

```
/**
 * @file negativo.cpp
 * @brief Calcula el negativo de una imagen PGM
 *
 * Formato:
 *   negativo <FichImagenOriginal> <FichImagenDestino>
 *   donde:
 *   <FichImagenOriginal> es la imagen PGM a la que se va a calcular
 *   su negativo. No se modifica.
 *   <FichImagenDestino> es la imagen PGM resultado de calcular el negativo
 *   de <FichImagenOriginal>.
 */

#include <iostream>
#include <cstring>
#include <cstdlib>

#include "imagenES.h"

using namespace std;

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

    char *origen, *destino; // nombres de los ficheros

    unsigned char *image;
    int nf, nc, npixeles; // Num. de filas y columnas de las imagenes

    // Comprobar validez de la llamada
    if (argc != 3){
        cerr << "Error: Numero incorrecto de parametros.\n";
        cerr << "Uso: negativo <FichImagenOriginal> <FichImagenDestino>\n";
        exit (1);
    }

    origen = argv[1];
    destino = argv[2];

    cout << endl;
    cout << "Fichero origen: " << origen << endl;
    cout << "Fichero resultado: " << destino << endl;

    // Leer la imagen del fichero de entrada
    image = LeerImagenPGM (origen, nf, nc);
    if (!image){
        cerr << "Error: No pudo leerse la imagen." << endl;
        cerr << "Terminando la ejecucion del programa." << endl;
        exit (1);
    }

    // Mostrar los parametros calculados por LeerImagenPGM()
    cout << endl;
    cout << "Dimensiones de " << origen << ":" << endl;

    cout << "    Imagen    = " << nf << " filas x " << nc << " columnas " << endl;

    // Calcular el negativo
    npixeles = nf*nc;
    for (int i=0; i<npixeles; i++){
        image[i] = 255 - image[i];
    }

    // Guardar la imagen resultado en el fichero
    if (EscribirImagenPGM (destino, image, nf, nc))
        cout << "La imagen se guardo en " << destino << endl;
    else{
        cerr << "Error: No pudo guardarse la imagen." << endl;
        cerr << "Terminando la ejecucion del programa." << endl;
        exit (2);
    }

    // Liberar la imagen
    delete [] image;
```

```
    return (0);  
}
```