

## Informe de responsabilidades

La subida de este documento firmado al repositorio es obligatoria para acceder a la sustentación y debe hacerse a más tardar el día anterior a la misma. Es obligatorio solo para entregas en pareja.

Nombre del componente desarrollado (clase , métodos o funciones)	Descripción breve de dicho componente	Porcentaje de complejidad respecto al desarrollo global	Desarrollador principal (nombre de sólo una persona)
Bullet.h - Bullet.cpp	Aquí está el proyectil estándar. Se mueve en línea recta a velocidad constante. Detecta colisiones con enemigos o con el jugador para aplicar daño y autodestruirse.	2%	Jeisson Martinez
BunkerBossItem.h - BunkerBossItem.cpp	Es el jefe final del Nivel 1, es una Entidad estática de alta resistencia. Posee animaciones de estados simple (inactivo, atacando, destruido) y dispara proyectiles hacia el jugador en intervalos definidos.	4%	Jeisson Martinez
EnemyItem.h - EnemyItem.cpp	Representa a los soldados enemigos del nivel 1 con IA reactiva para esquivar disparos si se detecta un evento de ataque. También gestionan su propia animación de muerte.	6%	Jeisson Martinez
GameWindow.h - GameWindow.cpp	Gestiona el bucle principal del juego. inicializa los niveles, maneja las transiciones de estado, controla la música de fondo y administra la memoria de todas las entidades	20%	Jeisson Martinez
interfaz.h - interfaz.cpp	Provee el menú principal con opciones para iniciar el juego "Jugar" o ver la información general del proyecto "Créditos"	2%	Jeisson Martinez
PlayerItem.h - PlayerItem.cpp	Representa al jugador en el modo de desplazamiento lateral, Implementa la lógica de movimiento físico como lo es la gravedad, salto parabólico, fricción y colisiones con el suelo/plataformas. También maneja las animaciones de	12%	Jeisson Martinez

	correr/saltar/agacharse.		
Projectile.h - Projectile.cpp	Proyectil especial que implementa un movimiento parabólico afectado por la gravedad. Al tocar el suelo detona causando daño de Área a los enemigos que se encuentran en el rango de explosión.	4%	Jeisson Martinez

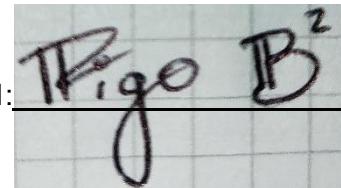
FlameArea.h – FlameArea.cpp	Implementa el área de fuego utilizada por el jugador en el Nivel 2. Representa un cono de daño corto que aparece frente al jugador al disparar el lanzallamas. Gestiona su propia textura, rotación, colisión precisa con enemigos, aplicación de daño por ráfaga y autodestrucción usando temporizadores. Es un módulo clave para el combate del modo Top-Down en el desafío final.	8%	Rigoberto Berrio
TopDownPlayerItem.h – TopDownPlayerItem.cpp	Implementa el jugador del modo Top-Down del Nivel 2. Gestiona el movimiento con WASD, aceleración, límites del mapa, animación del sprite y detección de colisiones. Emite señales de daño y muerte para ser manejadas por GameWindow. Centraliza la lógica de vida, invulnerabilidad temporal, respuesta a ataques y orientación del arma. Es el núcleo del control del jugador en la modalidad de supervivencia.	10%	Rigoberto Berrio
TopDownEnemy.h – TopDownEnemy.cpp	Representa a los enemigos del Nivel 2 con comportamiento de persecución hacia el jugador. Posee IA básica orientada al movimiento, cálculo de dirección, animación del sprite, detección de daño por el lanzallamas y secuencia de muerte. Se integra con el modo supervivencia mediante señales de muerte para que GameWindow gestione las oleadas. Son esenciales para generar dificultad creciente en el desafío Top-Down	10%	Rigoberto Berrio
GameWindow.h – GameWindow.cpp	Implementa toda la lógica del Nivel 2: inicialización del mapa Top-Down, creación del jugador, HUD especial de tiempo restante, temporizador de supervivencia,	20%	Rigoberto Berrio

	generación de oleadas mediante spawnSurvivalWave(), manejo de survivalEnemies_, actualización del contador con updateSurvivalTimer() y lógica del lanzallamas dentro del bucle onTick(). Coordina audio, colisiones y ciclo de juego hasta la victoria o Game Over		
Creditos.h – Creditos.cpp	Implementa la ventana emergente de créditos del juego. Utiliza QDialog y un QTextEdit con contenido HTML para mostrar información estructurada de autores, tecnologías y asignatura. Incluye la interfaz gráfica, maquetación con QVBoxLayout, manejo de eventos del botón de cierre y la integración con el menú principal. Facilita la navegación del usuario dentro del sistema de menús del proyecto	2%	Rigoberto Berrio

**Tabla resumen**

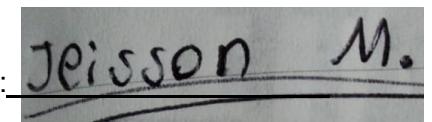
<b>Nombre del integrante (Una fila por integrante del equipo)</b>	<b>Nombre de todos los componentes desarrollados</b>	<b>Porcentaje total desarrollado (la suma de los ítems es 100)</b>
Jeisson Martinez	Bullet.h;Bullet.cpp, BunkerBossItem.h;BunkerBossItem.cpp, EnemyItem.h;EnemyItem.cpp, interfaz.h;interfaz.cpp, GameWindow.h;GameWindow.cpp, PlayerItem.h;PlayerItem.cpp, Projectile.h;Projectile.cpp	50%
Rigoberto Berrio	FlameArea.h;FlameArea.cpp, TopDownPlayerItem.h;TopDownPlayerItem.cpp, TopDownEnemy.h;TopDownEnemy.cpp, GameWindow.h;GameWindow.cpp, Creditos.h;Creditos.cpp	50%
	Total	100%

Nombre y firma autógrafa del integrante 1:



Rigo B<sup>2</sup>

Nombre y firma autógrafa del integrante 2:



Jeisson M.

**Nota:** La repartición de responsabilidades especificada en este formato no exime a ninguno de los miembros del equipo de la responsabilidad de conocer y explicar el análisis y diseño de las estrategias que fundamentan toda la solución entregada.