

Practica Final - Informe Práctico

Ricardo Figueiredo Minelli [CP989292] - Mario Rodriguez Ruiz [74737195Y]

31 de mayo de 2016



Índice

1. Regla de tres.	3
2. Completar y extender la clase Lista.	3
3. Modificar la representación de la clase Imagen.	5

Índice de figuras

3.1. error escribirPGMBinario	6
3.2. Compilación	7
3.3. Ejecución Valgrind	7

1. Regla de tres.

- destructor
- constructor de copia
- operador de asignación

```
void Imagen::destruir(){
    if(datos != 0){
        nfilas = 0 ;
        ncolumnas = 0 ;
        delete [] datos[0] ;
        delete [] datos ;
    }
}
```

```
void Imagen::copiaDatos(const Imagen &otra){
    crear(otra.filas(), otra.columnas()) ;
    for(int i = 0 ; i < nfilas ; i++)
        for(int j = 0 ; j < ncolumnas ; j++)
            datos[i][j] = otra.datos[i][j] ;
}
```

```
Imagen &Imagen::operator= (const Imagen &otra) {
    if(this != &otra) {
        // Libera la memoria ocupada
        destruir();
        copiaDatos(otra);
    }
    return *this;
}
```

2. Completar y extender la clase Lista.

- destructor
- constructor de copia
- operador de asignación

```
void Lista::destruir(){
    Celda * aux = cabecera;
    while(cabecera != 0){
        aux = cabecera;
        cabecera = cabecera->siguiente;
        delete aux;
    }
    cabecera = 0;
}
```

```
void Lista::copiaDatos(const Lista &otra){
    Celda *pCelda = otra.cabecera;
    while (pCelda){
        insertar(pCelda->datos) ;
        pCelda = pCelda->siguiente ;
    }
}
```

```
Lista &Lista::operator= (const Lista &otra) {
    if(this != &otra) {
        destruir();
        copiaDatos(otra);
    }
    return *this;
}
```

3. Modificar la representación de la clase Imagen.

```
class Imagen{
    private:
        static const int MAXPIXELS = 1000000;
        byte **datos;
        int nfilas;
```

```
void Imagen::crear(int filas , int columnas){
    destruir() ;
    nfilas = filas ;
    ncolumnas = columnas ;
    datos = new byte *[filas] ;
    datos[0] = new byte[filas*columnas] ;
    for(int i = 1 ; i < filas ; i++){
        datos[i] = datos[i-1] + columnas ;
    }
}
```

```
Imagen Imagen::operator+ (const Imagen &otra) {
    int fil , col , colNegros ;
    byte *negros ;
    col = ncolumnas + otra.columnas() ;
    if(otra.filas() > nfilas){
        fil = otra.filas() ;
        colNegros = ncolumnas ;
    }
    else{
        fil = nfilas ;
        colNegros = otra.columnas() ;
    }
    negros = new byte[colNegros] ;
    for(int i = 0 ; i < colNegros ; i++){
        negros[i] = 0 ;
    }
    Imagen nueva(fil , col) ;
    for(int i = 0 ; i < fil ; i++){
        if(i < nfilas && i < otra.filas()){
            nueva.modificaFila(i , datos[i] , ncolumnas , otra.
                datos[i] , otra.columnas()) ;
        }
        else if(i < nfilas && i >= otra.filas()){
            nueva.modificaFila(i , datos[i] , ncolumnas , negros ,
                colNegros) ;
        }
        else{
            nueva.modificaFila(i , negros , colNegros , otra.datos
                [i] , otra.columnas()) ;
        }
    }
    delete [] negros ;
    return nueva ;
}
```

Hemos tenido algunos problemas con la implementación del método escribirPGMBinario del fichero pgm.cpp, hemos resuelto cambiando el argumento a escribir "datos[0]".

```
bool escribirPGMBinario (const char nombre[], const unsigned char * const *
    datos, int filas, int columnas)
{
    ofstream f(nombre);
    bool res= true;

    if (f) {
        f << "P5" << endl;
        f << columnas << ' ' << filas << endl;
        f << 255 << endl;
        f.write(reinterpret_cast<const char *>(datos[0]), filas *
            columnas);
        if (!f) res=false;
    }
    return res;
}
```

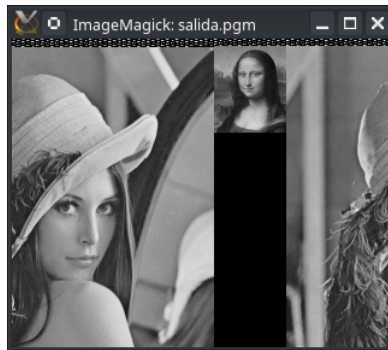


Figura 3.1: error escribirPGMBinario

Capturas de ejecución del programa final, ya corregido todos los fallos:

```
minelli@archMinelli: ~/Documentos/2015-2016/II/2_Semestre/MP/Practica/final/Final/matriz — Konsole
Archivo Editar Ver Bookmarks Preferencias Ayuda
minelli@archMinelli ..emestre/MP/Practica/final/Final/matriz % make clean mrproper
limpiando...
rm obj/*.o
rm lib/*
rm bin/*
Borra imágenes y ficheros .txt que no sirven...
rm: no se puede borrar 'ascii*.txt': No existe el fichero o el directorio
makefile:92: fallo en las instrucciones para el objetivo 'borra'
make: *** [borra] Error 1

2 minelli@archMinelli ..emestre/MP/Practica/final/Final/matriz % make
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -c src/arteASCII.cpp -o obj/arteASCII.o -Iinclude/
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -c src/arteASCII2.cpp -o obj/arteASCII2.o -Iinclude/
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -c src/pgm.cpp -o obj/pgm.o -Iinclude/
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -c src/byte.cpp -o obj/byte.o -Iinclude/
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -c src/lista.cpp -o obj/lista.o -Iinclude/
ar rsv lib/libimagen.a obj/arteASCII.o obj/arteASCII2.o obj/pgm.o obj/byte.o obj/lista.o
ar: creando lib/libimagen.a
a - obj/arteASCII.o
a - obj/arteASCII2.o
a - obj/pgm.o
a - obj/byte.o
a - obj/lista.o
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -o bin/arteASCII obj/arteASCII.o -limgen
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -c src/arteASCII2.cpp -o obj/arteASCII2.o -Iinclude/
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -c bin/arteASCII2.o -limgen -limgen
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -c src/testplano.cpp -o obj/testplano.o -Iinclude/
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -o bin/testplano obj/testplano.o -limgen -limgen
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -c src/testimagen.cpp -o obj/testimagen.o -Iinclude/
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -o bin/testimagen obj/testimagen.o -limgen -limgen
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -c src/testarteASCII.cpp -o obj/testarteASCII.o -Iinclude/
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -o bin/testarteASCII obj/testarteASCII.o -limgen -limgen
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -c src/suma.cpp -o obj/suma.o -Iinclude/
g++ -g -Wall -pedantic -ansi -std=c++11 -Iinclude -o bin/suma obj/suma.o -limgen -limgen
make 5.24s user 0.50s system 91% cpu 6.24s total
minelli@archMinelli ..emestre/MP/Practica/final/Final/matriz %
```

Figura 3.2: Compilación

```
minelli@archMinelli: display salida.pgm — Konsole
Archivo Editar Ver Bookmarks Preferencias Ayuda
minelli@archMinelli ..emestre/MP/Practica/final/Final/matriz % make valgrind
valgrind --leak-check=full --track-origins=yes bin/suma imagenes/gato.pgm imagenes/gato.pgm salida.pgm b
==9324== Memcheck, a memory error detector
==9324== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==9324== Using Valgrind-3.11.0 and libVEX; rerun with -h for copyright info
==9324== Command: bin/suma imagenes/gato.pgm imagenes/gato.pgm salida.pgm b
==9324==
==9324== HEAP SUMMARY:
==9324==    in use at exit: 72,704 bytes in 1 blocks
==9324==   total heap usage: 22 allocs, 21 frees, 363,596 bytes allocated
==9324==
==9324== LEAK SUMMARY:
==9324==    definitely lost: 0 bytes in 0 blocks
==9324==    indirectly lost: 0 bytes in 0 blocks
==9324==    possibly lost: 0 bytes in 0 blocks
==9324==    still reachable: 72,704 bytes in 1 blocks
==9324==    suppressed: 0 bytes in 0 blocks
==9324== Reachable blocks (those to which a pointer was found) are not shown.
==9324== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==9324==
==9324== For counts of detected and suppressed errors, rerun with: -v
==9324== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
minelli@archMinelli ..emestre/MP/Practica/final/Final/matriz % display salida.pgm
```

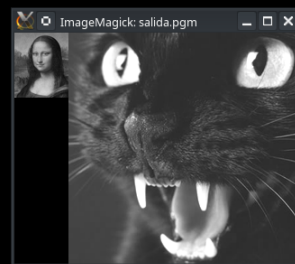


Figura 3.3: Ejecución Valgrind