

INFORME PRÁCTICA 6 FINAL - METODOLOGÍA DE LA PROGRAMACIÓN

Para empezar, queremos dejar claro que la práctica ha sido realizada exclusivamente por nosotros dos, dedicando el mismo trabajo por parte de ambos.

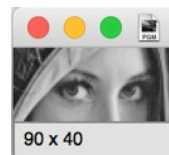
También hemos de afirmar que apenas hemos tenido problemas para la implementación de esta práctica, y es que además de haber podido reciclar código y métodos de prácticas anteriores, en todo momento hemos llevado control de las escasas fugas de memoria (o 'memory leaks').

Los únicos contratiempos que nos han frenado un poco a la hora de finalizar la práctica han sido detalles minúsculos sin mucha importancia (más bien despistes) por simples errores a la hora de declarar un método, comprometiendo así los objetos que llaman a dicho método y se benefician de su funcionamiento.

```
nacho@MacBook-Pro-de-Ignacio ~/E/P/p/i/matriz> make
g++ -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude src/arteASCII2.cpp -o obj/arteASCII2.o
g++ -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude src/imagen.cpp -o obj/imagen.o
g++ -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude src/pgm.cpp -o obj/pgm.o
g++ -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude src/byte.cpp -o obj/byte.o
g++ -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude src/lista.cpp -o obj/lista.o
ar rvs lib/libimagen.a obj/imagen.o obj/pgm.o obj/byte.o obj/lista.o
ar: creating archive lib/libimagen.a
a - obj/imagen.o
a - obj/pgm.o
a - obj/byte.o
a - obj/lista.o
g++ obj/arteASCII2.o -limagen -o bin/arteASCII2 -lstdc++ -lstdc++
g++ -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude src/testimagen.cpp -o obj/testimagen.o
g++ obj/testimagen.o -limagen -o bin/testimagen -lstdc++ -lstdc++
g++ -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude src/suma.cpp -o obj/suma.o
g++ obj/suma.o -limagen -o bin/suma -lstdc++ -lstdc++
g++ -ansi -pedantic -Wall -Wextra -Wwrite-strings -Werror -g -c -std=c++0x -Iinclude src/testFinal.cpp -o obj/testFinal.o
g++ obj/testFinal.o -limagen -o bin/testFinal -lstdc++ -lstdc++
```

- Comprobación de testimagen para lena.pgm:

```
[nacho@MacBook-Pro-de-Ignacio ~/E/P/p/i/matriz> bin/testimagen
degradado.pgm guardado correctamente
trozo.pgm guardado correctamente
```



- Comprobación ArteASCII:

```

[nacho@MacBook-Pro-de-Ignacio ~/E/P/p/i/matriz] bin/arteASCII2 imagenes/gato.pgm grises.txt
Exitó convirtiendo la imagen a aArteASCII 1.
Exitó convirtiendo la imagen a aArteASCII 2.
Exitó convirtiendo la imagen a aArteASCII 3.
Exitó convirtiendo la imagen a aArteASCII 4.
Exitó en la conversión.
Chequee los ficheros asciiX.txt

```



- Comprobación de suma:

```

[nacho@MacBook-Pro-de-Ignacio ~/E/P/p/i/matriz] valgrind bin/suma imagenes/lena.pgm imagenes/gato.pgm resultado.pgm t
==1876== Memcheck, a memory error detector
==1876== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==1876== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==1876== Command: bin/suma imagenes/lena.pgm imagenes/gato.pgm resultado.pgm t
==1876==
--1876-- run: /usr/bin/dsymutil "bin/suma"

Concatenación realizada. Comprueba el fichero de salida.
==1876==
==1876== HEAP SUMMARY:
==1876==    in use at exit: 99,053 bytes in 191 blocks
==1876==   total heap usage: 289 allocs, 98 frees, 378,861 bytes allocated
==1876==
==1876== LEAK SUMMARY:
==1876==    definitely lost: 0 bytes in 0 blocks
==1876==    indirectly lost: 0 bytes in 0 blocks
==1876==    possibly lost: 2,064 bytes in 1 blocks
==1876==    still reachable: 76,800 bytes in 2 blocks
==1876==    suppressed: 20,189 bytes in 188 blocks
==1876== Rerun with --leak-check=full to see details of leaked memory
==1876==
==1876== For counts of detected and suppressed errors, rerun with: -v
==1876== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

```

