

UNIVERSIDAD DE ANTIOQUÍA

Ingeniería de Telecomunicaciones

Informática II

Proyecto final

Informe Técnico en C++

Autor: Andrés Felipe Sierra Fernández

Profesor: Anibal Jose Guerra

Fecha: 18 de noviembre de 2025



**UNIVERSIDAD
DE ANTIOQUIA**

1 8 0 3

1 Resumen

Este videojuego está inspirado en la Guerra de los Mil Días, un conflicto civil colombiano ocurrido entre 1899 y 1902. El jugador asume el papel de un joven liberal que, decidido a evitar más muertes, debe cumplir misiones que pondrán a prueba su sigilo, ingenio y valentía. A lo largo de tres niveles, deberá infiltrarse entre enemigos para entregar mensajes secretos, cruzar ríos peligrosos durante tormentas y liderar a su grupo en la batalla final por la libertad. Cada etapa introduce nuevas perspectivas, retos físicos y estrategias.

2 Descripción general del proyecto

Nivel 1: El Mensajero del Campamento

Contexto: En medio del fragor de la Guerra de los Mil Días, los liberales se refugian en campamentos improvisados entre las montañas. Las tropas conservadoras patrullan los alrededores, buscando interceptar cualquier comunicación enemiga. El jugador encarna a un joven mensajero que, bajo la luz tenue del amanecer, debe abrirse paso sigilosamente para entregar un mensaje que podría evitar una emboscada.

Entorno visual y jugabilidad: El nivel presenta una *vista cenital fija*, donde los caminos y obstáculos se despliegan como un pequeño laberinto natural. El jugador controla al mensajero, moviéndose con cautela entre árboles, carpas y patrullas enemigas. Los enemigos lanzan *proyectiles con trayectorias parabólicas*, lo que obliga a anticipar los movimientos para esquivar el fuego.

Objetivo y reto principal: El objetivo es *entregar el mensaje sin ser visto ni atrapado*, alcanzando el punto de entrega antes de que el tiempo se agote. El nivel cuenta con un *temporizador visible*, que añade tensión y urgencia. La precisión en los desplazamientos, la lectura de los patrones enemigos y la gestión del tiempo son clave para superar esta primera misión.

Nivel 2: El Cruce del Río

Contexto: Tras cumplir su primera misión, el joven mensajero debe ahora cruzar un río crecido por las lluvias torrenciales. Las corrientes son fuertes y el cielo ruge con relámpagos; al otro lado lo espera un grupo aliado que necesita los suministros para resistir el avance enemigo. En esta travesía, la naturaleza se convierte en un enemigo tan implacable como la guerra misma.

Entorno visual y jugabilidad: El nivel se desarrolla con una *vista lateral fija*, que permite observar el movimiento del agua y la fuerza de la corriente. El jugador controla una pequeña *balsa de madera*, manejando su equilibrio mientras evita troncos, rocas y remolinos. La *física se basa en un movimiento oscilatorio*, que simula el vaivén de las olas, obligando a reaccionar con precisión para mantener la estabilidad.

Objetivo y reto principal: El objetivo es *cruzar el río sin perder los suministros*. El jugador debe mantener el equilibrio de la balsa, esquivar obstáculos flotantes y resistir las ráfagas del viento. Si la balsa se inclina demasiado o se golpea contra los obstáculos, se hundirá. La concentración, el control del movimiento y el ritmo de reacción son esenciales para sobrevivir al paso del río.

Nivel 3: La Batalla Final

Contexto: Después de días de lucha y sacrificio, el joven mensajero llega al frente de la última gran confrontación entre liberales y conservadores. El aire está cargado de pólvora y esperanza: este enfrentamiento definirá el destino de cientos de combatientes y marcará el final de la guerra. Ya no se trata solo de entregar mensajes; ahora, el protagonista debe liderar un pequeño grupo hacia la victoria, demostrando que su valor puede cambiar el rumbo de la historia.

Entorno visual y jugabilidad: El nivel utiliza una *vista cenital con desplazamiento lateral (scroll)*, que sigue el avance del grupo a través de trincheras, barricadas y zonas de combate. El jugador controla al protagonista y a su escuadrón, dirigiendo movimientos estratégicos y enfrentando a los enemigos en tiempo real. Las *físicas se basan en un movimiento lineal con aceleración*, tanto en el desplazamiento del personaje como en la trayectoria de los proyectiles, intensificando la acción a medida que avanza el combate.

Objetivo y reto principal: El objetivo es *tomar el control del territorio enemigo y derrotar al comandante conservador*. Para lograrlo, el jugador debe combinar estrategia, velocidad y precisión en el ataque. Cada decisión afecta al grupo, y el entorno —con zonas de cobertura y puntos de emboscada— exige pensar tácticamente. Al completar este nivel, el jugador no solo alcanza la victoria militar, sino también el cierre simbólico de una guerra que puso a prueba el valor, la lealtad y la esperanza.

3 Vistas, interacciones y físicas de cada nivel

En esta sección se describen de forma detallada las vistas, interacciones, físicas y elementos gráficos (sprites) correspondientes a cada uno de los tres niveles del videojuego. Esta información amplía y complementa la contextualización realizada en el Momento I, permitiendo establecer con mayor precisión las bases para su implementación en la capa lógica.

3.1 Nivel 1: El Mensajero del Campamento

Tipo de vista: Cenital fija. **Entorno:** Campamento liberal ubicado en una zona montañosa, con árboles, carpas y caminos angostos.

Interacciones:

- **Jugador-enemigos:** si el jugador ingresa en el rango visual de una patrulla conservadora, es detectado y perseguido.
- **Jugador-entorno:** los árboles y carpas bloquean el desplazamiento, generando rutas laberínticas.
- **Temporizador:** existe un tiempo límite para entregar el mensaje, aumentando la tensión y la dificultad.

Físicas utilizadas:

- Movimiento uniforme para el desplazamiento del jugador.
- Movimiento parabólico en los proyectiles enemigos.

Sprites propuestos: jugador (mensajero), enemigo (patrulla), árbol, carpa, proyectil, icono de reloj.

Dibujo nivel 1: Se muestra una vista superior del campamento con caminos, carpas y árboles distribuidos como obstáculos. El jugador debe recorrer el área evitando las patrullas hasta llegar al punto de entrega.



Figura 1: Vista general del Nivel 1.

3.2 Nivel 2: El Cruce del Río

Tipo de vista: Lateral fija. **Entorno:** Río crecido por la lluvia, con corriente intensa, truenos y obstáculos flotantes.

Interacciones:

- **Jugador–corriente:** la fuerza del agua afecta la estabilidad de la balsa.
- **Jugador–obstáculos:** las colisiones con troncos o rocas pueden causar pérdida de equilibrio o hundimiento.
- **Jugador–viento:** las ráfagas alteran la dirección del movimiento, obligando a reaccionar rápidamente.

Físicas utilizadas:

- Movimiento oscilatorio para simular el vaivén de la balsa.
- Movimiento armónico simple para representar el balanceo continuo causado por la corriente.

Sprites propuestos: balsa, tronco, roca, olas, lluvia, cielo tormentoso.

Dibujo nivel 2: El escenario se observa en vista lateral: el jugador controla una balsa que avanza por un río turbulento, evitando obstáculos y manteniendo el equilibrio hasta alcanzar la orilla contraria.

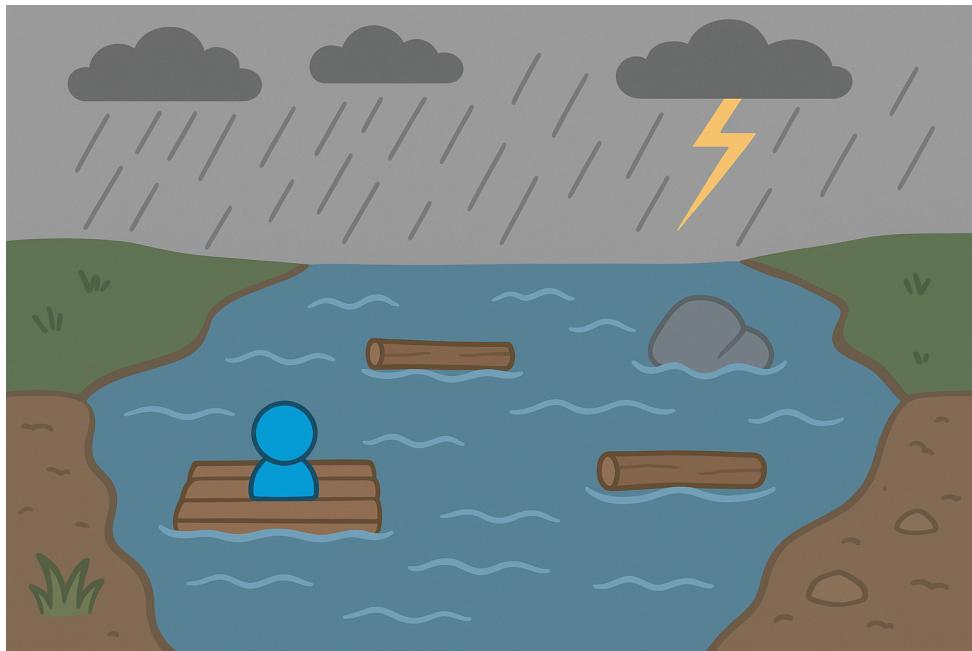


Figura 2: Vista general del Nivel 2.

3.3 Nivel 3: La Batalla Final

Tipo de vista: Cenital con desplazamiento lateral (scroll). **Entorno:** Campo de batalla con trincheras, barricadas y zonas de fuego cruzado.

Interacciones:

- **Jugador–enemigos:** combate directo mediante disparos; los enemigos poseen comportamiento autónomo.
- **Aliados–enemigos:** los aliados apoyan al jugador y pueden cubrirse o avanzar según la situación.
- **Entorno–personajes:** las barricadas y trincheras sirven como zonas de cobertura que reducen el daño recibido.

Físicas utilizadas:

- Movimiento lineal con aceleración para desplazamientos y proyectiles.
- Ecuaciones de colisión para determinar impactos entre disparos y personajes.

Sprites propuestos: jugador, aliado, enemigo, proyectil, trinchera, barricada, bandera de objetivo.

Dibujo nivel 3: La cámara sigue al protagonista y su escuadrón en una vista aérea con desplazamiento lateral. El jugador debe coordinar ataques y proteger a sus aliados hasta derrotar al comandante enemigo.

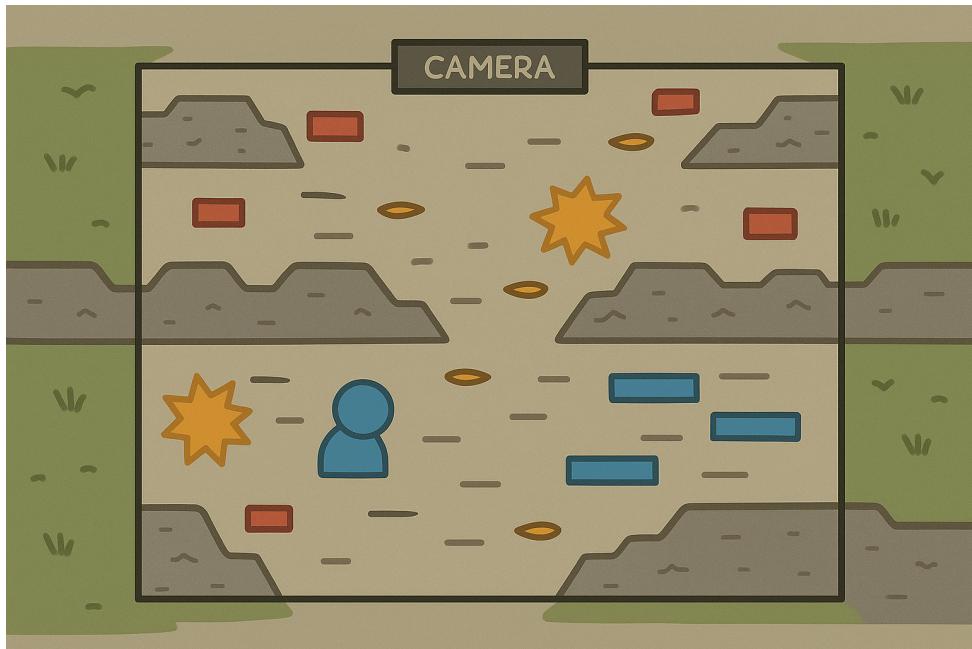


Figura 3: Vista general del Nivel 3.

4 Diagrama de clases de la capa lógica

En esta sección se presenta el diseño lógico del videojuego, estructurado mediante un diagrama de clases que define las relaciones, jerarquías y dependencias entre los componentes principales del sistema. El objetivo de este diseño es garantizar una arquitectura modular, reutilizable y coherente con la dinámica de los tres niveles definidos anteriormente.

El sistema se ha dividido en dos partes principales: la **lógica general del juego** y el **comportamiento inteligente de los enemigos**. A continuación, se detallan ambos diagramas y su función dentro del proyecto.

4.1 Estructura general del juego

La capa lógica está conformada por clases que representan los elementos centrales del videojuego:

- **GameManager:** coordina el flujo general del juego, la carga de niveles y el control del estado global.
- **Level:** representa cada escenario (campamento, río y campo de batalla). Contiene los elementos del entorno, los enemigos, el jugador y las físicas correspondientes.
- **Entity:** clase base abstracta que define propiedades comunes como posición, velocidad, energía y sprite. De esta derivan las clases **Player**, **Enemy** y **Ally**.
- **PhysicsModel:** clase base polimórfica encargada de representar distintos modelos de movimiento. De ella heredan **LinearMotion**, **ParabolicMotion** y **OscillatoryMotion**.
- **Projectile:** gestiona las trayectorias parabólicas de los disparos y su interacción con el entorno.

Estas clases se relacionan principalmente a través de composición y herencia: **GameManager** contiene un conjunto de **Level**, cada **Level** posee múltiples **Entity** (jugadores, enemigos y aliados), y cada **Entity** utiliza un modelo físico para definir su movimiento.

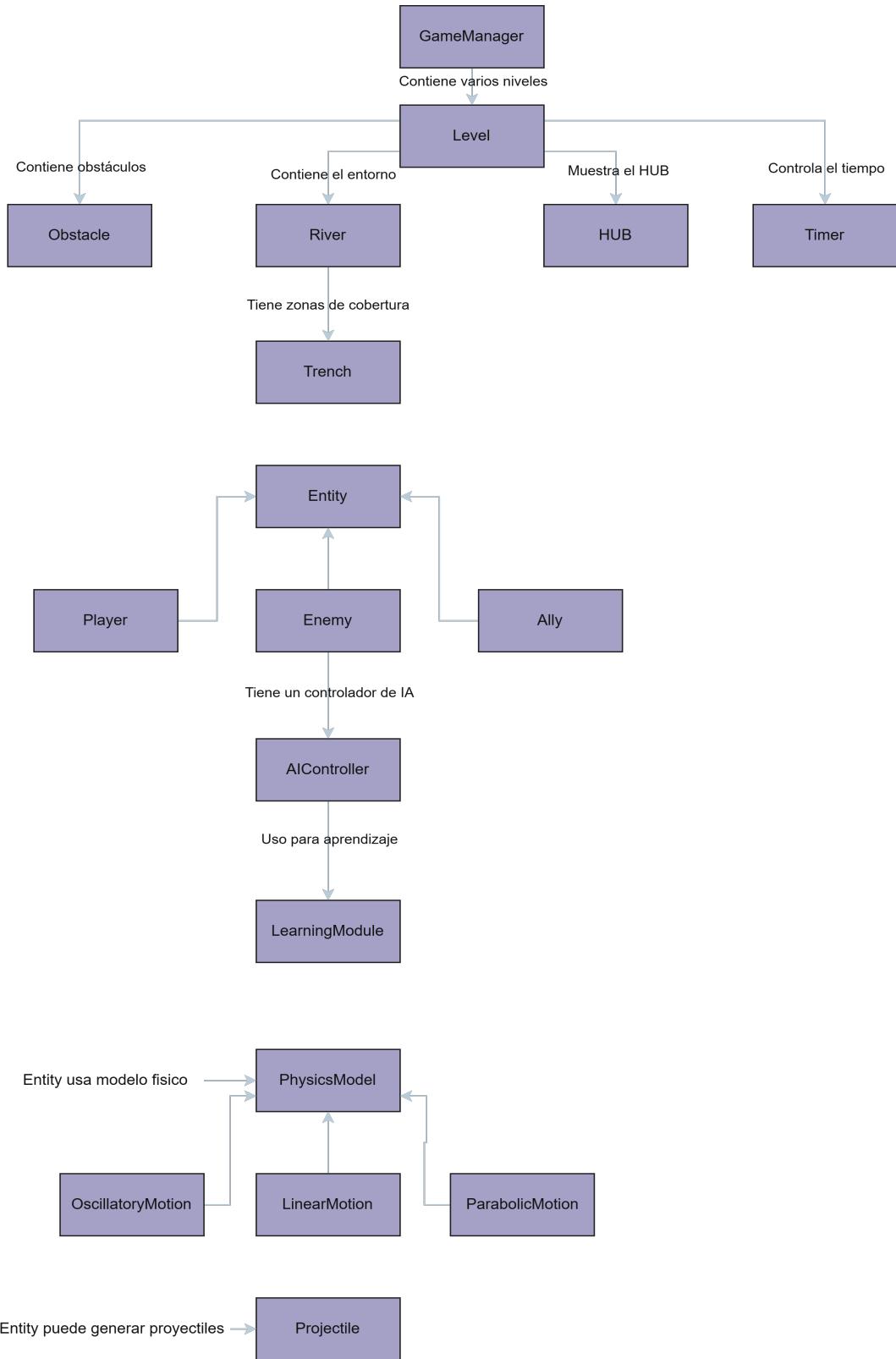


Figura 4: Diagrama de flujo lógico

5 Agente autónomo inteligente

En esta sección se profundiza en el diseño funcional del agente inteligente presentado en el diagrama de clases. Se describen sus componentes, procesos internos y la manera en que se integra dentro de la lógica general del videojuego.

5.1 Descripción en alto nivel de los componentes del agente

5.1.1. Percepción

El agente percibe al jugador cuando este ingresa en su rango de visión o produce ruido. Esta detección se basa en un radio de alcance configurable y en condiciones de visibilidad (por ejemplo, si hay obstáculos o cobertura entre ambos).

5.1.2. Razonamiento

Al percibir un estímulo, el agente evalúa su estado actual y determina la acción más conveniente:

- Si el jugador está cerca y en línea de visión → cambiar al estado **Atacar**.
- Si el jugador está lejos → cambiar al estado **Perseguir**.
- Si el jugador desaparece del rango → pasar a **Buscar cobertura**.

Este proceso se implementa mediante una máquina de estados finitos, donde las transiciones se activan según los eventos percibidos.

5.1.3. Acción

El agente ejecuta las decisiones tomadas, ya sea desplazarse, disparar o cubrirse. Durante estas acciones, se aplican las físicas correspondientes al entorno (movimiento lineal o parabólico para proyectiles) y se mantienen animaciones coherentes con cada comportamiento.

5.1.4. Aprendizaje

El módulo de aprendizaje (**LearningModule**) registra el historial de disparos. Cada vez que un proyectil falla, ajusta ligeramente el sesgo de puntería del enemigo. Así, con el tiempo, los enemigos mejoran su precisión sin perder equilibrio de dificultad en el juego.

5.2 Integración con el sistema del juego

El agente autónomo está vinculado directamente con la clase **Enemy**. Durante la ejecución del juego:

- El **AIController** interpreta los eventos y decide las acciones del enemigo.
- El **LearningModule** actualiza los parámetros de puntería y reacción.
- El **GameManager** se encarga de mantener sincronizadas las interacciones entre enemigos, jugador y entorno.



Figura 5: Diagrama de flujo del agente inteligente

6 Conclusión

Durante el desarrollo del Momento II se consolidó la estructura lógica y el diseño funcional del videojuego. Se definieron con claridad las vistas y dinámicas de los tres niveles, las físicas asociadas a cada entorno y la arquitectura de clases que permitirá su implementación modular en C++. Además, se diseñó el agente autónomo inteligente, cuya máquina de estados y módulo de aprendizaje ligero otorgan un comportamiento adaptativo a los enemigos, mejorando la inmersión y el desafío del jugador.

Este proceso de diseño sentó las bases técnicas y conceptuales para el Momento III, donde se integrarán las físicas parametrizables, los comportamientos autónomos y las vistas interactivas dentro de la implementación final en C++.

7 Inicio de la implementación (Momento III)

Para el desarrollo de la implementación final del videojuego se estableció una arquitectura modular basada en la separación entre la capa lógica del juego, la gestión de niveles y la escena gráfica. Durante esta primera fase del Momento III se construyeron las estructuras fundamentales sobre las cuales se implementarán las dinámicas, físicas y comportamientos inteligentes definidos en el Momento II.

7.1 Estructura base del sistema

Se crearon las clases que conforman la columna vertebral de la ejecución del videojuego:

- **GameScene:** derivada de `QGraphicsScene`, es responsable de administrar los elementos visuales en pantalla y del ciclo de actualización del juego mediante un temporizador interno.
- **Level (clase abstracta):** representa la estructura lógica común a todos los niveles. Define los métodos `load()` y `update()` que cada nivel deberá implementar. También gestiona la colección de entidades que pertenecen al escenario.
- **Level1, Level2 y Level3:** clases derivadas de `Level` que corresponden a los tres escenarios del videojuego. En esta etapa se implementó su estructura base, permitiendo su carga inicial sin lógica detallada todavía.
- **GameManager:** componente encargado de coordinar la carga, eliminación y transición entre niveles. Centraliza el flujo del videojuego y actúa como puente entre la interfaz principal y las escenas jugables.

7.2 Integración con la interfaz gráfica

La ventana principal del programa se ajustó para integrar de forma nativa la `GameScene` dentro de un `QGraphicsView`, permitiendo ejecutar y visualizar los niveles directamente desde la interfaz. Además, se incorporó el `GameManager`, encargado de cargar el Nivel 1 al inicio de la aplicación.

7.3 Estado actual de la implementación

En esta etapa se logró establecer:

- la estructura principal del sistema de niveles,
- la conexión funcional entre la interfaz gráfica y la lógica del juego,
- el ciclo de actualización listo para integrar físicas y comportamientos,
- los tres niveles creados a nivel estructural y listos para su programación específica.