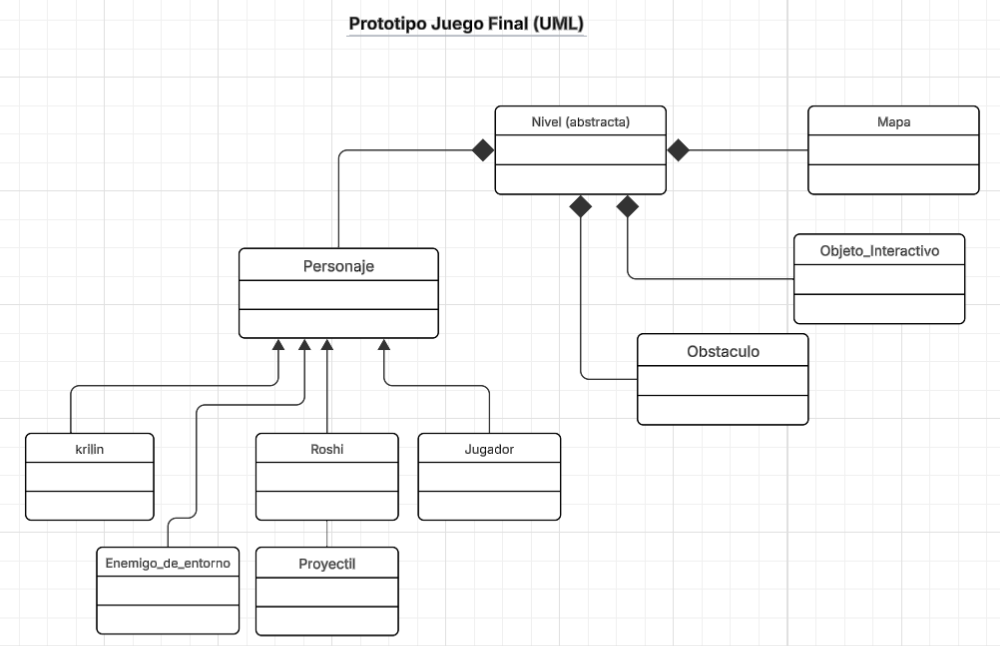
# **Momento II: Diseño y Análisis**

## **1. Diagrama de Clases – Capa Lógica**

El diseño del videojuego se basa en principios de Programación Orientada a Objetos (POO) con herencia, uso de memoria dinámica. A continuación se presenta el diagrama de clases con nivel de detalle medio:

### **Clases principales:**

* Mapa
* Personaje (superclase)  
  + Krilin, Roshi
  + Jugador
  + Enemigo\_de\_entorno
* Objeto\_Interactivo
* Proyectil
* Obstaculo
* Nivel

## **2. Cambios y Detalles Ampliados desde el Momento I**

Se han definido **tres niveles** distintos, con mecánicas y físicas completamente únicas. Además, se ha formalizado la estructura de clases, incorporando la reutilización de código con herencia para personajes y objetos.

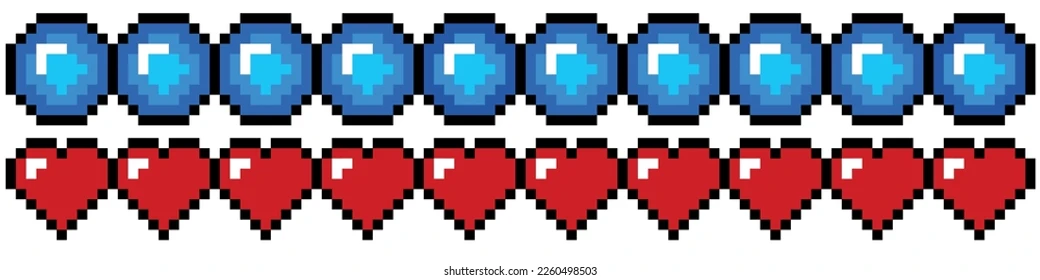
## **3. Sprites por nivel (mínimo 4 por nivel)**

**Nivel 1 – Carrera con Leche**

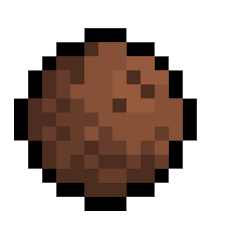
* Sprite de Goku corriendo.
* Sprite de troncos/huecos/rocas como obstáculos.  
  
* Sprite de fondo tipo montaña/terreno agreste.
* Sprite del contenedor de leche.



**Nivel 2 – Lago con Piedra**

* Sprite de Goku nadando.
* Sprite de peces enemigos.  
  
* Sprite de la piedra del Maestro Roshi.
* Sprite del medidor de oxígeno.  
  

**Nivel 3 – Esquiva y Contraataque**

* Sprite de Goku en postura de pelea.  
  
* Sprite de Roshi lanzando objetos.
* Sprite de proyectiles (chanclas, cocos, piedras).
* Sprite del botón o animación de contraataque.

## 

## 

## 

## **4. Descripción de Vistas, Interacciones y Físicas**

### **🟦 Nivel 1 – *Carrera con Leche***

* **Vista:** scroll horizontal, Goku avanza automáticamente.
* **Interacción:** el jugador controla los saltos y esquives con teclado.
* **Físicas:**
  + Movimiento parabólico para los saltos.
  + Caída libre si no se salta a tiempo.
  + Colisión con obstáculos restan vidas (máximo 3).

### **🟦 Nivel 2 – *Lago con la Piedra del Maestro Roshi***

* **Vista:** entorno acuático en scroll vertical y horizontal limitado.
* **Interacción:** jugador controla el nado y dirección de Goku.
* **Físicas:**
  + Movimiento oscilatorio (seno/coseno) para simular el nado.
  + Fuerza de arrastre (resistencia del agua).
  + Movimiento de peces con trayectorias circulares.

### **🟦 Nivel 3 – *Esquiva y Contraataque***

* **Vista:** Goku en el centro, Roshi al fondo lanzando objetos.
* **Interacción:** el jugador esquiva y contraataca con precisión de tiempo.
* **Físicas:**
  + Proyectiles en trayectoria parabólica.
  + Colisión detectada con hitbox de Goku.
  + Si el jugador esquiva justo a tiempo, puede contraatacar.

## **5. Consideraciones Técnicas Generales**

* **POO y memoria dinámica:** Clases como Jugador, Personaje, ObjetoInteractivo y Proyectil usan punteros y contenedores dinámicos.
* **Herencia:** Clase base Personaje, de la cual heredan Krilin, Roshi, Jugador y Enemigo\_de\_entorno.
* **Contenedores:** Uso de QList y QVector para gestionar múltiples enemigos, objetos y proyectiles.
* **GUI:** Qt maneja la interfaz gráfica, eventos de teclado y temporizadores.
* **Dificultad progresiva:** Cada nivel aumenta en complejidad, tiempo de reacción y variedad de obstáculos.
* **Manejo de excepciones:** En colisiones, estados fuera de pantalla, errores de punteros, etc.