**PROYECTO PGM**

Juan Manuel Rodríguez Gómez

Doble Grado en Ingeniería Informática y Matemáticas

Fundamentos de la Programación

Curso 2019 – 2020

**1. Diseño General**

Este proyecto consiste en crear cuatro programas en el lenguaje de programación C++ para trabajar con imágenes. Para almacenar dichas imágenes utilizaremos ficheros de texto con extensión .pgm (PGM es el acrónimo de *Portable Gray Map file format*). Cada programa hace lo siguiente:

* **ProyectoPGM\_Blanquear\_JMRG.exe:** Convierte la imagen en otra completamente blanca.
* **ProyectoPGM\_Contraste\_JMRG.exe:** Contrasta al máximo la imagen.
* **ProyectoPGM\_Negativo\_JMRG.exe:** Muestra el negativo de la imagen.
* **ProyectoPGM\_Rotar\_JMRG.exe:** Rota la imagen 90º a la derecha.

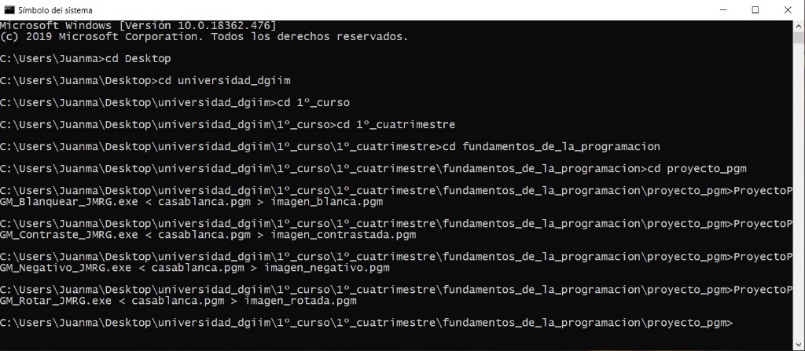
Cada programa funcionará mediante cuatro funciones:

1. Función principal (que contiene a las tres funciones siguientes).
2. Función de lectura de datos de la imagen.
3. Función que realiza las acciones necesarias sobre la imagen.
4. Función que devuelve los datos de la imagen resultante.

donde las funciones de lectura y devolución de datos serán iguales para los cuatro programas.

Para ejecutar uno de los programas anteriores y que este devuelva la imagen

resultante, tenemos que hacerlo desde la terminal de nuestro sistema operativo (en mi caso, Windows 10). Por otro lado, para visualizar la imagen, usaremos una aplicación gratuita para Windows llamada *Brennig’s*.



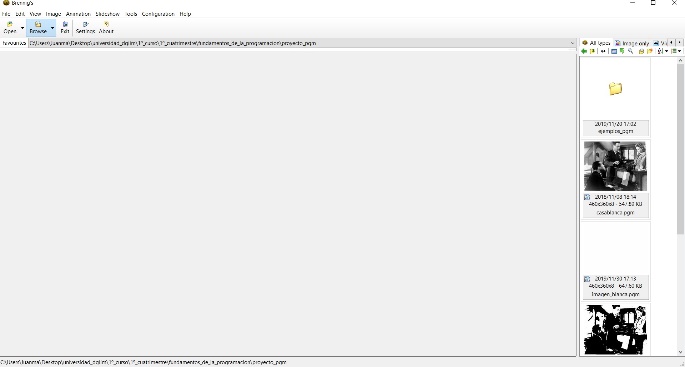
****

Imagen 1.2: *Brennig’s*.

Imagen 1.1: Programas ejecutados desde la terminal de Windows 10.

Vamos a usar como ejemplo la imagen *casablanca.pgm*:



Imagen 1.3: *casablanca.pgm*

Tras ejecutar los programas, las imágenes resultantes serían las siguientes (el resultado

Del programa 1 no se muestra ya que simplemente es una imagen en blanco):

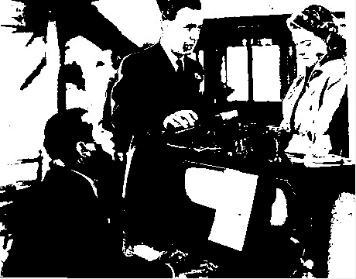


Imagen 1.5: *imagen\_negativo.pgm*



Imagen 1.4: *imagen\_contrastada.pgm*

Imagen 1.6: *imagen\_rotada.pgm*

**2. Programación**

A la hora de trabajar con imágenes, hace falta saber que una imagen es en realidad una matriz de *n* filas por *m* columnas, donde cada elemento de la matriz es un píxel.

Ahora pasamos a analizar las funciones de los programas:

1. **Función principal:**

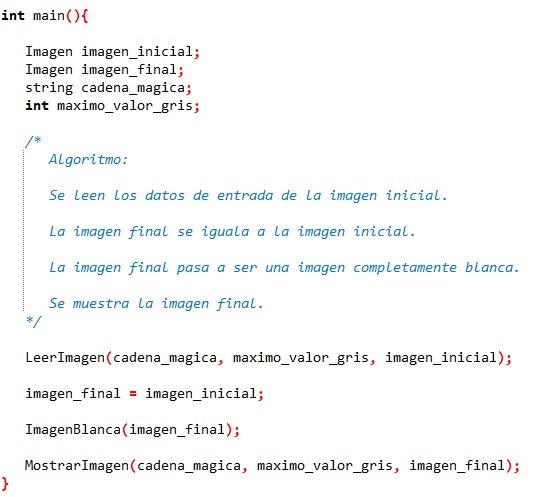


Imagen 2.1: Función principal del programa *ProyectoPGM\_Blanquear\_JMRG.exe*

Lo único que varía en la función principal de cada programa es la función que realiza las acciones necesarias sobre la imagen (siendo en este ejemplo *ImagenBlanca*).

1. **Funciones de lectura y devolución de datos:**

Recordemos que son iguales para los cuatro programas.

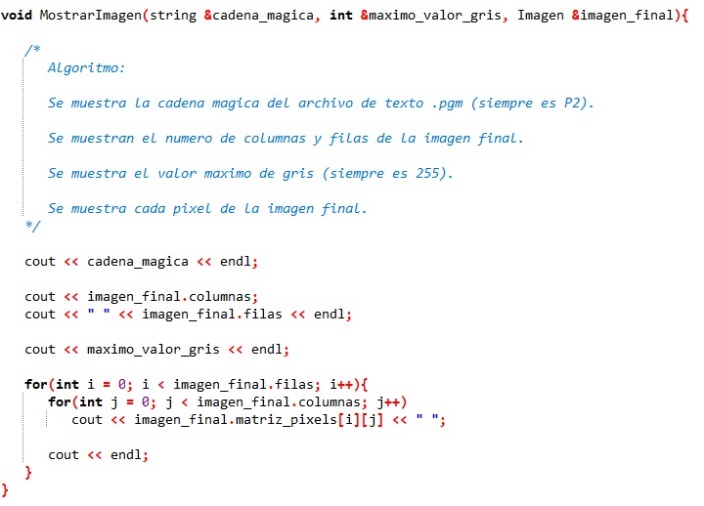
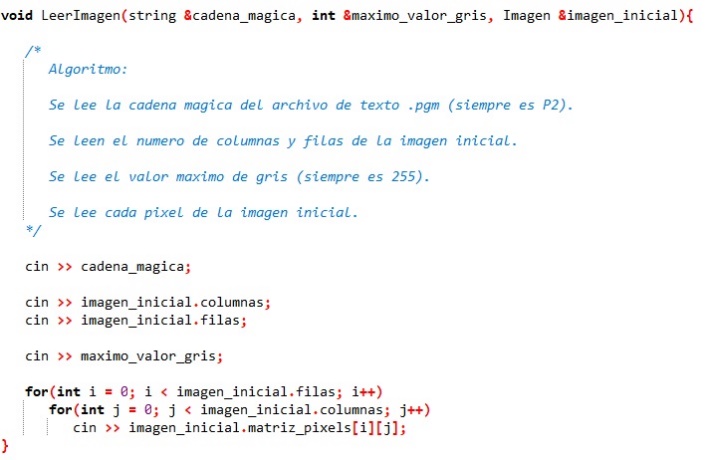
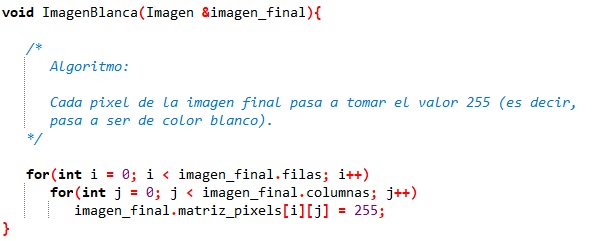


Imagen 2.3: Función de devolución de datos

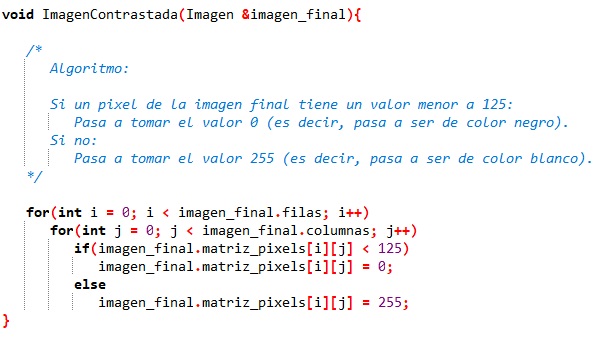
Imagen 2.2: Función de lectura de datos

1. **Funciones que realizan las acciones necesarias sobre la imagen:**

****

* **ProyectoPGM\_Blanquear\_JMRG.exe:** Para convertir la imagen en otra completamente blanca, el programa asigna a cada píxel el valor 255.

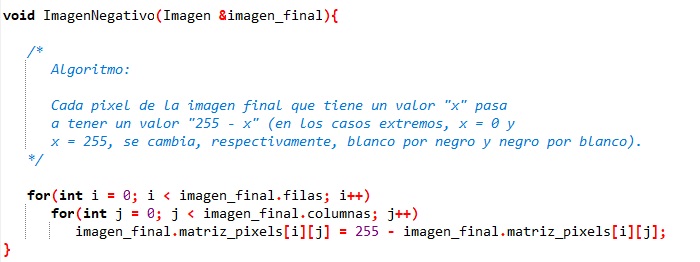
Imagen 2.4: Función *ImagenBlanca* en C++.

****

* **ProyectoPGM\_Contraste\_JMRG.exe:** Para contrastar al máximo la imagen, el programa cambia los valores de los píxeles menores de 125 a 0 y los mayores o iguales los cambia a 255.

Imagen 2.5: Función *ImagenContrastada* en C++.

* **ProyectoPGM\_Negativo\_JMRG.exe:** Para mostrar el negativo de la imagen, el programa cambia un píxel con el valor *x* a *255 – x* (en los casos extremos, *x = 0* y *x = 255*, el programa cambia, respectivamente, negro por blanco y blanco por negro).

****

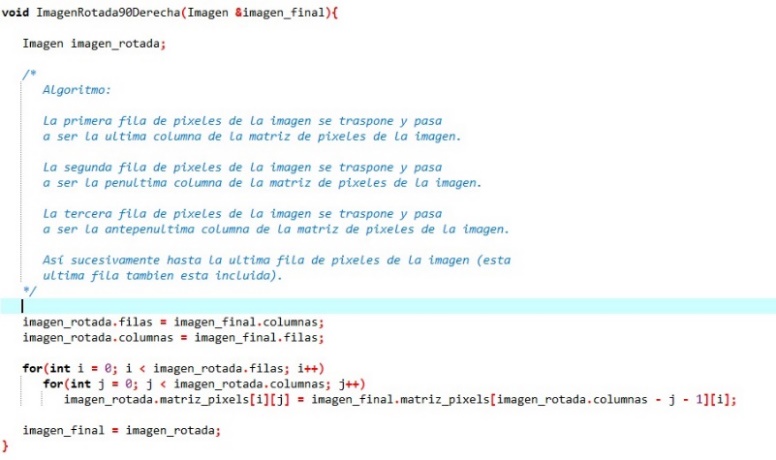
****

Imagen 2.7: Función *ImagenRotada90Derecha* en C++.

* **ProyectoPGM\_Rotar\_JMRG.exe:** Para rotar la imagen 90º a la derecha, el programa traspone la primera fila de la imagen y la coloca en la última columna, luego traspone la segunda fila y la coloca en la penúltima columna y así sucesivamente hasta completar todas las filas de la imagen.

Imagen 2.6: Función *ImagenNegativo* en C++.