



UNIVERSIDAD DE ANTIOQUIA  
FACULTA DE INGENIERÍA  
PROGRAMA DE INGENIERÍA DE TELECOMUNICACIONES  
SEMESTRE II

**Desafío 3**  
**Creación de videojuego en lenguaje C++**  
**Diagrama de Clases Bart contra Australia**

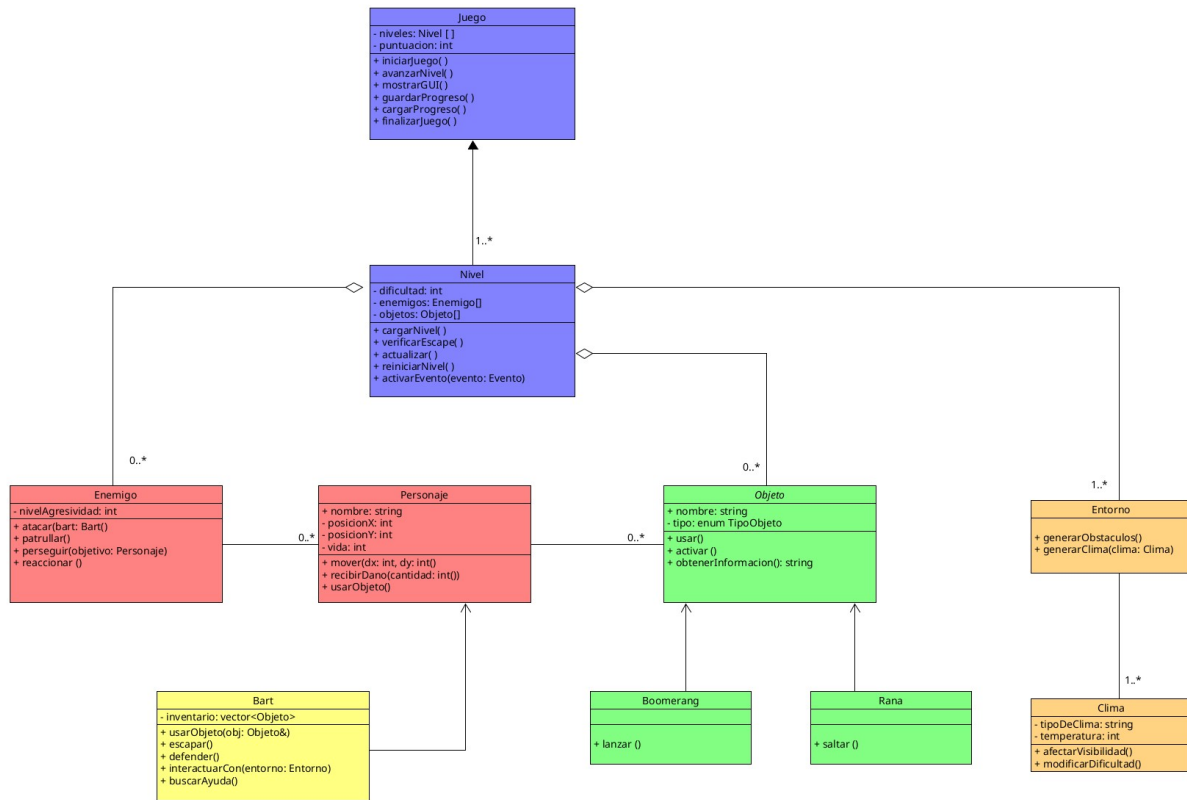
Asignatura:  
Informática II Teoría

Autores:  
Oscar Miguel López Peña  
Luis Carlos Romero Cardenas

Tutores:  
Anibal Guerra  
Augusto Salazar

Medellín, Octubre de 2024

## DIAGRAMA DE CLASES DESAFÍO III



*Nota: Elaborado en la aplicación UMLet V. 15.1*

### Análisis general del Diagrama de Clases

Este diagrama de clases está diseñado para modelar un juego donde el personaje principal, Bart, interactúa con enemigos, objetos y el entorno, enfrentándose a desafíos representados en diferentes niveles del juego. Cada clase cumple un rol específico en la estructura del juego, y las relaciones establecen cómo estas clases interactúan entre sí.

### Clases y explicación detallada

#### 1. Clase Juego

Descripción: Representa el juego en su totalidad, compuesto por múltiples niveles, y administra la puntuación y el progreso.

### **Atributos**

- + niveles: Nivel[] (Público). Un arreglo de niveles que componen el juego. Esto permite que los niveles sean accesibles desde otras partes del sistema.
- - puntuacion: int (Privado). La puntuación acumulada del jugador, protegida contra cambios externos.

### **Métodos**

- + iniciarJuego() (Público). Inicia el juego.
- + avanzarNivel() (Público). Avanza al siguiente nivel.
- + mostrarGUI() (Público). Muestra la interfaz gráfica del usuario.
- + guardarProgreso() (Público). Guarda el progreso del juego.
- + cargarProgreso() (Público). Carga un progreso guardado.
- + finalizarJuego() (Público). Termina el juego y muestra el puntaje final.

## **2. Clase Nivel**

Descripción: Cada instancia representa un nivel individual del juego, con su dificultad, enemigos y objetos.

### **Atributos**

- - dificultad: int (Privado). La dificultad específica del nivel, protegida para evitar modificaciones no autorizadas.
- + enemigos: Enemigo[] (Público). Un arreglo de enemigos en el nivel, permitiendo el acceso a los enemigos desde fuera de la clase.
- + objetos: Objeto[] (Público). Un arreglo de objetos en el nivel, accesible para su uso en otros módulos del juego.

### **Métodos**

- + cargarNivel() - (Público). Carga todos los elementos del nivel (enemigos, objetos, entorno).
- - verificarEscape() (Privado). Verifica internamente si el jugador ha escapado del nivel.
- + actualizar() (Público). Actualiza el estado del nivel.
- + reiniciarNivel() (Público). Reinicia el nivel actual.

- + activarEvento(evento: Evento) (Público). Activa un evento específico dentro del nivel.

### 3. *Clase Enemigo*

Descripción: Representa a los enemigos que Bart enfrenta en cada nivel.

#### Atributos

- - nivelAgresividad: int (Privado). Define el nivel de agresividad del enemigo, controlado dentro de la clase.

#### Métodos

- + atacar(bart: Bart) (Público). Permite al enemigo atacar a Bart.
- - patrullar() (Privado). Define el comportamiento de patrullaje del enemigo, sin necesidad de acceso externo.
- + perseguir(objetivo: Personaje) (Público). Permite que el enemigo persiga a un personaje específico.
- - reaccionar() (Privado). Reacciona a estímulos en el entorno, manejado internamente.

### 4. *Clase Objeto*

Descripción: Representa los objetos con los que Bart y otros personajes pueden interactuar.

#### Atributos

- - nombre: string (Privado). El nombre del objeto, controlado internamente.
- - tipo: enum TipoObjeto (Privado). Tipo de objeto (por ejemplo, Boomerang, Rana).

#### Métodos

- + usar() (Público). Permite utilizar el objeto.
- - activar() (Privado). Activa una función especial del objeto en situaciones específicas.
- + obtenerInformacion(): string (Público). Retorna información relevante sobre el objeto.

## 5. *Clase Bart*

Descripción: Representa al personaje principal del juego, controlado por el jugador.

### Atributos

- - inventario: vector<Objeto> (Privado). Un inventario de objetos que Bart puede llevar y usar, accesible solo dentro de la clase.

### Métodos

- + usarObjeto(obj: Objeto&) (Público). Permite a Bart usar un objeto específico de su inventario.
- + escapar() (Público). Bart intenta escapar de una situación peligrosa.
- + defender() (Público). Bart se defiende de un ataque.
- + interactuarCon(entorno: Entorno) (Público). Permite que Bart interactúe con el entorno.
- - buscarAyuda() (Privado). Representa una acción interna de Bart en busca de ayuda.

## 6. *Clase Personaje*

Descripción: Clase base para todos los personajes en el juego, de la cual heredan Bart y Enemigo.

### Atributos

- + nombre: string (Público). Nombre del personaje.
- + posicionX: int y + posicionY: int (Público). Coordenadas que representan la posición del personaje en el nivel.
- - vida: int (Privado). Puntos de vida, protegidos contra modificaciones externas.

### Métodos

- + mover(dx: int, dy: int) (Público). Cambia la posición del personaje en el nivel.
- + recibirDano(cantidad: int) (Público). Disminuye la vida del personaje.
- + usarObjeto() (Público). Permite al personaje utilizar un objeto.

## 7. *Clase Entorno*

Descripción: Representa el entorno en el que se desarrolla el nivel, permitiendo la generación de obstáculos y cambios de clima.

### **Métodos**

- + generarObstaculos() (Público). Genera obstáculos en el entorno.
- + generarClima(clima: Clima) (Público). Configura las condiciones climáticas.

## 8. *Clase Clima*

Descripción: Clase que describe el clima en el juego, afectando potencialmente la visibilidad y dificultad.

### **Atributos**

- + tipoDeClima: string (Público). Describe el tipo de clima (por ejemplo, soleado, lluvioso).
- + temperatura: int (Público). Define la temperatura actual en el entorno.

### **Métodos**

- + afectarVisibilidad() (Público). Reduce la visibilidad en función del clima.
- + modificarDificultad() (Público). Ajusta la dificultad del nivel dependiendo del clima.

## **Relaciones y multiplicidades**

### ***Juego - Nivel***

- Relación: Composición
- Multiplicidad: 1..\* desde Juego hacia Nivel.
- Explicación: Un juego está compuesto por uno o más niveles.

### ***Nivel - Enemigo y Objeto***

- Relación: Agregación
- Multiplicidad: 0..\* desde Nivel hacia Enemigo y Objeto.

- Explicación: Un nivel contiene múltiples enemigos y objetos que existen independientemente.

### ***Enemigo - Personaje***

- Relación: Asociación
- Multiplicidad: 0..\* desde Enemigo hacia Personaje.
- Explicación: Un enemigo puede atacar o interactuar con un personaje.

### ***Bart - Objeto (Inventario)***

- Relación: Agregación
- Multiplicidad: 0..\* desde Bart hacia Objeto.
- Explicación: Bart puede tener múltiples objetos en su inventario.

### ***Entorno - Clima:***

- Relación: Asociación
- Multiplicidad: 1..1 entre Entorno y Clima.
- Explicación: Un entorno tiene un clima que define sus condiciones.

## **Relaciones de herencia**

### ***Herencia de Personaje***

Bart y Enemigo heredan de Personaje. Esto significa que tanto Bart como los enemigos tienen características comunes definidas en Personaje (como nombre, posicionX, posicionY, y vida), y también pueden ejecutar métodos como mover() y recibirDano().

Multiplicidad: La herencia no requiere multiplicidad. Cada subclase hereda de la superclase en una relación 1..1, donde Bart y Enemigo representan tipos específicos de Personaje.

### ***Herencia de Objeto***

Boomerang y Rana heredan de Objeto. Esto indica que ambos son tipos específicos de Objeto, con funcionalidades propias como lanzar() y saltar(), respectivamente.

Multiplicidad: La herencia no requiere multiplicidad. Cada subclase de Objeto (Boomerang y Rana) tiene una relación 1..1 con Objeto, representando variaciones especializadas de esta clase.

### **Resumen**

Este diagrama de clases está diseñado para capturar la lógica del juego "*Bart contra Australia*". En este contexto:

Juego controla los niveles y la puntuación del jugador, mientras que cada Nivel representa un espacio en el que Bart se enfrenta a enemigos y encuentra objetos.

Bart es el personaje principal y puede usar varios Objetos que recoge a lo largo de los niveles. Su inventario es una agregación de objetos, lo que le permite recolectar y utilizar elementos sin afectar directamente su existencia.

Enemigo es una clase especializada de Personaje y actúa de manera independiente en el nivel, con comportamientos como patrullar, perseguir y atacar, especialmente dirigidos hacia Bart.

Entorno y Clima proporcionan contexto ambiental, con Entorno estableciendo obstáculos y Clima afectando la dificultad y visibilidad en el nivel, lo cual añade profundidad a la jugabilidad y presenta desafíos adicionales para el jugador.

La herencia de Personaje a Bart y Enemigo, y de Objeto a Boomerang y Rana, ayuda a definir comportamientos y características específicos según el rol de cada clase en el juego.