Momento II

# Hugo Esteban Barrero García – Emmanuel López Ramírez

# Universidad de Antioquia

# Informática 2

# Aníbal José Guerra Soler – Augusto Salazar

# Medellín, Colombia

Junio, 2025

## **Descripción del Diagrama de Clases (en palabras)**

### **🔷 1. Clase Juego**

* **Responsabilidad: Es la clase coordinadora principal del juego.**
* **Contiene:**
  + **Instancias de los distintos Niveles (Nivel1, Nivel2, etc.).**
  + **Controla la transición entre niveles (menú → juego, derrota → reinicio, etc.).**
* **Relación:**
  + **Usa composición: contiene punteros a los niveles.**
  + **Utiliza señales y slots de Qt para cambiar de escena.**

### **🔷 2. Clase base Nivel**

* **Responsabilidad: Define la interfaz común para todos los niveles (cargar, reiniciar, detener música, etc.).**
* **Clase abstracta (tiene métodos virtual).**
* **Métodos clave: cargarNivel(), reiniciarNivel(), detenerMusica().**
* **Relación:**
  + **Heredada por Nivel1, Nivel2, y Nivel3.**
  + **Es QWidget, así que cada nivel es visualmente una pantalla propia.**

### **🔷 3. Clases Nivel1, Nivel2, Nivel3**

* **Heredan de: Nivel**
* **Cada uno tiene lógica independiente:**
  + **Nivel1: scroll lateral, selección de personaje, enemigos tipo soldado, sistema de corazones y proyectiles.**
  + **Nivel2: submarino, proyectiles, enemigos submarinos, minas, escape automático.**
  + **Nivel3: planeado como pelea 1vs1.**
* **Contienen:**
  + **Un Personaje controlado por el jugador (Roshi, Lunch o SubmarinoJugador).**
  + **Vectores de enemigos, proyectiles, objetos.**
  + **Timers (QTimer) para lógica del juego.**
  + **Música (QMediaPlayer) y audio (QAudioOutput).**
  + **Controles (QSet<int>), vidas, UI (corazones, diálogos, contador de enemigos).**
* **Relaciones:**
  + **Composición: contienen instancias de personajes, enemigos, proyectiles, etc.**
  + **Emite señales (volverAlMenu, reiniciar) hacia Juego.**

### **🔷 4. Clase base Entidad**

* **Responsabilidad: Cualquier cosa que se mueva y tenga posición en el juego.**
* **Atributos clave: posicion, velocidad, sprite, muerto, etc.**
* **Métodos clave: setPosicion(), getSprite(), aplicarFisica(), actualizar().**
* **Relaciones:**
  + **Heredada por:**
    - **Personaje (Roshi, Lunch, SubmarinoJugador)**
    - **Enemigo (SoldadoPatrullaRoja, SubmarinoEnemigo)**
    - **Proyectil y Torpedo**

### **🔷 5. Clase Personaje (abstracta)**

* **Heredada por:**
  + **Roshi y Lunch en Nivel1**
  + **SubmarinoJugador en Nivel2**
* **Responsabilidad: Moverse, disparar, recibir daño, mostrar animaciones.**
* **Relación:**
  + **Tiene métodos virtual como actualizarSprite(), atacar(), disparar().**
  + **Usa QTimer para animaciones.**
  + **Controlado por el jugador a través de keyPressEvent().**

### **🔷 6. Clases Roshi y Lunch**

* **Heredan de: Personaje**
* **Contienen:**
  + **Lógica de ataque (puño o disparo).**
  + **Animaciones de caminar, disparar, transformarse (Lunch).**
* **Relación:**
  + **Solo se usan en Nivel1.**

### **🔷 7. Clase SubmarinoJugador**

* **Heredada de: Personaje**
* **Responsabilidad: Movimiento fluido tipo inercia, disparar torpedos.**
* **Relación:**
  + **Solo se usa en Nivel2.**

### **🔷 8. Clase Enemigo (abstracta o base conceptual)**

* **Responsabilidad: Ser un enemigo controlado por IA.**
* **Heredada por:**
  + **SoldadoPatrullaRoja en Nivel1**
  + **SubmarinoEnemigo en Nivel2**
* **Métodos comunes: actualizar(), disparar(), recibirDanio(), iniciarMuerte()**
* **Contienen:**
  + **QProgressBar para la barra de vida.**
  + **IA básica para caminar o disparar al jugador.**
* **Relación:**
  + **Tienen acceso al Nivel donde están para generar proyectiles o leer estado del jugador.**

### **🔷 9. Clases SoldadoPatrullaRoja y SubmarinoEnemigo**

* **Especializan: Enemigo**
* **Responsabilidad:**
  + **IA de patrullaje, disparo, animación.**
  + **Disparo sincronizado con sprite.**
  + **Eliminación con efecto visual.**
* **Relación:**
  + **Se crean dentro del Nivel.**

### **🔷 10. Clase Proyectil**

* **Responsabilidad: Objeto móvil que puede dañar enemigos o al jugador.**
* **Atributos: posición, velocidad, daño, tipo ("bazuca", "subfusil", "energia").**
* **Relación:**
  + **Hereda de Entidad**
  + **Puede ser lanzado por jugador o enemigo (esDelJugador).**
  + **Tiene métodos para movimiento y colisión.**

### **🔷 11. Clase Torpedo (especialización de Proyectil)**

* **Responsabilidad: Disparo en el agua, usado en Nivel2.**
* **Relación:**
  + **Se mueve con velocidad constante.**
  + **Tiene animación y sonido opcional.**

### **🔷 12. Clase Corazon**

* **Hereda de: Objeto**
* **Responsabilidad: Ítem curativo.**
* **Atributos: sprite, temporizador para desaparecer, posición lógica.**
* **Relación:**
  + **Creado en Nivel1 cada 6 enemigos derrotados.**
  + **Curación gestionada por el nivel directamente (vidas++).**

### **🔷 13. Clase Objeto**

* **Responsabilidad: Base para ítems o elementos no hostiles.**
* **Métodos: moverSpriteConOffset(), getSprite().**
* **Heredada por: Corazon, Mina (si se implementa).**

### **🔷 14. Clase QTimer y otros de Qt**

* **QTimer: Usado para:**
  + **Animaciones (timerAnimacion)**
  + **Disparo automático (timerDisparo)**
  + **Bucle principal (timer en Nivel1, Nivel2)**
* **QMediaPlayer / QAudioOutput: Sonidos y música por nivel.**
* **QLabel: Todos los sprites visuales son QLabel con QPixmap.**
* **QVector2D: Movimiento, posición, desplazamiento.**
* **QSet<int>: Teclas presionadas.**

## **🔁 Relaciones entre clases**

* **Nivel ← heredan Nivel1, Nivel2, Nivel3**
* **Entidad ← heredan Personaje, Proyectil, Enemigo**
* **Personaje ← heredan Roshi, Lunch, SubmarinoJugador**
* **Enemigo ← heredan SoldadoPatrullaRoja, SubmarinoEnemigo**
* **Proyectil ← heredan Torpedo (si se especializa)**
* **Nivel contiene → Jugador, Enemigo\*, Proyectil\*, Corazon\***
* **Proyectil conoce si es del jugador o enemigo**
* **Enemigo tiene referencia a Nivel y a Jugador para IA**
* **Corazon interactúa con Nivel1 para curación**