

Fundamentos de Programación y Laboratorio

Práctica No. 1.

Allan Jair Escamilla Hernández

Este documento presenta el desarrollo llevado a cabo para crear un programa que permite la manipulación de imágenes PGM. El programa será capaz de ecualizar, ampliar, reducir y obtener el negativo de una imagen dada por el usuario.

I. INTRODUCCIÓN

Este trabajo presenta un programa de computadora que le permite a un usuario manipular una imagen en formato PGM. El programa le preguntará al usuario el nombre de la imagen que desea manipular y posteriormente le desplegará un menú de opciones para que el usuario elija lo que quiere hacer con la imagen.

Las opciones que se le mostrarán al usuario en el menú son las siguientes: Obtener negativo de la imagen, Ecualizar imagen, Ampliar imagen y Reducir imagen. Una vez que el usuario haya seleccionado la opción que desea aplicarle a la imagen, se le pedirá que ingrese un nuevo nombre para almacenar la imagen modificada.

II. ANÁLISIS

Se solicita un programa que procese digitalmente imágenes en formato PGM. El programa deberá permitir al usuario ecualizar, ampliar, reducir y obtener el negativo de la imagen seleccionada por el usuario. Al principio del programa se deberá de desplegar el nombre de la persona que elaboró el programa, posteriormente se le deberá de preguntar al usuario el nombre de la imagen a modificar. Después, al usuario se le desplegará un menú de opciones para que elija lo que le desea hacer a la imagen. Una vez modificada la imagen, se le pedirá al usuario que indique el nombre del archivo nuevo en el que se almacenará la imagen procesada.

A. Entradas

Nombre Archivo. Será el nombre del archivo a procesar.

Alto, Ancho. Serán las dimensiones del archivo a procesar.

Matriz. Será la matriz donde se almacenará la imagen a procesar.

Nombre Nuevo. Será el nuevo nombre para el archivo de imagen una vez que haya sido procesado.

B. Procesos

Desplegar Datos. Desplegará los datos del desarrollador.

Pedir Archivo. Pedirá el nombre del archivo de imagen a procesar.

Validar Archivo. Validará la existencia del archivo de imagen introducida por el usuario.

Obtener Dimensiones. Leerá del archivo las dimensiones de la imagen a procesar.

Cargar Archivo. Cargará a una matriz la imagen a procesar.

Negativo. Obtendrá el negativo de la imagen introducida por el usuario.

Ecualizar. Ecualizará la imagen introducida por el usuario.

Reducción. Reducirá a la mitad las dimensiones de la imagen introducida por el usuario.

Ampliación. Duplicará las dimensiones de la imagen ingresada por el usuario.

Nuevo Nombre. Pedirá al usuario un nuevo nombre para almacenar la imagen una vez haya sido procesada.

Guardar. Guarda la imagen procesada.

C. Salidas

Nombre Archivo. Nombre del archivo a procesar.

Validación. Dato que servirá para validar la existencia del archivo a procesar.

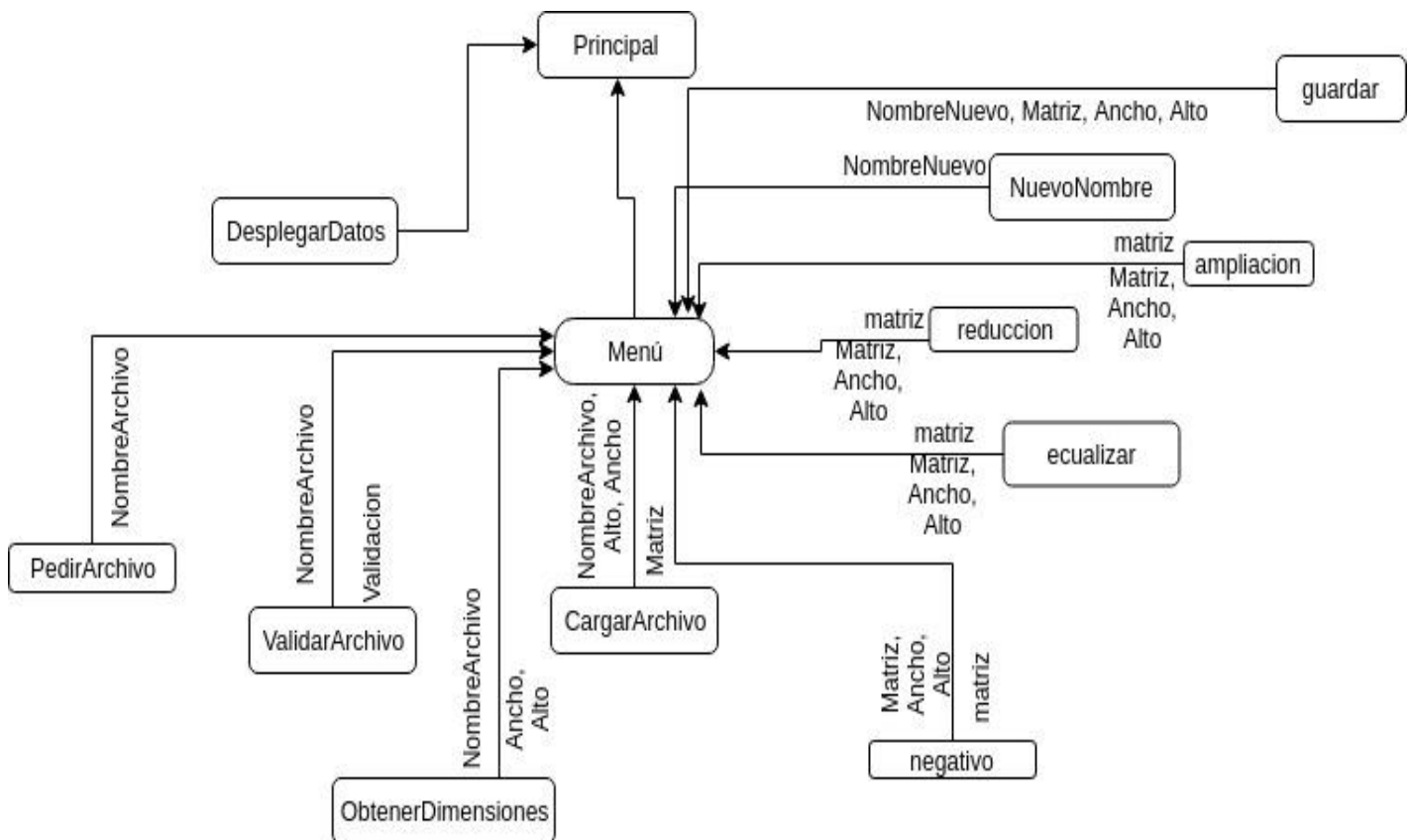
Ancho, Alto. Dimensiones del archivo a procesar.

Matriz. Matriz donde se guardará la imagen a procesar.

Nombre Nuevo. Nombre del archivo una vez procesado.

III. DISEÑO

A. Diagrama IPO



B. Pseudocódigo

```

Principal( | ){
    DesplegarDatos( | );
    Menu( | );
}

DesplegarDatos( | ){
    imprimir("Allan Jair Escamilla Hernández");
    imprimir("Ingresar letra para continuar... ");
    leer(letra);
    imprimir("Objetivo: Procesar digitalmente una imagen PGM dada por el usuario");
}

Menu( | ){
    opcion = 0;
    PedirArchivo( | NombreArchivo);
  
```

```
ValidarArchivo( NombreArchivo | Validacion);
Si(validacion == 0){
    ObtenerDimensiones( NombreArchivo | Ancho, Alto);
    cargarArchivo( NombreArchivo, Ancho, Alto | Matriz);
    imprimir("1.- Negativo de la imagen.");
    imprimir("2.- Ecualizar imagen.");
    imprimir("3.- Reducir imagen.");
    imprimir("4.- Ampliar imagen.");
    imprimir("Elije una opcion: ");
    leer(opcion);
    Seleccion(opcion) {
        caso 1:
            negativo( matriz, Ancho, Alto | matriz );
        caso 2:
            ecualizar( matriz, Ancho, Alto | matriz );
        caso 3:
            reduccion( matriz, Ancho, Alto | matriz2 );
        caso 4:
            ampliacion( matriz, Ancho, Alto | matriz2 );
        caso por defecto:
            imprimir(No se ha encontrado la opcion);
    }
    NuevoNombre( | NombreNuevo );
    guardar( NombreNuevo, Matriz, Altura, Ancho | );
}Sino{
    imprimir("No se ha encontrado el archivo:");
}
}

PedirArchivo( | NombreArchivo){
    imprimir("Ingresar nombre del archivo: ");
    leer(NombreArchivo);
}
ValidarArchivo( | Validacion){
    Abrir(NombreArchivo);
    Si(NombreArchivo es Nulo){
        Validacion = 1;
    }Sino{
        Validacion = 0;
    }
    Cerrar(NombreArchivo);
}
cargarArchivo( NombreArchivo, Ancho, Alto | Matriz){
    Abrir(NombreArchivo);
    Desde i = 0 hasta i = Alto i = i + 1{
        leerdeArchivo(Matriz[i], tamañode(entero), Ancho, NombreArchivo);
    }
    Cerrar(NombreArchivo);
}
ObtenerDimensiones( NombreArchivo | Ancho, Alto){
    Abrir(NombreArchivo);
    Desde i = 0 hasta i = 3 i = i +1{
        leer(Datos[i], tamañode(entero), 1, NombreArchivo);
    }
    Ancho = Datos[1];
    Alto = Datos[2];
}
```

```
Cerrar(NombreArchivo);
}

negativo( matriz, Ancho, Alto | matriz){
    Desde i = 0; hasta i = Alto i=i+1{
        Desde j = 0; hasta j = Ancho j=j+1{
            matriz[i][j] = matriz[i][j]*(-1) + 255;
        }
    }
}

ecualizar( matriz, Ancho, Alto | matriz ){
    obtenerEcuacion( matriz, Ancho, Alto | m, b);
    Desde i = 0 hasta i = Alto i= i+1{
        Desde j = 0 hasta j = Ancho j = j+1{
            matriz[i][j] = m*(matriz[i][j]) + b;
        }
    }
}

obtenerEcuacion( matriz, Ancho, Alto | m, b){
    mayor = matriz[0][0];
    menor = matriz[0][0];
    Desde i = 0 hasta i = Ancho i= i+1{
        Desde j = 0 hasta j = Alto j= j+1{
            Si(matriz[i][j] > mayor)
                mayor = matriz[i][j];
            Si(matriz[i][j] < menor)
                menor = matriz[i][j];
        }
    }
    m = (255)/(mayor - menor);
    b = 255-(m*mayor);
}

reduccion( matriz, Ancho, Alto | matriz2 ){
    Desde i = 0 hasta i = Alto i=i+2{
        Desde j = 0 hasta j = Ancho j=j+2{
            matriz2[i][j] = matriz[i][j];
        }
    }
}

ampliacion( matriz, Ancho, Alto | matriz2 ){
    k = 0;
    h = 0;
    Desde i = 0 hasta i = Alto i=i+1{
        Desde j = 0 hasta j = Ancho j=j+1{
            matriz2[h][k] = matriz[i][j];
            k = k + 2;
        }
        h = h + 2;
    }
    k = 1;
    h = 1;
    Desde i = 0 hasta i = Alto i=i+1{
```

```
        Desde j = 0 hasta j = Ancho j =j +1{
            matriz2[h][k] = matriz[i][j];
            k = k + 2;
        }
        h = h + 2;
    }
}
NuevoNombre( | NombreNuevo ){
    imprimir("Ingresar el nuevo nombre del archivo: ");
    leer(NombreNuevo);
}
guardar( NombreNuevo, Matriz, Altura, Ancho | ){
    Abrir(NombreNuevo);
    Desde i = 0 hasta i = Altura*2 i=i+1{
        EscribirenArchivo(Matriz[i], tamañode(entero), Ancho, NombreNuevo);
    }
    Cerrar(NombreNuevo);
}
```

IV. CONCLUSIONES

A lo largo del desarrollo de esta práctica he podido reforzar mis conocimientos adquiridos a lo largo del curso de Fundamentos de Programación, así como también algunos temas vistos en las primeras clases del curso de Programación Aplicada. Además, me ha servido para demostrar la importancia que tiene el diseño al momento de desarrollar un programa informático.

V. RECOMENDACIONES ESPECIALES

A. Ecuaciones

Ecuación punto-pendiente. $y = mx+b$

Pendiente de una recta. $m = (y_2 - y_1) / (x_2 - x_1)$

REFERENCIAS

- [1] P Sznajdleder, Algoritmos a fondo con implementaciones en C y Java. Argentina. Alfaomega, 2012, pp. 188-192