

# **DOCUMENTACIÓN PROYECTO FINAL MP**

1.

Alumno: Raúl Castro Moreno.

Grupo: D-2 (VIERNES).

2.

Al empezar con la práctica ya me aparecieron los primeros problemas con la clase Partícula, debido a la implementación del color de la partícula, ya que como era algo de la parte de miniwin, no sabía muy bien como hacerlo en ese momento, lo cual pregunte y me sugirieron hacer un vector con los posibles colores dentro. El otro problema surge a la hora de hacer que se muevan las partículas, ya que nos dieron el código, pero estaba en un main.cpp, y ese mover había que implementarlo como método de la clase y me costo un poco. Por lo demás sobre esa clase, era muy sencillo lo demás.

Ahora, pasando a la clase Conjunto de Partículas, si encontramos muchos más problemas, ya que aquí viene el uso de gestión de memoria dinámica. Empezemos por lo que me dio algunos problemitas pero no muy altos, y estos son los operadores que había que sobrecargar, ya que era algo que habíamos dado nuevo en teoría, y no tenía mucho idea de ellos, sobre todo del operador "<<", ya que el uso del flujo y demás no es algo que nos resultara familiar y demás, pero al final, con la ayuda de los apuntes y demás, conseguí implementarlos correctamente, menos uno que hablaremos más adelante. Después de eso, fui a implementar los métodos de memoria dinámica, aquí ya vino la salsa porque había que jugar con el puntero que era un dato privado de la clase, y el tener cuidado para no borrar los datos importantes, hizo que, digamos, sobrecuidara ese puntero en la gestión de memoria dinámica, pero eso hizo el efecto contrario, ya que a parte de mal implementar los métodos, perdía memoria según el valgrind. La cosa es que me funcionaba el programa perfectamente, entonces lo dejé así. Por último, el problema que me vino a la hora de buscar la colisión de las partículas, que fue por que estaba ejecutando el bucle for mal, ya que estaba mirando si colisionaban todas las partículas,

incluyendo las que ya había comprobado, con lo que solo hubo que cambiar las condiciones del segundo bucle for.

Ahora con la clase Simulador, fue para mí de lo más sencillo de hacer en este proyecto, solo tuve un problema, el cual fue a la hora de hacer colisionar las partículas móviles con las fijas, ya que en mi cabeza lo que había puesto funcionaba, pero tomaba en cuenta la misma partícula del conjunto y le cambiaba a esa misma la velocidad, claro, luego de consultarlo con el profesor vi que eso era un error ya que no podía modificarlo, y termine creando una partícula nueva, la cual copiaba a la que tenía que rebotar, pero le cambiaba la velocidad para que hiciera el rebote elástico y la reemplazaba con el método que ya tenia por esa misma.

Una vez hecho todo esto, comencé a usar el main.cpp que nos proporcionó el profesor, el cual era el que tenía que funcionar para la entrega. Las 3 opciones me funcionaban a la perfección, el problema vino al pasarle el valgrind, ya que perdía bytes en todas las opciones. Esto me hizo remirar mis métodos privados de gestión de memoria, los cuáles revise varias veces, reescribi y demás pero no conseguí nada. Finalmente , acudiendo al profesor, y con ayuda de algún que otro compañero vi el error, el cuál venía de parte del operator= de la clase Conjunto de Partículas, donde , me falto liberar la memoria antes de hacer la copia, hay se perdía todo. Todo por culpa de no saber correctamente el uso de ese operador, pero bueno, finalmente he conseguido una ejecución correcta y sin pérdidas de memoria.

### 3.

Sobre las tareas adicionales añadidas, que no se si esto es adicional o no pero lo pongo aquí, escogí la opción 4 del pdf de la parte B , la cual era el añadir y borrar partículas mediante el uso de teclas. En el main.cpp es la opción 4, con la cuál puedes usar esta función.

La explicación consiste en que esa opción crea un simulador, en el cual, mediante las funciones que he añadido:

**Flecha hacia arriba**= Añade una Partícula Móvil.

**Flecha hacia abajo**= Borra una Partícula Móvil.

**Flecha hacia la derecha**= Añade una Partícula Fija.

**Flecha hacia la izquierda**= Borra una Partícula Fija.

Decir también, que le he puesto un contador para que se cierre solo al igual que las otras opciones se cierran, pero si pulsas la tecla ESCAPE que puedes forzar el final de la ejecución.

## 4.

### Capturas de Valgrind en cada opción.

#### Opción 1.

```
Actividades Terminal mié 12:57
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/NetBeansProjects/ProyectoFinal

Partícula: 4 ( 711, 482) V(x): 5 V(y): 5
Partícula: 5 ( 746, 256) V(x): 2 V(y): 3

Capacidad: 10
Nro de Partículas: 10
Partícula: 0 ( 229, 225) V(x): 2 V(y): 4
Partícula: 1 ( 231, 229) V(x): 2 V(y): 4
Partícula: 2 ( 44, 127) V(x): 4 V(y): 3
Partícula: 3 ( 425, 12) V(x): 2 V(y): -2
Partícula: 4 ( 711, 482) V(x): 5 V(y): 5
Partícula: 5 ( 746, 256) V(x): 2 V(y): 3
Partícula: 6 ( 44, 127) V(x): 4 V(y): 3
Partícula: 7 ( 425, 12) V(x): 2 V(y): -2
Partícula: 8 ( 711, 482) V(x): 5 V(y): 5
Partícula: 9 ( 746, 256) V(x): 2 V(y): 3

^C==1892==
==1892== Process terminating with default action of signal 2 (SIGINT)
==1892== at 0x5AC410D: _lll_lock_wait (lowlevellock.S:135)
==1892== by 0x5ABD022: pthread_mutex_lock (pthread_mutex_lock.c:78)
==1892== by 0x60CEA7: xcb_writev (in /usr/lib/x86_64-linux-gnu/libxcb.so.1.1.0)
==1892== by 0x4E7A145: XSend (in /usr/lib/x86_64-linux-gnu/libX11.so.6.3.0)
==1892== by 0x4E7A66B: XReply (in /usr/lib/x86_64-linux-gnu/libX11.so.6.3.0)
==1892== by 0x4E761CC: XSync (in /usr/lib/x86_64-linux-gnu/libX11.so.6.3.0)
==1892== by 0x4E59E1D: XCloseDisplay (in /usr/lib/x86_64-linux-gnu/libX11.so.6.3.0)
==1892== by 0x108E1E: main (minlwin.cpp:740)
==1892==
==1892== HEAP SUMMARY:
==1892== in use at exit: 86,234 bytes in 33 blocks
==1892== total heap usage: 106 allocs, 73 frees, 169,166 bytes allocated
==1892==
==1892== 288 bytes in 1 blocks are possibly lost in loss record 22 of 38
==1892== at 0x4C31B25: calloc (in /usr/lib/valgrind/vgpreload_memcheck-and64-linux.so)
==1892== by 0x40134A6: allocate_dtv (dl-tls.c:286)
==1892== by 0x40134A6: _dl_allocate_tls (dl-tls.c:530)
==1892== by 0x5AB8227: allocate_stack (allocatestack.c:627)
==1892== by 0x5AB8227: pthread_create@@GLIBC_2.2.5 (pthread_create.c:644)
==1892== by 0x108C72: maybe_call_main() (minlwin.cpp:673)
==1892== by 0x108CE4: _process_event() (minlwin.cpp:689)
==1892== by 0x108D06: main (minlwin.cpp:733)
==1892==
==1892== LEAK SUMMARY:
==1892== definitely lost: 0 bytes in 0 blocks
==1892== indirectly lost: 0 bytes in 0 blocks
==1892== possibly lost: 288 bytes in 1 blocks
==1892== still reachable: 85,946 bytes in 32 blocks
==1892== suppressed: 0 bytes in 0 blocks
==1892== Reachable blocks (those to which a pointer was found) are not shown.
==1892== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==1892==
==1892== For counts of detected and suppressed errors, rerun with: -v
==1892== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)

raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/NetBeansProjects/ProyectoFinal$
```

#### Opción 2.

```
Actividades Terminal mié 12:57
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/NetBeansProjects/ProyectoFinal

==1892== by 0x5AB8227: allocate_stack (allocatestack.c:627)
==1892== by 0x5AB8227: pthread_create@@GLIBC_2.2.5 (pthread_create.c:644)
==1892== by 0x108C72: maybe_call_main() (minlwin.cpp:673)
==1892== by 0x108CE4: _process_event() (minlwin.cpp:689)
==1892== by 0x108D06: main (minlwin.cpp:733)
==1892==
==1892== LEAK SUMMARY:
==1892== definitely lost: 0 bytes in 0 blocks
==1892== indirectly lost: 0 bytes in 0 blocks
==1892== possibly lost: 288 bytes in 1 blocks
==1892== still reachable: 85,946 bytes in 32 blocks
==1892== suppressed: 0 bytes in 0 blocks
==1892== Reachable blocks (those to which a pointer was found) are not shown.
==1892== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==1892==
==1892== For counts of detected and suppressed errors, rerun with: -v
==1892== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)

raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/NetBeansProjects/ProyectoFinal$ valgrind --leak-check=full ./dist/Debug/GNU-Linux/proyectoFinal
==1966== Memcheck, a memory error detector
==1966== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==1966== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==1966== Command: ./dist/Debug/GNU-Linux/proyectoFinal
==1966==

Opcion a probar: 2

** Prueba Conjunto Partículas **
==1966==
==1966== HEAP SUMMARY:
==1966== in use at exit: 448 bytes in 2 blocks
==1966== total heap usage: 3,644 allocs, 3,642 frees, 1,284,346 bytes allocated
==1966==
==1966== 288 bytes in 1 blocks are possibly lost in loss record 2 of 2
==1966== at 0x4C31B25: calloc (in /usr/lib/valgrind/vgpreload_memcheck-and64-linux.so)
==1966== by 0x40134A6: allocate_dtv (dl-tls.c:286)
==1966== by 0x40134A6: _dl_allocate_tls (dl-tls.c:530)
==1966== by 0x5AB8227: allocate_stack (allocatestack.c:627)
==1966== by 0x5AB8227: pthread_create@@GLIBC_2.2.5 (pthread_create.c:644)
==1966== by 0x108C72: maybe_call_main() (minlwin.cpp:673)
==1966== by 0x108CE4: _process_event() (minlwin.cpp:689)
==1966== by 0x108D06: main (minlwin.cpp:733)
==1966==
==1966== LEAK SUMMARY:
==1966== definitely lost: 0 bytes in 0 blocks
==1966== indirectly lost: 0 bytes in 0 blocks
==1966== possibly lost: 288 bytes in 1 blocks
==1966== still reachable: 160 bytes in 1 blocks
==1966== suppressed: 0 bytes in 0 blocks
==1966== Reachable blocks (those to which a pointer was found) are not shown.
==1966== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==1966==
==1966== For counts of detected and suppressed errors, rerun with: -v
==1966== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 280 from 1)

raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/NetBeansProjects/ProyectoFinal$
```

## Opción 3.

```
Actividades Terminal mié 12:58 ●
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/NetBeansProjects/ProyectoFinal

Particula: 2 ( 98, 558) V(x): 2 V(y): -4
Particula: 3 ( 165, 549) V(x): 1 V(y): 1
Particula: 4 ( 59, 593) V(x): 4 V(y): -1
Particula: 5 ( 525, 220) V(x): 4 V(y): -1
Particula: 6 ( 264, 232) V(x): 4 V(y): 5
Particula: 7 ( 61, 502) V(x): -1 V(y): 1
Particula: 8 ( 338, 716) V(x): 5 V(y): -3
Particula: 9 ( 308, 464) V(x): -3 V(y): -4
Particula: 10 ( 495, 12) V(x): 2 V(y): 5
Particula: 11 ( 189, 763) V(x): 5 V(y): -1
Particula: 12 ( 486, 758) V(x): 4 V(y): -2
Particula: 13 ( 35, 599) V(x): 2 V(y): -1
Particula: 14 ( 221, 657) V(x): 1 V(y): 2
Particula: 15 ( 19, 607) V(x): -1 V(y): -1
Particula: 16 ( 552, 540) V(x): -4 V(y): 4
Particula: 17 ( 583, 190) V(x): 5 V(y): 4
Particula: 18 ( 114, 765) V(x): -2 V(y): 1
Particula: 19 ( 428, 297) V(x): -5 V(y): 2
Particula: 20 ( 293, 389) V(x): -4 V(y): 1
Particula: 21 ( 272, 402) V(x): -2 V(y): 4
Particula: 22 ( 159, 30) V(x): -4 V(y): 3
Particula: 23 ( 164, 400) V(x): 5 V(y): -1
Particula: 24 ( 310, 618) V(x): -4 V(y): -3
Particula: 25 ( 95, 272) V(x): -3 V(y): -4
Particula: 26 ( 542, 246) V(x): 3 V(y): -4
Particula: 27 ( 396, 710) V(x): 3 V(y): 3
Particula: 28 ( 191, 237) V(x): 3 V(y): 3
Particula: 29 ( 97, 186) V(x): 5 V(y): -2

==2032== HEAP SUMMARY:
==2032== in use at exit: 448 bytes in 2 blocks
==2032== total heap usage: 2,141 allocs, 2,139 frees, 224,050 bytes allocated
==2032==
==2032== 288 bytes in 1 blocks are possibly lost in loss record 2 of 2
==2032== at 0x4C31B25: calloc (in /usr/lib/valgrind/vgpreload_mcheck-amd64-linux.so)
==2032== by 0x40134A6: allocate_dtv (dl-tls.c:286)
==2032== by 0x40134A6: _dl_allocate_tls (dl-tls.c:538)
==2032== by 0x5AB8227: allocate_stack (allocatestack.c:627)
==2032== by 0x5AB8227: pthread_create@@GLIBC_2.2.5 (pthread_create.c:644)
==2032== by 0x10BC72: _maybe_call_main() (miniwin.cpp:673)
==2032== by 0x10BCF4: _process_event() (miniwin.cpp:689)
==2032== by 0x10BD06: main (miniwin.cpp:733)
==2032==
==2032== LEAK SUMMARY:
==2032== definitely lost: 0 bytes in 0 blocks
==2032== indirectly lost: 0 bytes in 0 blocks
==2032== possibly lost: 288 bytes in 1 blocks
==2032== still reachable: 160 bytes in 1 blocks
==2032== suppressed: 0 bytes in 0 blocks
==2032== Reachable blocks (those to which a pointer was found) are not shown.
==2032== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==2032==
==2032== For counts of detected and suppressed errors, rerun with: -v
==2032== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 500 from 1)
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/NetBeansProjects/ProyectoFinal$
```

## Opción 4.

```
Actividades Terminal mié 12:58 ●
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/NetBeansProjects/ProyectoFinal

Particula: 12 ( 377, 763) V(x): 5 V(y): -1
Particula: 13 ( 138, 699) V(x): -5 V(y): 5
Particula: 14 ( 57, 288) V(x): -3 V(y): 3
Particula: 15 ( 50, 40) V(x): -2 V(y): -4
Particula: 16 ( 361, 284) V(x): -1 V(y): -4
Particula: 17 ( 376, 378) V(x): 4 V(y): -2
Particula: 18 ( 267, 758) V(x): 4 V(y): 1
Particula: 19 ( 522, 503) V(x): -2 V(y): -3
Particula: 20 ( 328, 352) V(x): -4 V(y): -4
Particula: 21 ( 526, 474) V(x): -4 V(y): -5
Particula: 22 ( 539, 680) V(x): -1 V(y): 4
Particula: 23 ( 350, 227) V(x): 4 V(y): -1

terminate called without an active exception
==2070== Process terminating with default action of signal 6 (SIGABRT)
==2070== at 0x5D1BE97: raise (raise.c:51)
==2070== by 0x5D12800: abort (abort.c:79)
==2070== by 0x5200956: ??? (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==2070== by 0x5206AB5: ??? (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==2070== by 0x5206AF0: std::terminate() (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==2070== by 0x52064B9: _cxx_personality_v0 (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==2070== by 0x58AB707: ??? (in /lib/x86_64-linux-gnu/libgcc_s.so.1)
==2070== by 0x58ABCFB: _Unwind_ForceUnwind (in /lib/x86_64-linux-gnu/libgcc_s.so.1)
==2070== by 0x5AC3F0f: _pthread_unwind (unwind.c:121)
==2070== by 0x5E3B85E: _pthread_unwind (forward.c:202)
==2070== by 0x5E02734: __libc_enable_asynccancel (cancellation.S:81)
==2070== by 0x5DE86D4: writev (writev.c:26)
==2070==
==2070== HEAP SUMMARY:
==2070== in use at exit: 100,994 bytes in 79 blocks
==2070== total heap usage: 4,053 allocs, 3,974 frees, 299,646 bytes allocated
==2070==
==2070== 288 bytes in 1 blocks are possibly lost in loss record 28 of 44
==2070== at 0x4C31B25: calloc (in /usr/lib/valgrind/vgpreload_mcheck-amd64-linux.so)
==2070== by 0x40134A6: allocate_dtv (dl-tls.c:286)
==2070== by 0x40134A6: _dl_allocate_tls (dl-tls.c:538)
==2070== by 0x5AB8227: allocate_stack (allocatestack.c:627)
==2070== by 0x5AB8227: pthread_create@@GLIBC_2.2.5 (pthread_create.c:644)
==2070== by 0x10BC72: _maybe_call_main() (miniwin.cpp:673)
==2070== by 0x10BCF4: _process_event() (miniwin.cpp:689)
==2070== by 0x10BD06: main (miniwin.cpp:733)
==2070==
==2070== LEAK SUMMARY:
==2070== definitely lost: 0 bytes in 0 blocks
==2070== indirectly lost: 0 bytes in 0 blocks
==2070== possibly lost: 288 bytes in 1 blocks
==2070== still reachable: 100,706 bytes in 78 blocks
==2070== suppressed: 0 bytes in 0 blocks
==2070== Reachable blocks (those to which a pointer was found) are not shown.
==2070== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==2070==
==2070== For counts of detected and suppressed errors, rerun with: -v
==2070== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 717 from 1)
Abortado ('core' generado)
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/NetBeansProjects/ProyectoFinal$ cd NetBeansProjects/ProyectoFinal$
```

**Por último:** Sobre lo de la auto-evaluación, me gustaría ponerme un sobresaliente, ya no por que la ejecución vaya correcta y no haya ninguna pérdida de memoria, sino ya más por el esfuerzo que supone hacer estos proyectos, que le he dedicado mucho tiempo, tanto en mi casa como asistiendo a clases para resolver dudas en un horario que no era el mío.

**P.D. :** Le iba a añadir lo que hablé con el profesor en clase sobre que dejara una partícula móvil, un rastro de solo 3 partículas fijas, el cual se iba actualizando con el movimiento de la partícula móvil, pero al final, después de intentarlo 2 veces, ver que no funcionaba, y ver que la agenda esta llena de exámenes, he preferido no implementarlo por usar ese tiempo en las demás asignaturas, o en estudiar el examen de teoría de esta asignatura.