

# Diagrama de Clases del proyecto de software

José Ignacio Botero Osorio

**Instructor**

Lyam Acosta Forero

**Aprendiz**

SENA – Servicio Nacional de Aprendizaje

Tecnología en análisis y Desarrollo de Software

3118526

Bogotá D.C.

## **Contenido**

<b>Diagrama de Clases del proyecto de software .....</b>	<b>3</b>
<b>Diagrama de clases para nuestro proyecto.....</b>	<b>3</b>
<b>Buenas prácticas de diseño orientado a objetos aplicados .....</b>	<b>3</b>
<b>Diagrama.....</b>	<b>4</b>

# Diagrama de Clases del proyecto de software

En nuestro proyecto elaboraremos un diagrama de clases aplicando buenas prácticas de diseño orientado a objetos, es importante seguir unos pasos claves y considerar ciertos elementos estructuras y diseños.

## Diagrama de clases para nuestro proyecto

El diagrama de clases define la estructura del sistema, mostrando clases con sus atributos y métodos, así como la relación entre ellas tales como herencias, asociaciones y otros.

En nuestro videojuego las clases que incluiremos son:

1. **Jugador (Player):** Representa al usuario o personaje principal con atributos como salud, posición, puntaje y métodos para moverse o atacar.
2. **Enemigos (Enemy):** Diferentes tipos de enemigos. Con atributos y acciones propias.
3. **PowerUps o ítems:** Las clases que representan objetos que el jugador puede recoger.
4. **Nivel (Level):** Controlamos el escenario, enemigos y condiciones de finalización.
5. **Controlador del juego (GameController):** Maneja la lógica general y el estado del juego.

## Buenas prácticas de diseño orientado a objetos aplicados

1. Encapsulamiento con atributos privados o protegidos con método público de acceso.
2. Uso de herencias para reutilización, por ejemplo, la clase Enemy como base para tipos específicos de enemigo.
3. Polimorfismo para manejar diferentes comportamientos a través e interfaces o métodos sobrecargados.
4. Aplicación de patrones de diseño como Factory para crear enemigos o ítems, y Observer para eventos del juego.
5. Nombres claros y constantes para clases, atributos o métodos.
6. Separación lógica de responsabilidad según el principio de responsabilidad única.

Estas prácticas aseguran que el diagrama refleje un diseño sólido, mantenible y escalable para nuestro videojuego.

## Diagrama

A continuación, revisaremos el diagrama visual:

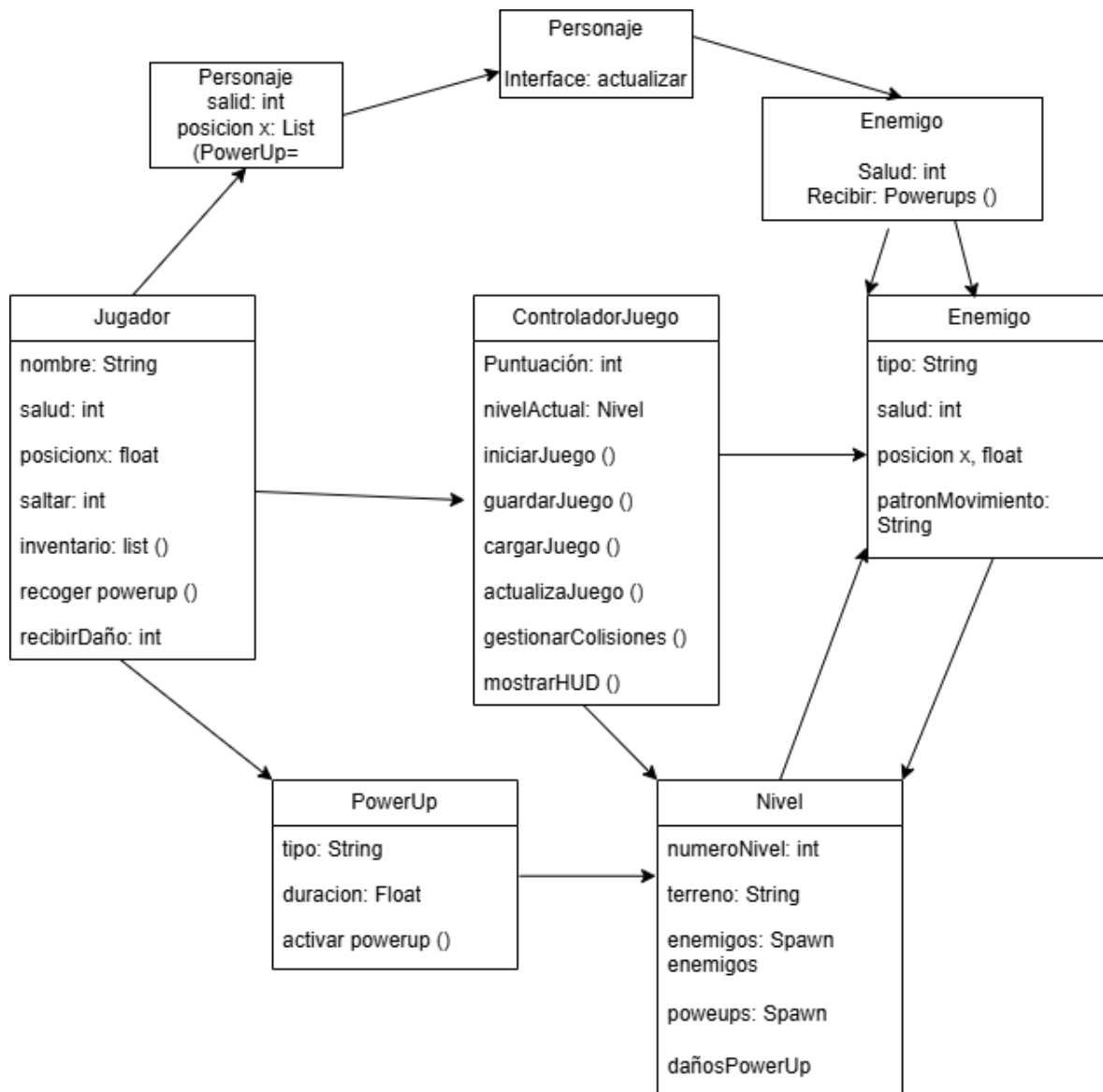


Ilustración Diagrama de Clases