

# MICRO2.pdf



**David2907**



**Metodología de la Programación**



**1º Grado en Ingeniería Informática**



**Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba**

//Nos piden unificar dos ficheros binarios llamados productosA.bin y productos.bin en uno de fichero de texto (el nombre de fichero de texto a crear, lo introduces tú por lista de comandos).

## FUNCIONES.H

```
#ifndef __MICRO__
#define __MICRO__

struct producto{
    char nombre[50];
    int cod;
    float precio;
    int unidades;
};

struct producto*ficheroaVector(char*nombre,struct producto*v,long nele);

struct producto*reservarMemoria(long nele1);

long contarregistros(char*nombre);

int comprueba_ficheros(char*f1,char*f2);

void unificarFichero(char*nombre1, char*nombre2, char*nombre3, struct producto*v1,struct
producto*v2);

#endif
```

## FUNCIONES.C

```
#include "micro.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
int comprueba_ficheros(char*f1,char*f2){
FILE* fichero1;
FILE* fichero2;

    if ((fichero1=fopen(f1,"rb"))==NULL)
    {
        printf("\nNo existe el fichero1: %s",f1);
        exit(-1);
    }
    if ((fichero2=fopen(f2,"rb"))==NULL)
    {
        printf("\nNo existe el fichero2: %s",f2);
        exit(-1);
    }
}
```

```
long contarregistros(char*nombre){
```

```
FILE* f;
long nele;
```

```
if ((f=fopen(nombre,"rb"))==NULL)
{
    printf("\nNo existe el fichero1: %s",nombre);
    exit(-1);
}
```

```
struct producto productos;
fseek(f,0L,SEEK_END);
nele=ftell(f)/sizeof(struct producto);
return(nele);
fclose(f);
```

```
}
```

```
void    unificarFichero(char*nombre1,char*nombre2,char*nombre3,struct    producto*v1,struct
producto*v2){
```

```
struct producto productos;
FILE* f1;
FILE* f2;
FILE* f3;
```

```

if ((f1=fopen(nombre1,"rb"))==NULL)
{
printf("\nError no se pudo abrir fichero1");
exit(-1);
}

if ((f2=fopen(nombre2,"rb"))==NULL)
{
printf("\nError no se pudo abrir fichero1");
exit(-1);
}

if ((f3=fopen(nombre3,"w"))==NULL)
{
printf("\nError no se pudo abrir fichero1");
exit(-1);
}

for (int i = 0; i < contarregistros(nombre1); ++i)
{
fprintf(f3, "%s\n%d\n%f\n%d\n", v1[i].nombre, v1[i].cod, v1[i].precio, v1[i].unidades );
}

for (int i = 0; i < contarregistros(nombre2); ++i)
{
fprintf(f3, "%s\n%d\n%f\n%d\n", v2[i].nombre, v2[i].cod, v2[i].precio, v2[i].unidades );
}

fclose(f1);
fclose(f2);
fclose(f3);
}

struct producto*reservarMemoria(long nele1){
    struct producto*v;
    v=(struct producto*)malloc(nele1*sizeof(struct producto));
if (v==NULL)
    {
        printf("\nNo se pudo reservar vector");
        exit(-1);
    }
    return(v);
}

```

```

struct producto*ficheroaVector(char*nombre,struct producto*v,long nele){
    FILE* f1;
    if ((f1=fopen(nombre,"rb"))==NULL)
    {
        printf("\nError al abrir fichero");
        exit(-1);
    }
    fread(v,sizeof(struct producto),nele,f1);
    fclose(f1);
    return(v);
}

```

## MAIN.C

```

#include "micro.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[])

{
    long nele1,nele2;
    struct producto productos;
    struct producto*v1;
    struct producto*v2;
    if (argc!=4)
    {
        printf("\n Error, comprueba que has introducido correctamente los argumentos: ");
        printf("\n-> Ejecutable.exe productosA.bin productosB.bin fichero.txt");
        exit(-1);
    }

    nele1=contarregistros(argv[1]);
    nele2=contarregistros(argv[2]);

    v1=reservarMemoria(nele1);
    v2=reservarMemoria(nele2);

    comprueba_ficheros(argv[1],argv[2]);

    v1=ficheroaVector(argv[1],v1,nele1);
    v2=ficheroaVector(argv[2],v2,nele2);

    unificarFichero(argv[1],argv[2],argv[3],v1,v2);

    free(v1);
    free(v2);

    v1=NULL;
    v2=NULL;
}

```

```
    return 0;  
}
```

## MAKEFILE

all: mensajeSaludo micro.exe clean

mensajeSaludo:

@echo EJECUTANDO EL MAKEFILE DE NOMBRE\_AUTOR

micro.exe: mainmicro.o micro.o

gcc -o micro.exe mainmicro.o micro.o

mainmicro.o: mainmicro.c micro.h

gcc -c mainmicro.c

micro.o: micro.c micro.h

gcc -c micro.c

.PHONY: clean

clean:

@echo Borrado listo

@echo rm \*.o