPersonales*
                                                                                                                        struct DatosPersonales*
ficheroAVector(char* nombreFichero,
                                         ficheroAVector(char* nombreFichero,
                                                                                   ficheroAVector(char*
                                                                                                                        ficheroAVector(char* nombreFichero, long*
                                                                                   nombreFichero, long* nEle)
long* nEle)
                                         long* nEle)
                                                                                                                        nEle)
 FILE *pFichero:
                                           FILE *pFichero:
                                                                                    struct DatosPersonales* V:
                                                                                                                          struct DatosPersonales* V:
 struct DatosPersonales persona, *V;
                                           struct DatosPersonales persona, *V;
                                                                                    FILE* pFichero;
                                                                                                                          FILE* pFichero;
                                           long i=0:
 long i=0;
                                                                                                                          long registros, i;
                                           char linea[MAX LINEA];
                                                                                     *nEle =
 * nEle =
                                                                                   contarRegistros(nombreFichero);
                                                                                                                          //Contamos registros, incluidos los borrados
                                                                                                                          //contarRegistros solo cuenta los activos,
contarRegistros(nombreFichero);
                                           *nEle =
                                                                                    V = reservarVector(*nEle);
 V = reservarVector(*nEle):
                                         contarRegistros(nombreFichero):
                                                                                    pFichero = fopen(nombreFichero.
                                                                                                                        por lo que no es valida
 pFichero = fopen(nombreFichero, "r");
                                           V = reservarVector(*nEle):
                                                                                   "rb");
                                                                                                                          pFichero = fopen(nombreFichero, "rb");
                                           pFichero = fopen(nombreFichero, "r");
                                                                                                                          fseek(pFichero, 0, SEEK END);
 /*Tambien serviría un while
                                                                                    /*Leemos todo el fichero (mas
                                                                                                                          registros = ftell(pFichero)/sizeof(struct
(fscanf(...)!=EOF)*/
                                           /*Leemos todo el fichero*/
                                                                                   rapido que elemento a elemento)*/
                                                                                                                        DatosPersonales);
 while (fscanf(pFichero, "%ld %s %s %f",
                                           while (fgets(linea, MAX LINEA,
                                                                                    fread(V. sizeof(struct
&persona.dni, persona.nombre,
                                         pFichero)!=NULL)
                                                                                   DatosPersonales),*nEle,pFichero);
                                                                                                                          V = reservarVector(registros);
                                                                                    fclose(pFichero);
persona.apellido,
                                           //Procesamos dni
                                                                                                                         /*Leemos todo el fichero (mas rapido que
                                                                                    return(V);
&persona.salario)==NUM CAMPOS)
                                           sscanf(linea, "%ld", &persona.dni);
                                                                                                                        elemento a elemento)*/
                                                                                                                          fseek(pFichero, 0, SEEK SET);
  V[i]=persona;
                                           //Procesamos nombre
                                                                                                                          fread(V, sizeof(struct DatosPersonales),
                                           fgets(linea, MAX LINEA, pFichero);
                                                                                                                        registros, pFichero);
  i++;
                                           limpiarLinea(linea);
                                                                                                                         fclose(pFichero);
                                           strcpy(persona.nombre, linea);
                                                                                                                         //Nos quedamos solamente con los
 fclose(pFichero);
 return(V);
                                           //Procesamos apellidos - se admiten
                                                                                                                        registros no borrados
                                         espacios en blanco
                                                                                                                          for(i=0, *nEle=0; i<registros; i++)
                                           fgets(linea, MAX LINEA, pFichero);
                                           limpiarLinea(linea);
                                                                                                                           printf("\ncopiando registro: %ld nEle:
                                                                                                                        %ld", i, *nEle);
                                           strcpy(persona.apellido, linea);
                                                                                                                                          if(V[i].dni!=BORRADO)
                                           //Procesamos salario
                                           fgets(linea, MAX LINEA, pFichero);
                                                                                                                                                           V[*nEle] =
                                           sscanf(linea, "%f", &persona.salario);
                                                                                                                        V[i];
                                                                                                                                            (*nEle)++; //Parentesis
                                           V[i]=persona;
                                                                                                                        necesarios
                                           i++;
                                           fclose(pFichero);
                                                                                                                          V= (struct DatosPersonales*) realloc(V,
                                                                                                                        (*nEle)*sizeof(struct DatosPersonales));
                                           return(V);
                                                                                                                          return(V);
```

Funciones de escritura

void vectorAFichero(char* void vectorAFichero(char* void vectorAFichero(char* nombreFichero, struct nombreFichero, struct DatosPersonales* nombreFichero, struct DatosPersonales* DatosPersonales* V, long nEle) V, long nEle) V, long nEle) { FILE* pFichero; pFichero = fopen(nombreFichero, "wb"); FILE* pFichero; fwrite(V, sizeof(struct DatosPersonales), nEle, pFichero); FILE* pFichero; long i=0; long i=0; fclose(pFichero); pFichero = fopen(nombreFichero, pFichero = fopen(nombreFichero, "wt"); "w"); for (i=0; i<nEle; i++) for (i=0; i<nEle; i++) fprintf(pFichero, fprintf(pFichero, "%ld %s %s %.3f\n", "%ld\n%s\n%s\n%.3f\n", V[i].dni, V[i].dni, V[i].nombre, V[i].apellido, V[i].nombre, V[i].apellido, V[i].salario); V[i].salario); fclose(pFichero); fclose(pFichero);

void vectorAFichero(char* fichero, struct DatosPersonales* V, long nEle)

void anadirRegistro(char *fichero, struct DatosPersonales persona)

```
void anadirRegistro(char
*nombreFichero, struct
DatosPersonales persona)
FILE *pFichero;
//abre el fichero para anadir
pFichero =
fopen(nombreFichero, "a");
/* guarda los datos en el
fichero */
fprintf(pFichero, "%ld %s %s
%.3f\n", persona.dni,
persona.nombre,
     persona.apellido,
persona.salario);
fclose(pFichero);
```

```
void anadirRegistro(char*
nombreFichero, struct
DatosPersonales persona)
FILE *pFichero;
// abre el fichero para anadir
pFichero =
fopen(nombreFichero, "a");
/* escribe los datos en el
fichero */
fprintf(pFichero,
"%Id\n%s\n%s\n%.3f\n",
persona.dni, persona.nombre,
     persona.apellido,
persona.salario);
fclose(pFichero);
```

```
void anadirRegistro(char* nombreFichero, struct
DatosPersonales persona)
{
  FILE *pFichero;

  pFichero = fopen(nombreFichero, "ab"); /* abre
el fichero para anadir */
  /* guarda los datos en el fichero */
fwrite(&persona, sizeof(struct DatosPersonales),
1, pFichero);
fclose(pFichero);
}
```

```
int actualizarporDni(char* fichero, long dni)
int actualizarporDni(char* nombreFichero, long dni)
                                                                  int actualizarporDni(char* nombreFichero, long dni_buscar)
                                                                                                                                                   int actualizarporDni(char* nombreFichero, long dni)
FILE *pFichero1, *pFichero2;
                                                                  FILE *pFichero1, *pFichero2;
                                                                                                                                                   FILE *pFichero;
struct DatosPersonales aux:
                                                                  struct DatosPersonales persona;
                                                                                                                                                   struct DatosPersonales aux:
int encontrado = 0;
                                                                  char linea[MAX LINEA];
                                                                                                                                                   int encontrado =0;
                                                                  int encontrado = 0;
                                                                                                                                                   /* Se abre para lectura y escritura */
/* Se abre para lectura el fichero original */
                                                                                                                                                   pFichero = fopen(nombreFichero, "r+b");
pFichero1 = fopen(nombreFichero, "r");
                                                                  /* Se abre para lectura el fichero original */
                                                                  pFichero1 = fopen(nombreFichero, "r");
                                                                                                                                                   /* Se recorre el fichero */
/* Fichero temporal para volcar los registros que no se borran */
                                                                                                                                                   while(fread(&aux, sizeof(struct DatosPersonales),1,pFichero)
pFichero2 = fopen("temporal", "w");
                                                                  /* Fichero temporal para volcar los registros que no se borran */
                                                                                                                                                   == 1)
                                                                  pFichero2 = fopen("temporal", "w");
/* Recorre el fichero original y los registros que no hay que borrar
                                                                                                                                                    /* Se comprueba si el registro tiene ese dni*/
 se pasan al fichero temporal.
                                                                  /* Se recorre el fichero original y los registros que no hay que borrar
                                                                                                                                                    if (aux.dni == dni)
 Tambien serviría un while (fscanf(...)!=EOF)*/
                                                                    se pasan al fichero temporal */
                                                                                                                                                                         aux = introducirDatosPersonales():
while(fscanf(pFichero1, "%ld %s %s %f", &aux.dni, aux.nombre,
                                                                                                                                                                         encontrado = 1;
   aux.apellido, &aux.salario) == NUM_CAMPOS)
                                                                  while ((fgets(linea, MAX_LINEA, pFichero1)!=NULL))
                                                                                                                                                     /* Nos situamos al principio del registro encontrado */
 if (aux.dni == dni)
                                                                    //Procesamos el dni
                                                                                                                                                     fseek(pFichero, -(int)sizeof(struct DatosPersonales),
                                                                    sscanf(linea, "%ld", &persona.dni);
                                                                                                                                                  SEEK CUR);
                                                                                                                                                     /* Se reescribe el registro */
                      aux = introducirDatosPersonales();
                      encontrado = 1;
                                                                    //Procesamos el nombre
                                                                                                                                                     fwrite(&aux, sizeof(struct DatosPersonales), 1, pFichero);
                                                                    fgets(linea, MAX_LINEA, pFichero1);
                                                                                                                                                     //La siguiente orden es necesario para que en windows
 fprintf(pFichero2, "%ld %s %s %.3f\n", aux.dni, aux.nombre,
                                                                                                                                                     //se actualicen los flags de FILE*
                                                                    limpiarLinea(linea);
                                                                                                                                                     //En Linux, no es necesario, pero se puede dejar
     aux.apellido, aux.salario);
                                                                    strcpy(persona.nombre, linea);
                                                                                                                                                     //El estándar requiere fflush para hacer fread despues de
                                                                    //Leemos apellidos - se admiten espacios en blanco
                                                                                                                                                   fwrite
/* Se cierran los ficheros */
                                                                    fgets(linea, MAX_LINEA, pFichero1);
                                                                                                                                                     fflush(pFichero);
fclose(pFichero1);
                                                                    limpiarLinea(linea);;
fclose(pFichero2);
                                                                    strcpy(persona.apellido, linea);
                                                                                                                                                   /* Se cierra el fichero*/
/* Se borra el fichero original */
                                                                                         //Leemos el salario
                                                                                                                                                   fclose(pFichero);
                                                                    fgets(linea, MAX LINEA, pFichero1);
remove(nombreFichero);
                                                                                                                                                   return(encontrado)
                                                                    sscanf(linea, "%f", &persona.salario);
/* Se renombra el temporal con el nombre del original */
rename("temporal", nombreFichero);
                                                                    if (persona.dni == dni buscar)
return(encontrado);
                                                                                         persona = introducirDatosPersonales();
                                                                                         encontrado = 1;
                                                                    fprintf(pFichero2, "%ld\n%s\n%s\n%.3f\n", persona.dni, persona.nombre,
                                                                        persona.apellido, persona.salario);
                                                                   /* Se cierran los ficheros */
                                                                   fclose(pFichero1);
                                                                   fclose(pFichero2);
                                                                  /* Se borra el fichero original */
                                                                  remove(nombreFichero);
                                                                  /* Se renombra el temporal con el nombre del original */
                                                                  rename("temporal", nombreFichero);
                                                                  return(encontrado);
```

int borrarporDni(char *fichero, long dni)					
int borrarporDni(char *nombreFichero, long dni)	int borrarporDni(char *nombreFichero, long dni_buscar)		int borrarporDni(char *nombreFichero, long 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 borrado=0;	int encontrado = 0;		
/* Se abre para lectura el fichero original */		/* Se abre para lectura el fichero original */	/* Se abre para lectura y escritura el fichero */		
pFichero1 = fopen(nombreFichero, "r");	<pre>/* Se abre para lectura el fichero original */ pFichero1 = fopen(nombreFichero, "r");</pre>	<pre>pFichero1 = fopen(nombreFichero, "rb");</pre>	pFichero = fopen(nombreFichero, "r+b");		
/* Fichero temporal para volcar los registros que no se	/m == 1	/* Fichero temporal para volcar los registros que no se	/* Se recorre el fichero original y los registros a borrar se		
borran */	/* Fichero temporal para volcar los registros que no se borran */	borran */	reescriben marcandolos como borrados */		
pFichero2 = fopen("temporal", "w");	pFichero2 = fopen("temporal", "w");	pFichero2 = fopen("temporal", "wb");	while((fread(&aux, sizeof(struct DatosPersonales),1,pFichero) == 1)&&!encontrado)		
/* Recorre el fichero original y los registros que no hay		/* Se recorre el fichero original y los registros que no hay			
que borrar	/* Se recorre el fichero original y los registros que no hay	l '	/* Se comprueba si el registro tiene ese nombre y no		
se pasan al fichero temporal	que borrar	se pasan al fichero temporal */	está marcado */		
Tambien serviría un while (fscanf()!=EOF)*/	se pasan al fichero temporal */	131 /6 1/0 : //	if (aux.dni == dni)		
while(fscanf(pFichero1, "%ld %s %s %f", &aux.dni,	while (facts/lines MAY LINEA prichare1) -NIIII)	while(fread(&aux, sizeof(struct	/* Co marca como harrado */		
aux.nombre, aux.apellido, &aux.salario) == NUM CAMPOS)	while (fgets(linea, MAX_LINEA, pFichero1)!=NULL)	DatosPersonales),1,pFichero1) == 1) {	/* Se marca como borrado */ aux.dni = BORRADO;		
aux.apelliuo, Qaux.saiario) == Noivi_CAiviPos)	//Procesamos dni	/* Se comprueba si el registro tiene ese dni*/	/* Nos situamos al principio del registro encontrado */		
if (aux.dni != dni)	sscanf(linea, "%ld", &persona.dni);	if (dni buscar!=aux.dni)	fseek(pFichero, -(int)sizeof(struct DatosPersonales),		
fprintf(pFichero2, "%ld %s %s %.3f\n", aux.dni,	33cam(mica, 701a , epersona.um,	fwrite(&aux, sizeof(struct DatosPersonales), 1,	SEEK CUR);		
aux.nombre, aux.apellido, aux.salario);	//Procesamos nombre	pFichero2);	/* Se reescribe el registro */		
else	fgets(linea, MAX_LINEA, pFichero1);	else	fwrite(&aux, sizeof(struct DatosPersonales), 1,		
borrado = 1;	limpiarLinea(linea);	borrado = 1;	pFichero);		
}	strcpy(persona.nombre, linea);	}	//La siguiente orden es necesario para que en windows		
		/* Se cierran los ficheros */	//se actualicen los flags de FILE*		
/* Se cierran los ficheros */	//Procesamos apellidos - se admiten espacios en blanco		//En Linux, no es necesario, pero se puede dejar		
fclose(pFichero1);	fgets(linea, MAX_LINEA, pFichero1);	fclose(pFichero2);	//El estándar requiere fflush para hacer fread despues		
fclose(pFichero2);	limpiarLinea(linea);	/* C	de fwrite		
/* Se borra el fichero original */	strcpy(persona.apellido, linea);	/* Se borra el fichero original */ remove(nombreFichero);	fflush(pFichero); /* Se ha borrado al menos un registro */		
remove(nombreFichero);	//Procesamos salario	remove(nombrerichero),	encontrado = 1;		
remove(nombremeno),	fgets(linea, MAX_LINEA, pFichero1);	/* Se renombra el temporal con el nombre del original */	}		
/* Se renombra el temporal con el nombre del original */	sscanf(linea, "%f", &persona.salario);	rename("temporal", nombreFichero);	}		
rename("temporal", nombreFichero);	, , , , , , , , , , , , , , , , , , , ,	return(borrado);	/* Se cierra el fichero */		
return(borrado);	//El registro no contiene	}	fclose(pFichero);		
}	dni_buscar		retur		
	if (persona.dni != dni_buscar)				
	fprintf(pFichero2, "%ld\n%s\n%s\n%.3f\n", persona.dni,				
	persona.nombre,				
	persona.apellido, persona.salario);				
	else				
	borrado = 1;				
	1				
	/* Se cierran los ficheros */				
	fclose(pFichero1);				
	fclose(pFichero2);				
	/* Se borra el fichero original */				
	remove(nombreFichero);				
	/* Se renombra el temporal con el nombre del original */				
	rename("temporal", nombreFichero);				
	return(borrado);				
	J .				

Incrementar salarios

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

Borrado físico B2

int borrarMarcados(char *fichero)

```
int borrarMarcados(char *nombreFichero)
FILE *pFichero1, *pFichero2;
struct DatosPersonales aux;
int borrados = 0;
 /* Se abre para lectura el fichero original */
pFichero1 = fopen(nombreFichero, "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);
 else
 /* al menos hay uno marcado */
 borrados++;
/* Se cierran los ficheros */
fclose(pFichero1);
fclose(pFichero2);
/* Se borra el fichero original */
remove(nombreFichero);
/* Se renombra el temporal con el nombre del original */
rename("temporal", nombreFichero);
return borrados;
```