

1. Código Realizado

1.1. Lista

En lista.h y lista.cpp se encuentran la clase y métodos necesarios para almacenar información en listas enlazadas.

```
Lista::Lista() {
    cabecera = NULL;
}

Lista::Lista(string valor) {
    cabecera = new Celda;
    cabecera->datos = valor;
    cabecera->siguiente = NULL;
}

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

```

void Lista::insertar(string valor){
    Celda* nueva = new Celda;
    nueva->datos = valor;
    nueva->siguiente = NULL;

    if(!cabecera){
        cabecera = nueva;
    }else{
        Celda* aux = cabecera;
        while(aux->siguiente != NULL){
            aux = aux->siguiente;
        }
        aux->siguiente = nueva;
    }
}

string Lista::getCelda(int pos) const{
    string valor;
    int contador = 0;
    Celda* aux = cabecera;
    while(contador < pos && aux){
        aux = aux->siguiente;
        contador++;
    }

    if(aux != NULL){
        valor = aux->datos;
    }

    return valor;
}

int Lista::longitud() const{

    int contador = 0;
    Celda* aux;

    aux = cabecera;

    while(aux != NULL){
        aux = aux->siguiente;
        contador++;
    }

    return contador;
}

```

```

bool Lista::leerLista(const char nombrefichero[]){
    ifstream fin;
    fin.open(nombrefichero);
    if(!fin){
        return false;
    }else{
        string grises;
        int lineas;
        getline(fin,grises);
        fin >> lineas;
        getline(fin,grises);
        if (!fin){
            return false;
        }else {
            int i = 0;
            getline(fin,grises);
            while ((i < lineas)&&(fin)){
                if (grises != ""){
                    insertar(grises);
                    i++;
                }
                getline(fin,grises);
            }
        }
        fin.close();
    }
    return true;
}

```

1.2. Imagen

En imagen.h e imagen.cpp se han alterado los métodos para el uso de memoria dinámica en el programa.

```

class Imagen{
private:
    int MAXPIXELS = 0;
    byte *datos = 0;
    int nfilas;
    int ncolumnas;
public:
    /// Construye una imagen vacía (0 filas, 0 columnas)
    Imagen();

```

La primera imagen que corresponde con imagen.h muestra los cambios realizados para incluir memoria dinámica en los métodos posteriores.

```

Imagen::Imagen(){
    nfilas = 0;
    ncolumnas = 0;
}

Imagen::Imagen(int filas, int columnas){
    nfilas = filas;
    ncolumnas = columnas;
    MAXPIXELS = nfilas * ncolumnas;
    datos = NULL;
}

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

int Imagen::filas(){
    return nfilas;
}

int Imagen::columnas(){
    return ncolumnas;
}

void Imagen::set(int y, int x, byte v){
    datos[y*ncolumnas+x] = v;
}

byte Imagen::get(int y, int x){
    return datos[y*ncolumnas+x];
}

void Imagen::setPos(int i, byte v){
    datos[i] = v;
}

byte Imagen::getPos(int i){
    return datos[i];
}

```

```

bool Imagen::leerImagen(const char nombreFichero[]){
    bool flag = false;
    TipoImagen tipo = infoPGM(nombreFichero, nfilas, ncolumnas);
    Imagen::crear(nfilas, ncolumnas);
    if((tipo == IMG_PGM_BINARIO) && ((nfilas*ncolumnas) <= MAXPIXELS)){
        flag = leerPGMBinario(nombreFichero, datos, nfilas, ncolumnas);
    }else if((tipo == IMG_PGM_TEXTO) && ((nfilas*ncolumnas) <= MAXPIXELS)){
        flag = leerPGMTexto(nombreFichero, datos, nfilas, ncolumnas);
    }
    return flag;
}

bool Imagen::escribirImagen(const char nombreFichero[], bool esBinario){
    bool flag = false;
    if(esBinario)
        flag = escribirPGMBinario(nombreFichero, datos, nfilas, ncolumnas);
    else
        flag = escribirPGMTexto(nombreFichero, datos, nfilas, ncolumnas);
    return flag;
}

Imagen Imagen::plano(int k){
    Imagen img(nfilas, ncolumnas);
    byte aux;
    for(int i = 0; i < nfilas*ncolumnas; i++){
        aux = img.getPos(i);
        if(getbit(datos[i], k)){
            on(aux, 7);
        }else{
            off(aux, 7);
        }
        img.setPos(i, aux);
    }
    return img;
}

bool Imagen::aArteASCII(const char grises[], char arteASCII[], int maxlong){
    int cardinal = 0, contadorRead = 0, aux = 0, contadorWrite = 0;
    bool flag = (nfilas*ncolumnas) < maxlong;
    while (grises[cardinal]!='\0'){
        cardinal++;
    }
    while(contadorRead < nfilas*ncolumnas && contadorWrite < maxlong){
        if(contadorRead*ncolumnas == 0 && contadorRead >= ncolumnas){
            arteASCII[contadorWrite] = 10;
            contadorWrite++;
        }
        aux = datos[contadorRead];
        arteASCII[contadorWrite] = grises[aux*cardinal/256];
        contadorRead++;
        contadorWrite++;
    }
    if(contadorWrite+1 < maxlong){
        arteASCII[contadorWrite] = 10;
        arteASCII[contadorWrite+1] = '\0';
    }
    return flag;
}

```

```

bool Imagen::listaAArteASCII(const Lista celdas){
    bool resultado = true;
    string nombre = "ascii0";
    ofstream fsalida;
    const int MAXLONG = 4501;
    char arteASCII[MAXLONG];

    for(int i = 0; i < celdas.longitud(); i++){
        fsalida.open((nombre+to_string(i+1)).c_str());
        fsalida << "\nLa imagen en arte ASCII es:\n";
        if(aArteASCII(celdas.getCelda(i).c_str(), arteASCII, 4500)){
            fsalida << arteASCII;
        }else{
            fsalida << "La Conversion no ha sido Posible" << endl;
            resultado = false;
        }
        fsalida.close();
    }

    return resultado;
}

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

```

1.3. Makefile

El fichero makefile ha sido alterado para compilar los nuevos archivos incluidos en el programa.


```

SRC = src
INC = include
OBJ = obj
BIN = bin
LIB = lib

CXX = g++
CPPFLAGS = -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -I$(INC)

all: $(BIN)/arteASCII $(BIN)/testarteASCII $(BIN)/testplano $(BIN)/arteASCII2

$(BIN)/testplano: $(OBJ)/testplano.o $(LIB)/libimagen.a
    $(CXX) -o $(BIN)/testplano $(OBJ)/testplano.o -L$(LIB) -limagen

$(BIN)/testarteASCII: $(OBJ)/testarteASCII.o $(LIB)/libimagen.a
    $(CXX) -o $(BIN)/testarteASCII $(OBJ)/testarteASCII.o -L$(LIB) -limagen

$(BIN)/arteASCII: $(OBJ)/arteASCII.o $(LIB)/libimagen.a
    $(CXX) -o $(BIN)/arteASCII $(OBJ)/arteASCII.o -L$(LIB) -limagen

$(BIN)/arteASCII2: $(OBJ)/arteASCII2.o $(LIB)/libimagen.a
    $(CXX) -o $(BIN)/arteASCII2 $(OBJ)/arteASCII2.o -L$(LIB) -limagen

$(LIB)/libimagen.a: $(OBJ)/pgm.o $(OBJ)/imagen.o $(OBJ)/byte.o $(OBJ)/lista.o
    ar rsv $(LIB)/libimagen.a $(OBJ)/pgm.o $(OBJ)/imagen.o $(OBJ)/byte.o $(OBJ)/lista.o

$(OBJ)/testplano.o: $(SRC)/testplano.cpp
    $(CXX) -c $(SRC)/testplano.cpp -o $(OBJ)/testplano.o $(CPPFLAGS)

$(OBJ)/testarteASCII.o: $(SRC)/testarteASCII.cpp
    $(CXX) -c $(SRC)/testarteASCII.cpp -o $(OBJ)/testarteASCII.o $(CPPFLAGS)

$(OBJ)/arteASCII.o: $(SRC)/arteASCII.cpp
    $(CXX) -c $(SRC)/arteASCII.cpp -o $(OBJ)/arteASCII.o $(CPPFLAGS)

$(OBJ)/arteASCII2.o: $(SRC)/arteASCII2.cpp
    $(CXX) -c $(SRC)/arteASCII2.cpp -o $(OBJ)/arteASCII2.o $(CPPFLAGS)

$(OBJ)/imagen.o: $(SRC)/imagen.cpp
    $(CXX) -c $(SRC)/imagen.cpp -o $(OBJ)/imagen.o $(CPPFLAGS)

$(OBJ)/pgm.o: $(SRC)/pgm.cpp
    $(CXX) -c $(SRC)/pgm.cpp -o $(OBJ)/pgm.o $(CPPFLAGS)

$(OBJ)/byte.o: $(SRC)/byte.cpp
    $(CXX) -c $(SRC)/byte.cpp -o $(OBJ)/byte.o $(CPPFLAGS)

$(OBJ)/lista.o: $(SRC)/lista.cpp
    $(CXX) -c $(SRC)/lista.cpp -o $(OBJ)/lista.o $(CPPFLAGS)

```

```

# ***** documentación *****

documentacion:
    doxygen doc/doxys/Doxyfile

# ***** Limpieza *****

clean:
    echo "Limpiando..."
    rm -rf $(OBJ)/* $(LIB)/*

mrproper: clean
    rm -rf $(BIN)/* doc/html/* *.pgm *1 *2 *3 *4

```

2. Ejecución del Programa

2.1. arteASCII

```

rafa@rafa-PC:~/Escritorio/planosyascii$ ./bin/arteASCII
Introduce el Nombre y la Ruta de la Imagen de Entrada: imagenes/gio.pgm
Introduce el Nombre y la Ruta del Archivo de Entrada: grises.txt
Introduce el Nombre para los Ficheros: gio
rafa@rafa-PC:~/Escritorio/planosyascii$

```



2.3. Makefile

```
rafa@rafa-PC:~/Escritorio/planosyascii$ make all
g++ -c src/arteASCII.cpp -o obj/arteASCII.o -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude
g++ -c src/pgm.cpp -o obj/pgm.o -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude
g++ -c src/imagen.cpp -o obj/imagen.o -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude
g++ -c src/byte.cpp -o obj/byte.o -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude
g++ -c src/lista.cpp -o obj/lista.o -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude
ar rsv lib/libimagen.a obj/pgm.o obj/imagen.o obj/byte.o obj/lista.o
ar: creando lib/libimagen.a
a - obj/pgm.o
a - obj/imagen.o
a - obj/byte.o
a - obj/lista.o
g++ -o bin/arteASCII obj/arteASCII.o -Llib/ -limagen
g++ -c src/testarteASCII.cpp -o obj/testarteASCII.o -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude
g++ -o bin/testarteASCII obj/testarteASCII.o -Llib/ -limagen
g++ -c src/testplano.cpp -o obj/testplano.o -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude
g++ -o bin/testplano obj/testplano.o -Llib/ -limagen
g++ -c src/arteASCII2.cpp -o obj/arteASCII2.o -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude
g++ -o bin/arteASCII2 obj/arteASCII2.o -Llib/ -limagen
```

```
rafa@rafa-PC:~$ cd "/home/rafa/Escritorio/planosyascii"
rafa@rafa-PC:~/Escritorio/planosyascii$ make mrproper
echo "Limpiando..."
Limpiando...
rm -rf obj/* lib/*
rm -rf bin/* doc/html/* *.pgm *1 *2 *3 *4
```

2.4. Valgrind

```
HEAP SUMMARY:
  in use at exit: 0 bytes in 0 blocks
  total heap usage: 36 allocs, 36 frees, 66,310 bytes allocated

All heap blocks were freed -- no leaks are possible
```

3. Complicaciones en la Realización

3.1. Saltos de Línea

Al ejecutar el programa, las imágenes en ASCII salen con los saltos de línea del fichero. Para solucionar esto, hay que usar el comando

```
Perl -pi -w -e 's/\r\n/\n/g;' grises.txt
```

En el terminal de comandos para arreglar el archivo y que no dé es error. Una vez hecho, la salida del programa es correcta.

3.2. Valgrind

Al ejecutar el programa con Valgrind obtenemos el mensaje

```
All heap blocks were freed – no leaks are posible
```

Esto indica que el programa reserva y libera memoria correctamente. Sin embargo, obtenemos mensajes de error en el código según Valgrind. Al revisar las líneas indica errores en todas las líneas del código, incluyendo las que solo contienen una asignación de una variable o un cierre de llaves. Cabe destacar que el programa funciona y el compilador no muestra error alguno.