

UNIVERSIDAD DE GUAYAQUIL

MATERIA: PROCESO DE SOFTWARE

GRUPO 7

PROYECTO CON MÉTODO DE CASCADA -
DESARROLLO DE UN VIDEOJUEGO
EDUCATIVO CON TEMÁTICA ECUAT

INTEGRANTES

Silva Delgado elkin jared

Jimenez Castro Jose Luis

Rodas Gómez Julio

Jostin Daniel Suárez Martinez

Año 2025

Lectivo 2026

➤ *Análisis de requisitos*

Descripción del propósito educativo.

Primeramente, se intenta acercar a los jugadores con el contexto de el acontecimiento de la Batalla de pichincha, segundamente desarrollar la resolución de problemas u obstáculo, y la coordinación Visio-manual presentados en el videojuego

Público objetivo (nivel educativo, edad).

Nivel educativo: Educación básica en adelante

Edad: 8 – 40 años

Requisitos funcionales y no funcionales.

- *Requisitos funcionales*
- Correr
- Saltar
- Detectar colisiones (plataformas, checkpoint, cofres, enemigos, trampas, y final)
- Puntaje
- Guardar ubicación (checkpoint)
- Reproducir sonidos de saltos o colisión
- Cambiar de nivel

Requisitos No Funcionales

- Diseño fácil y comprensible
- Rendimiento fluido (30 fps)
- Controles sencillos
- Adaptabilidad a equipos

Identificación de actores y escenarios.

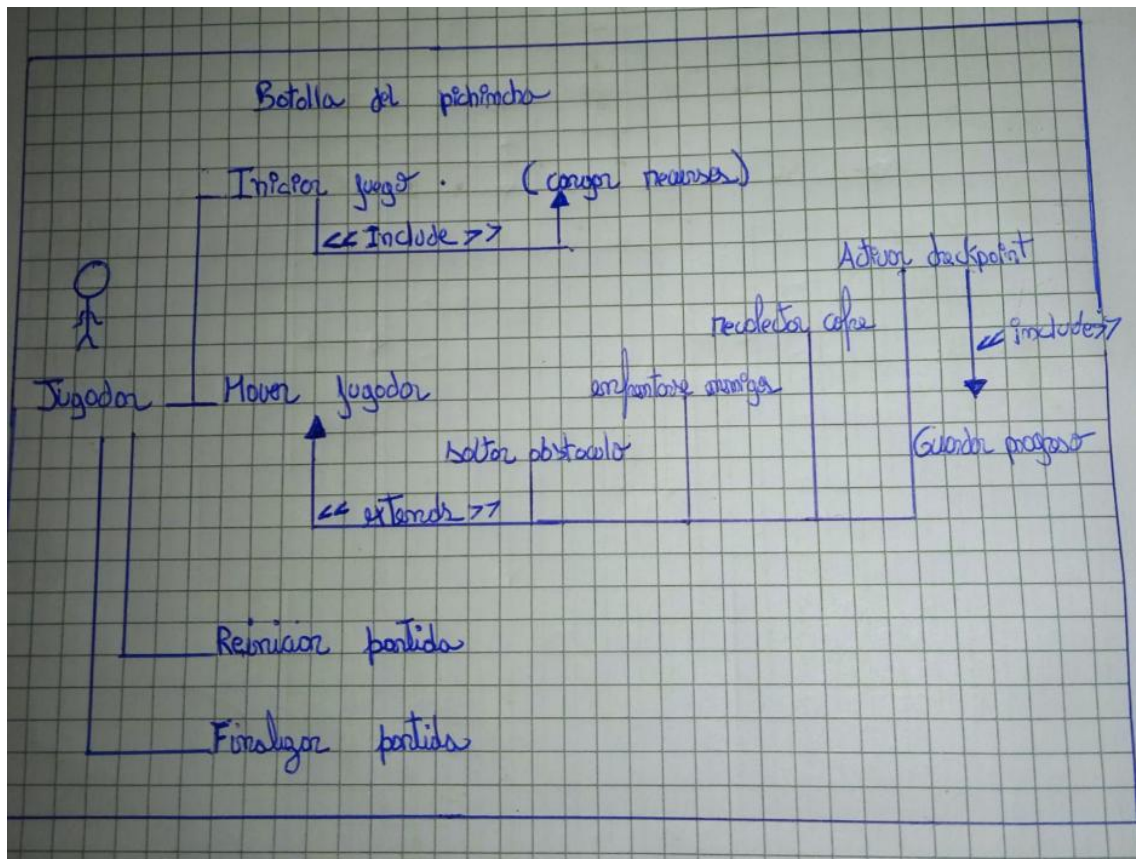
Actores

Player, Cofre, checkpoint, enemigos, trampas, objetivo final

Escenario

1 inicio del nivel, 2 Jugador recoge cofre o mata a un enemigo (aumenta el score), 3 El jugador muera por las trampas o enemigos, 4 Llega al checkpoint y se guarda la ultima ubicación, 5 Se llega al final del nivel para pasar al siguiente

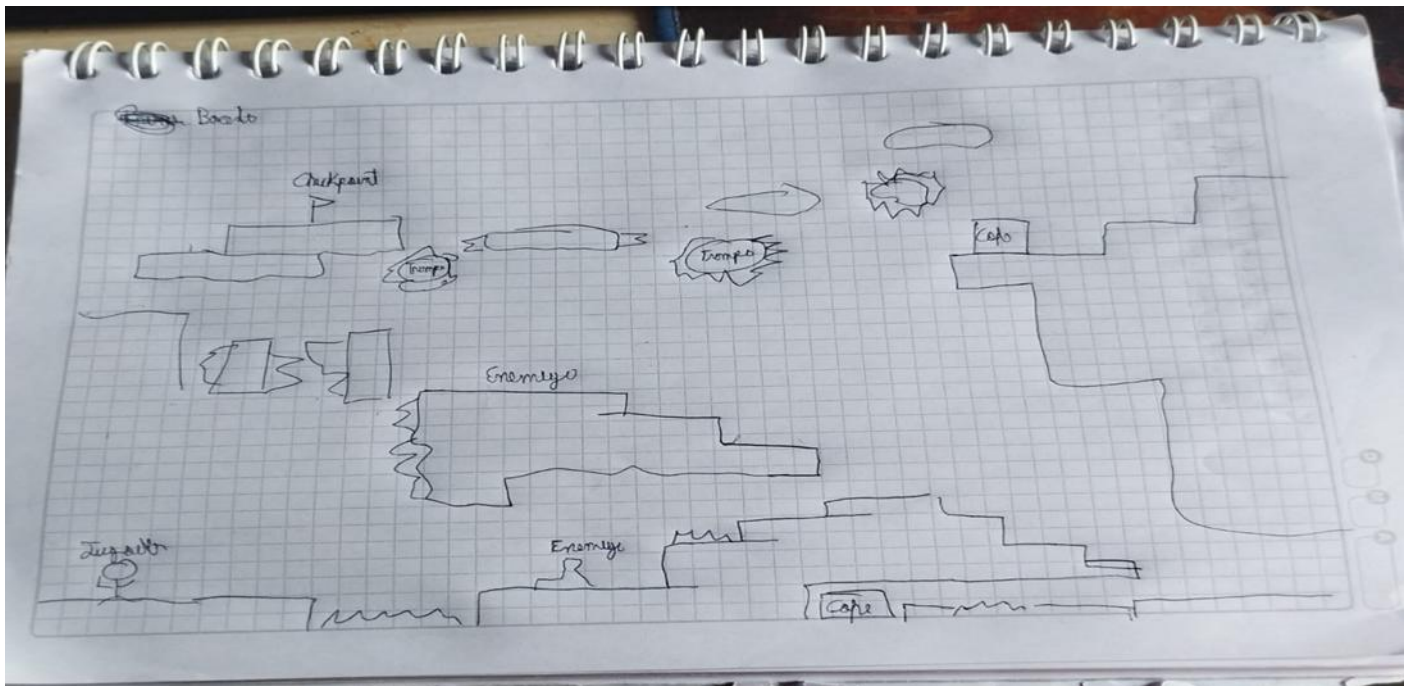
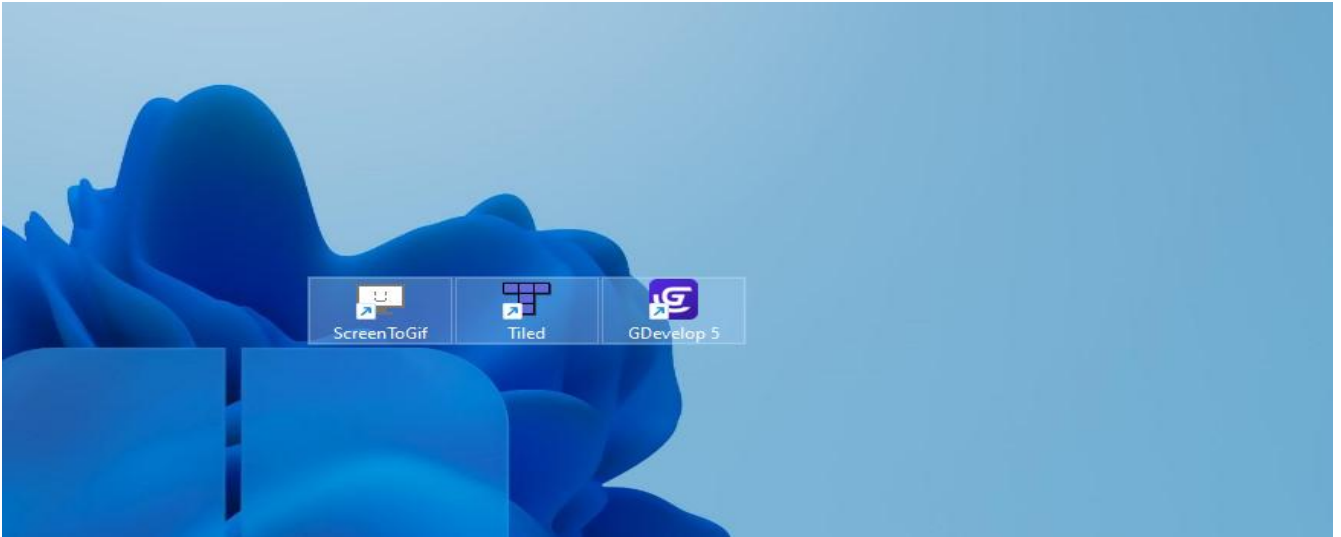
•Diagrama de casos de uso.



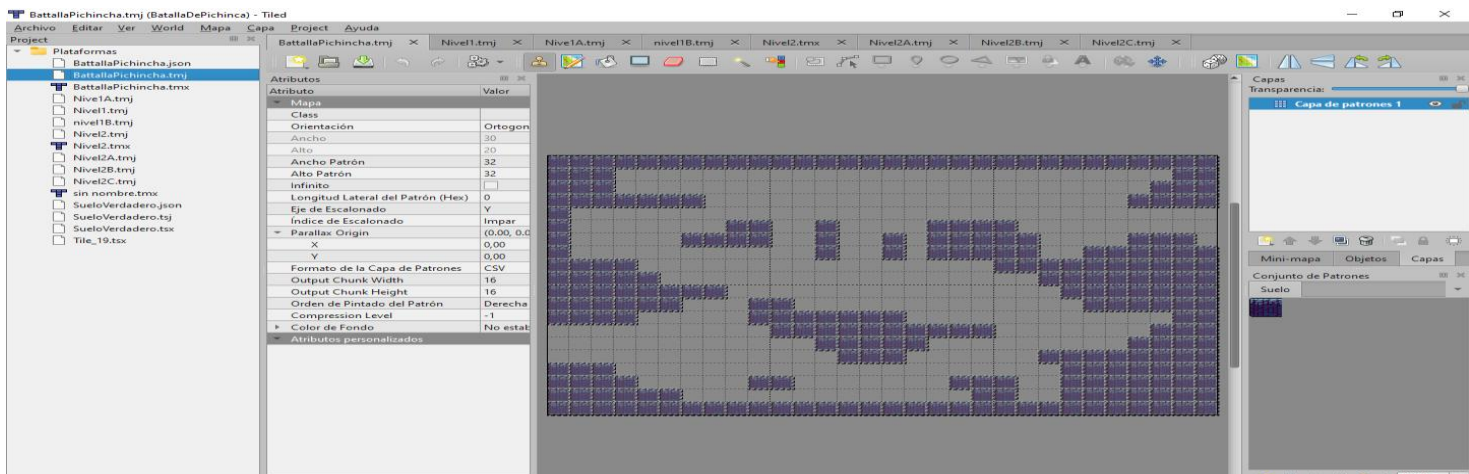
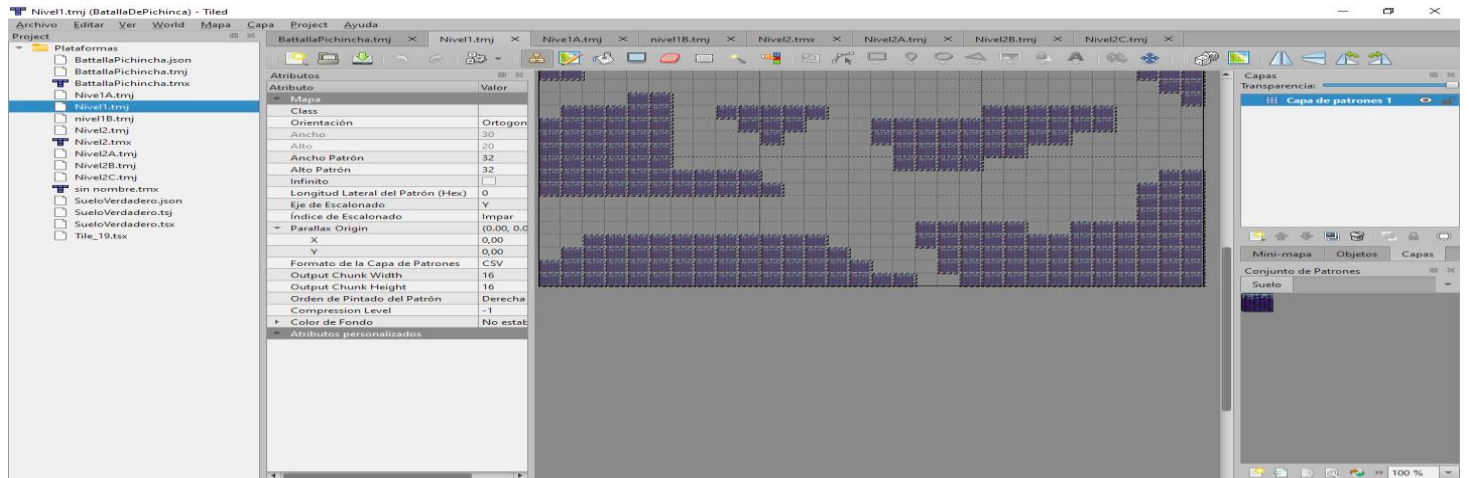
➤ Diseño

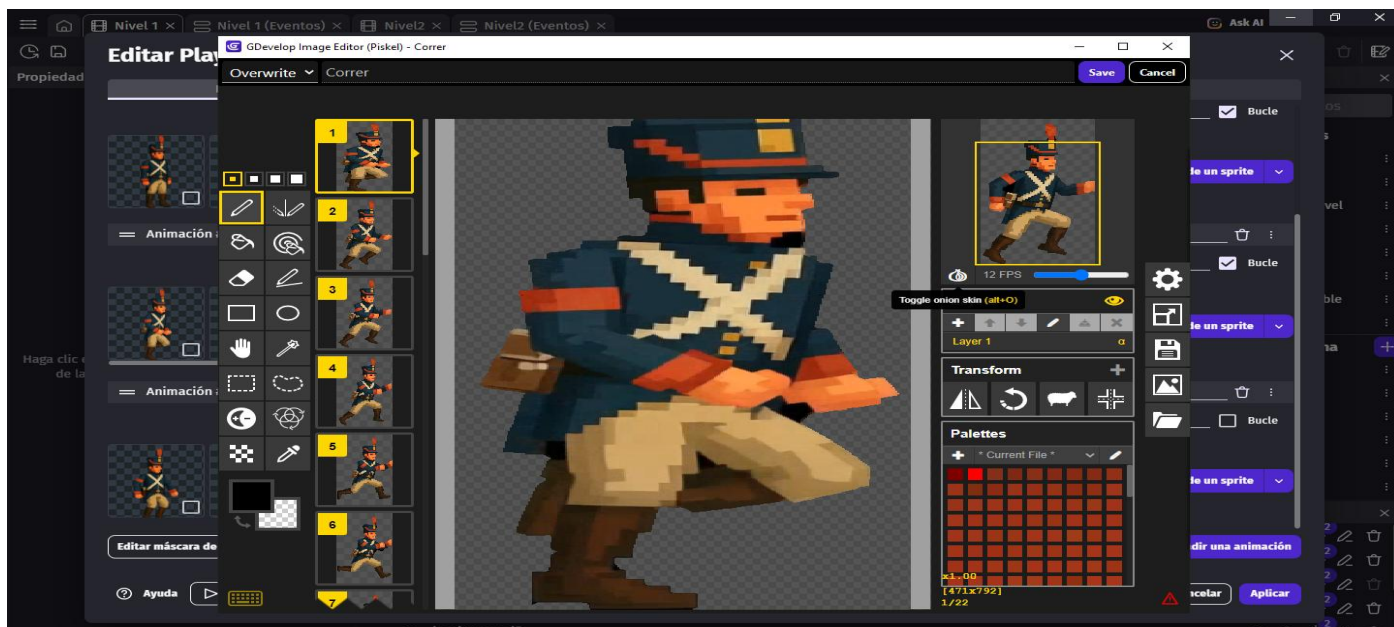
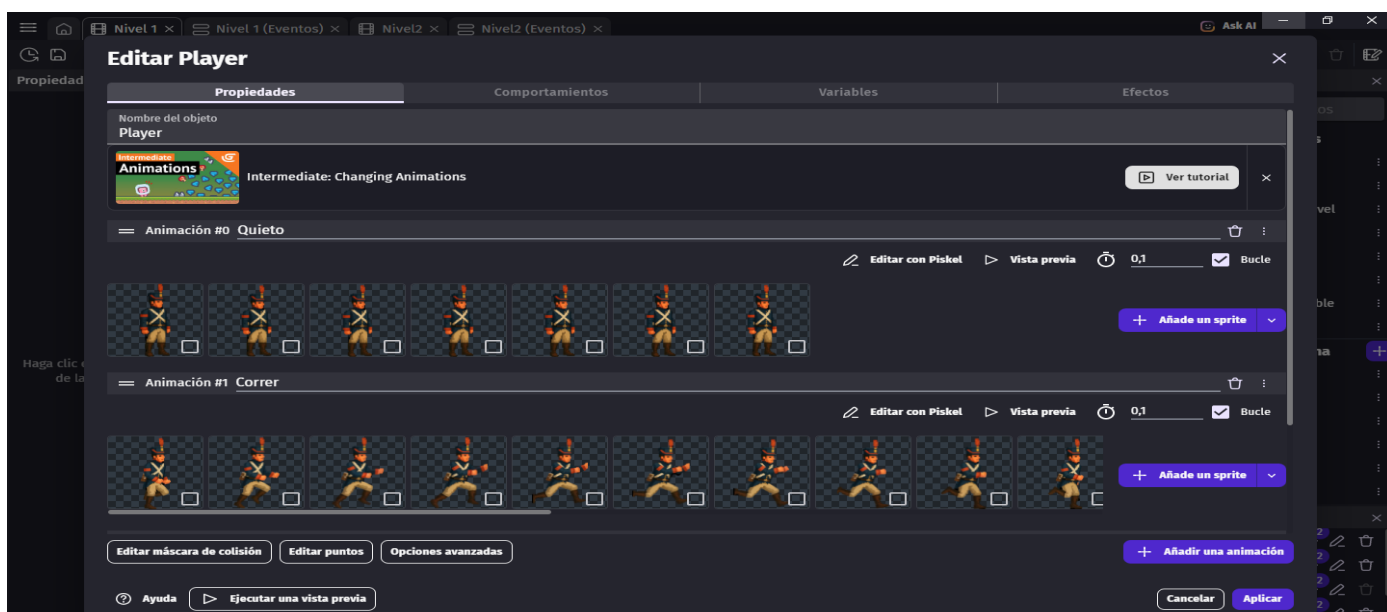
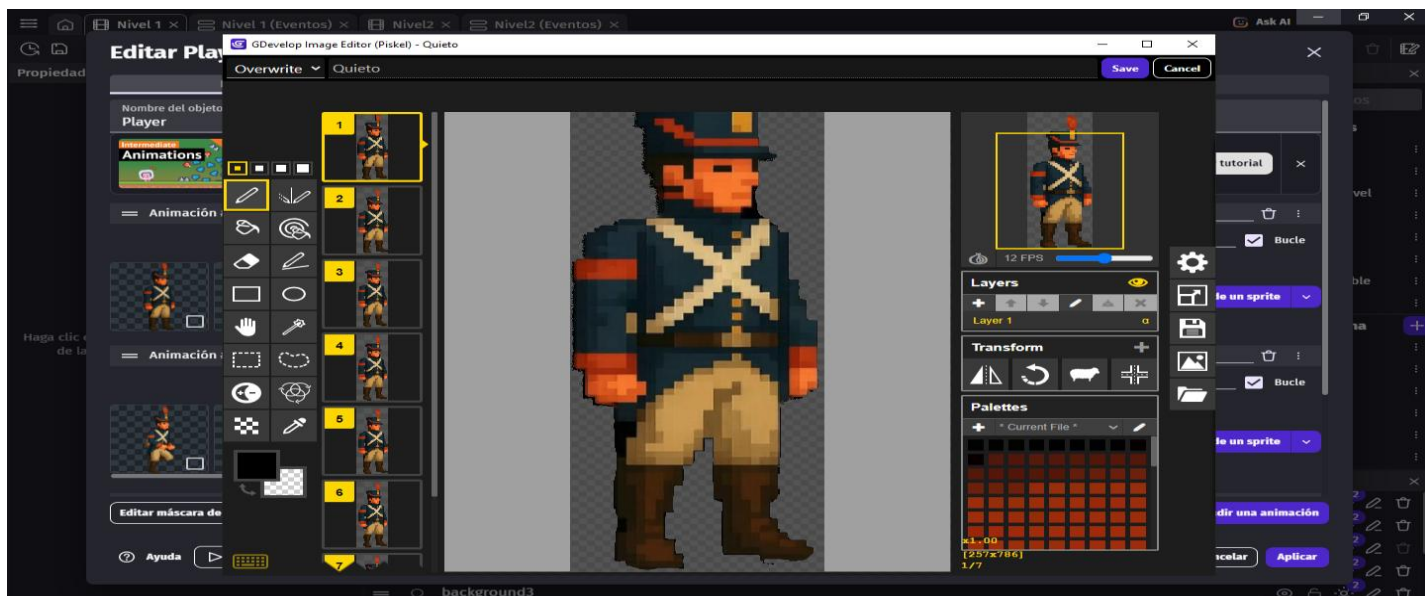
- **Bocetos o wireframes de pantallas.**

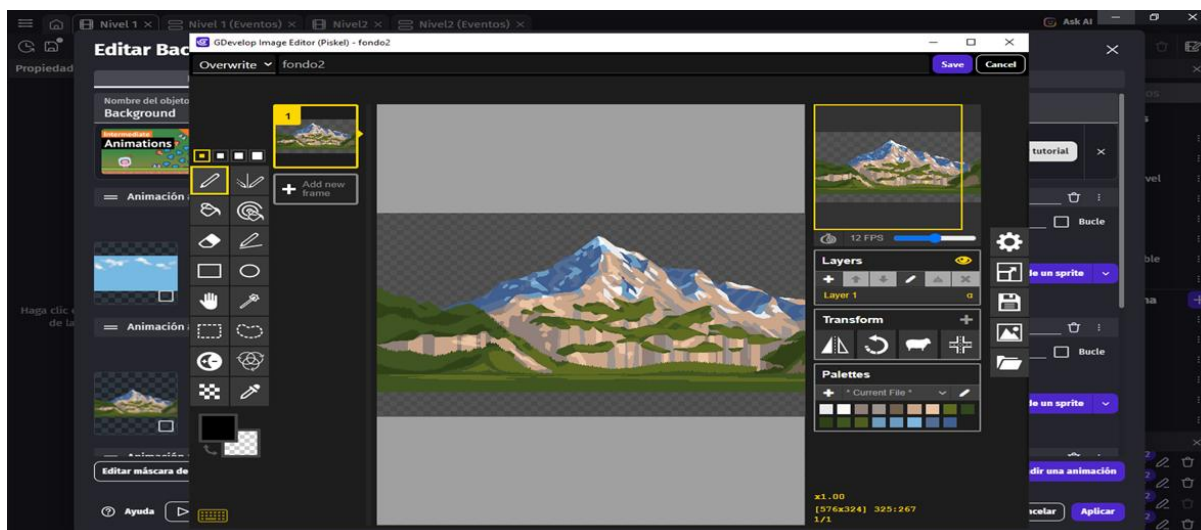
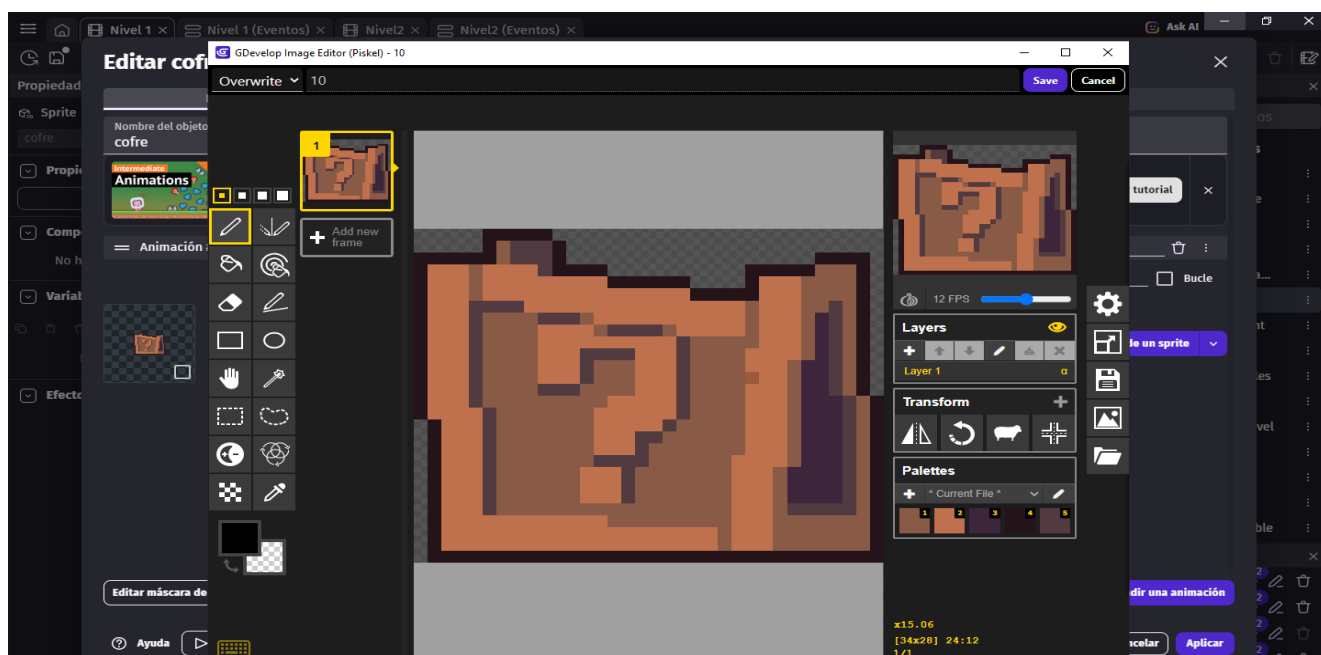
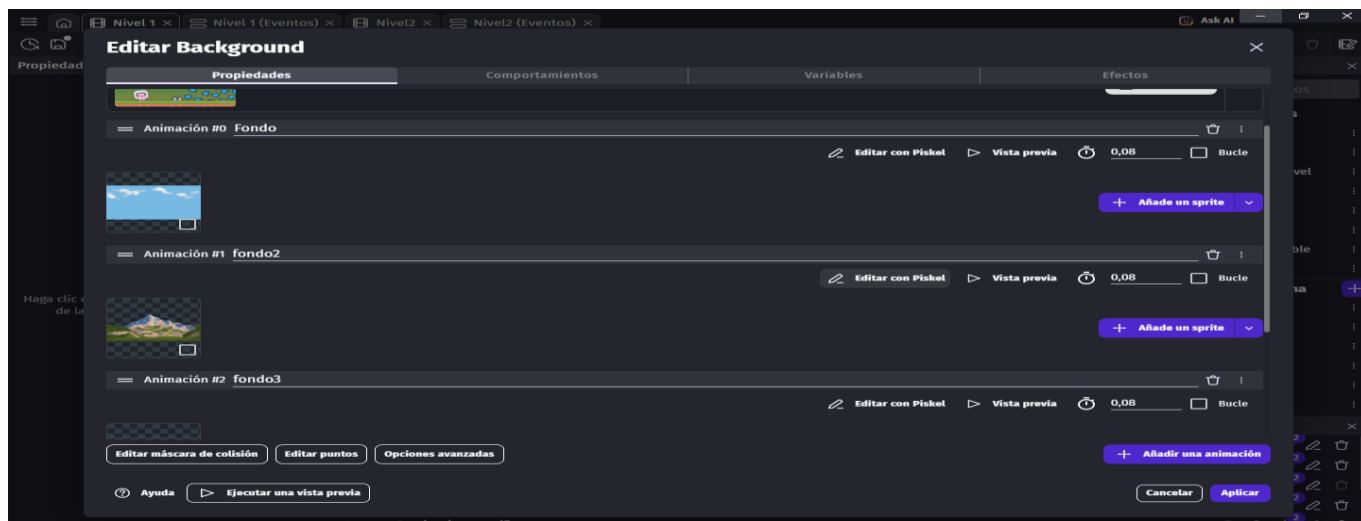
- Bocetos o wireframes de pantalla



Diseño de personajes, fondos y recursos visuales.







Arquitectura del videojuego (diagrama de módulos).

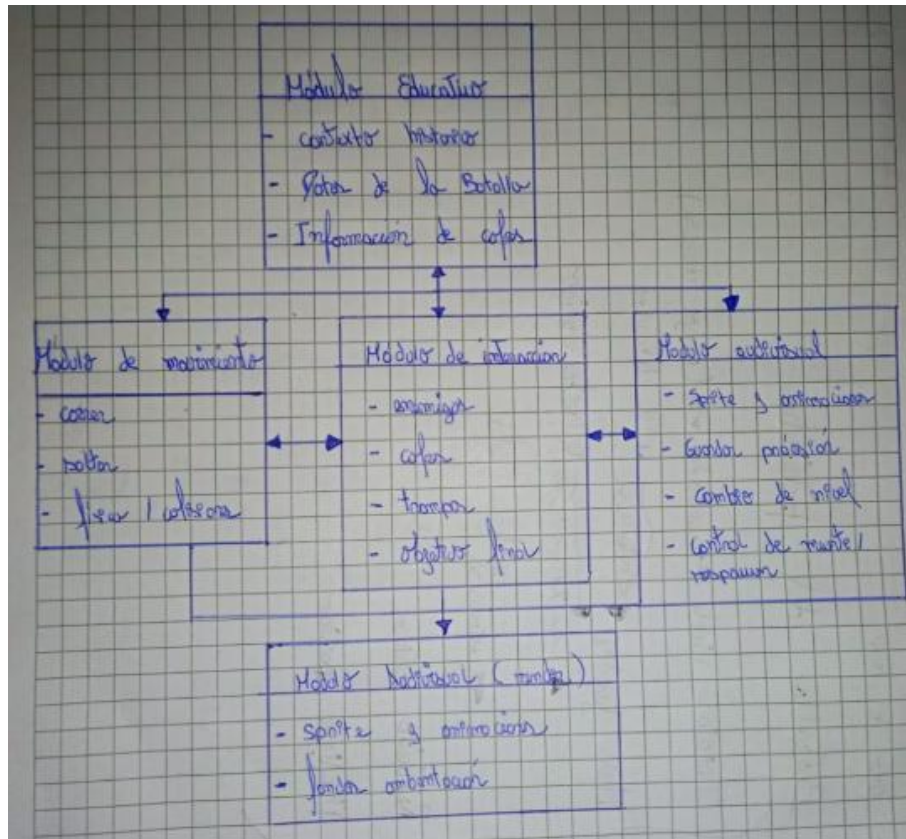
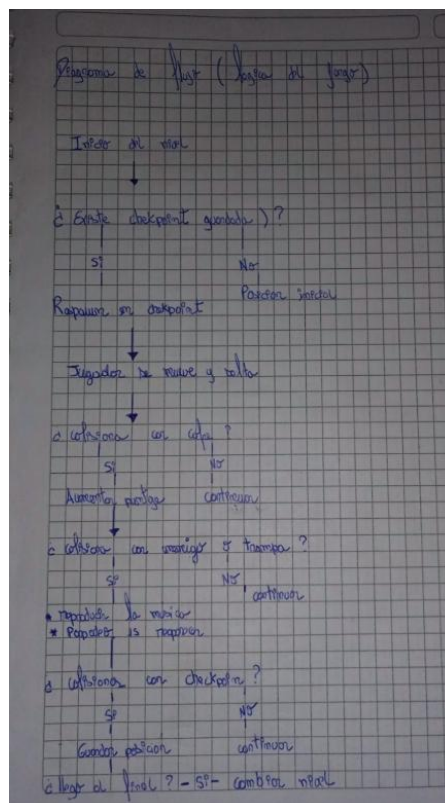
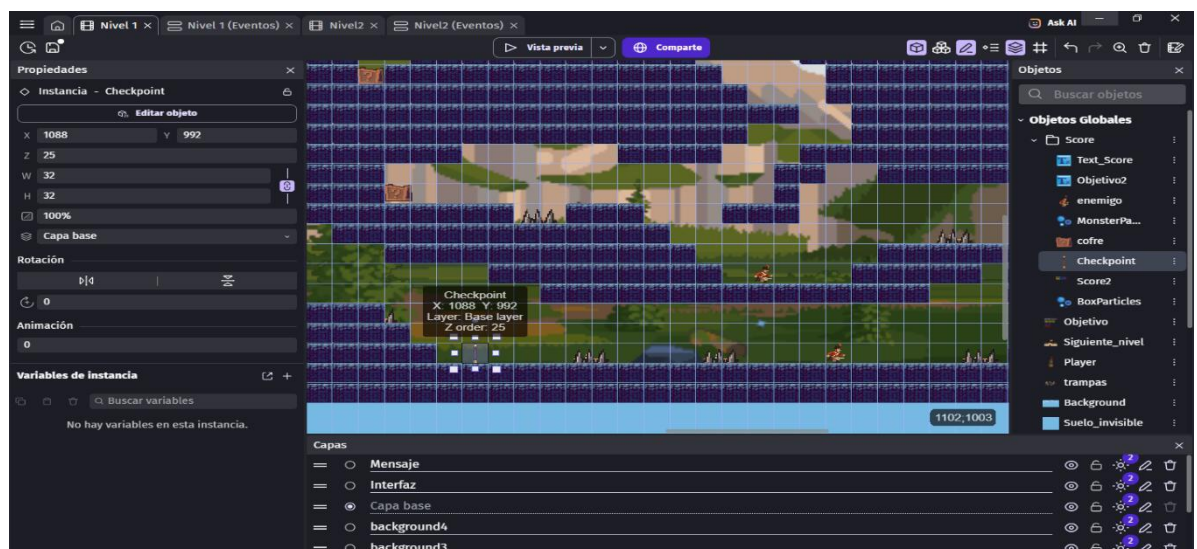
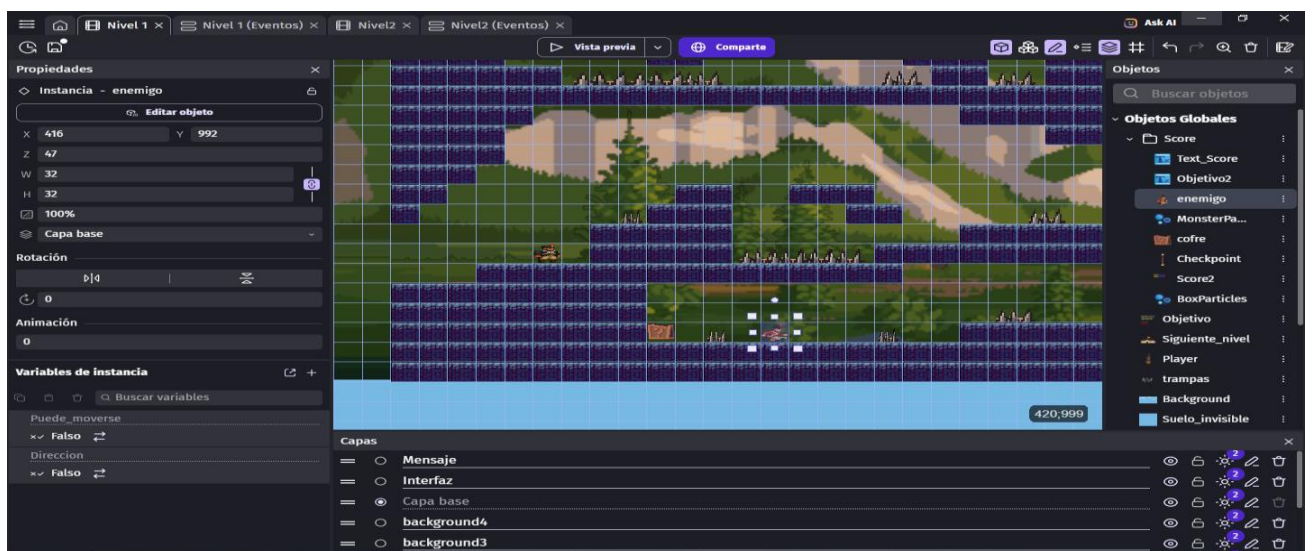
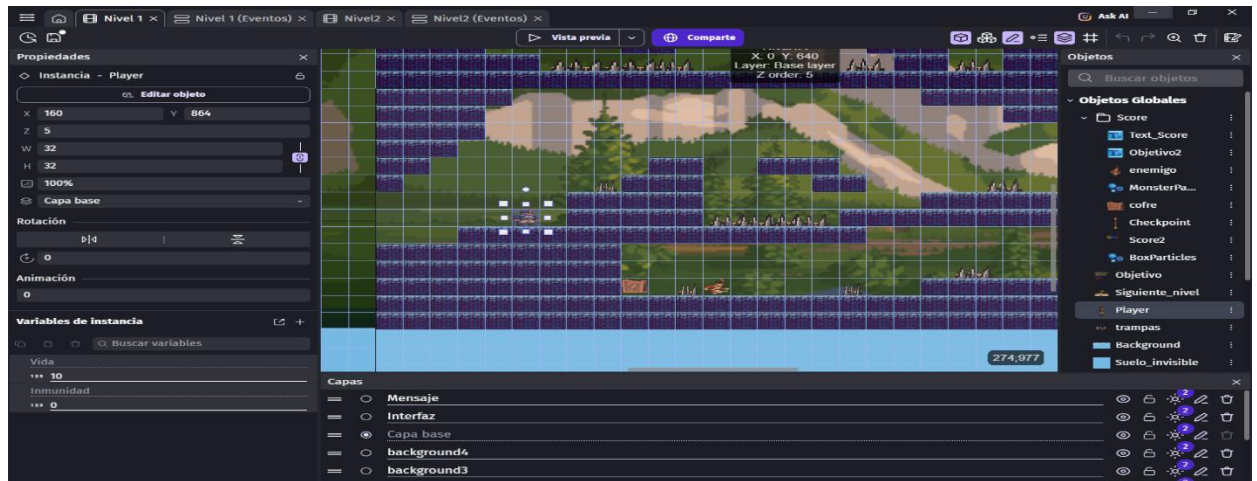


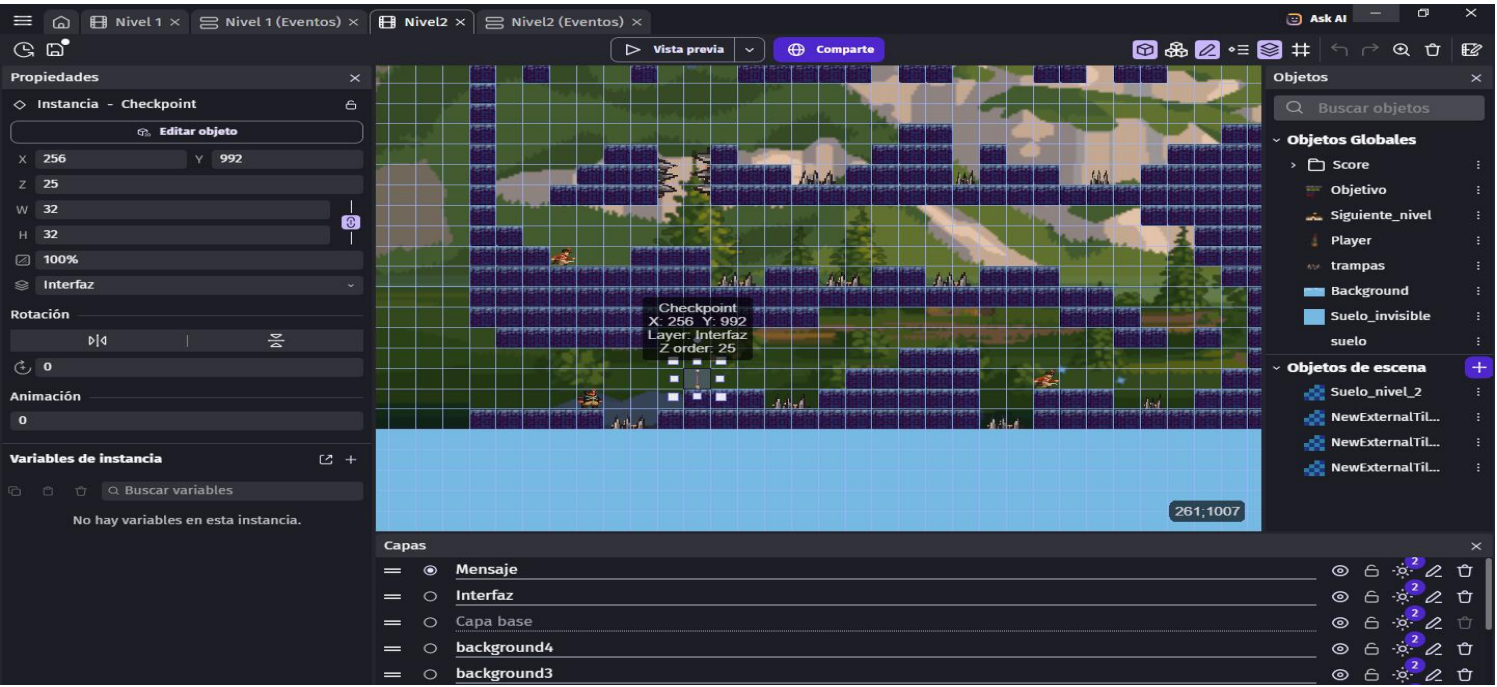
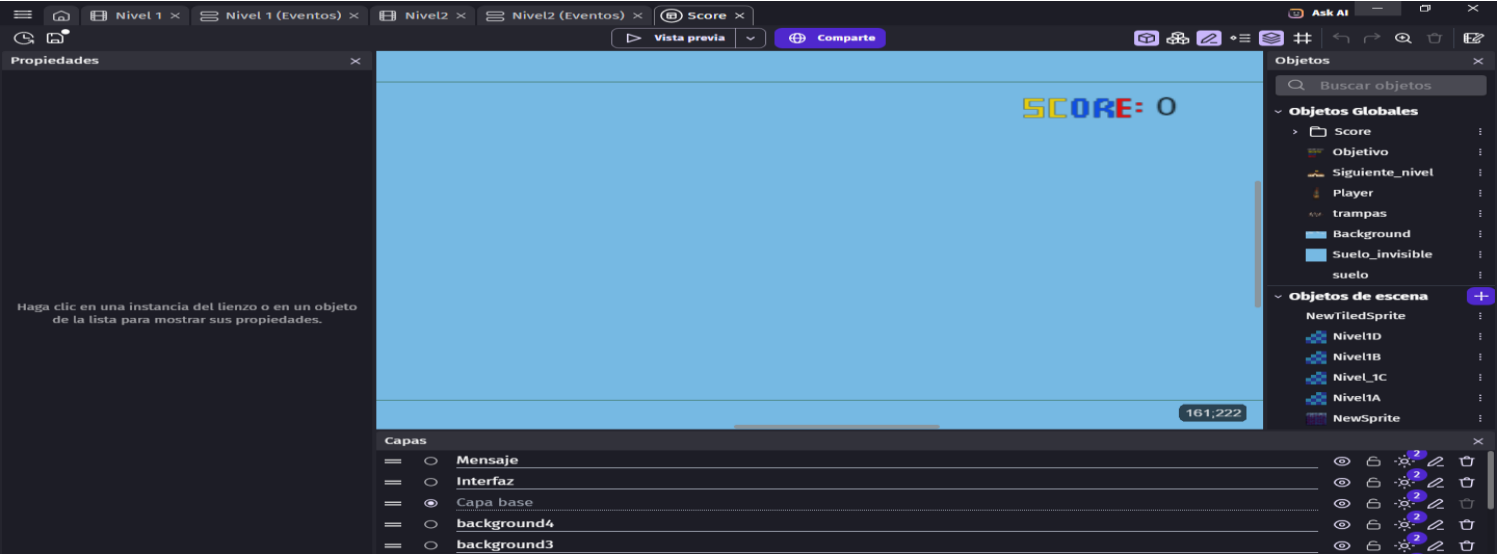
Diagrama de flujo de lógica de juego.



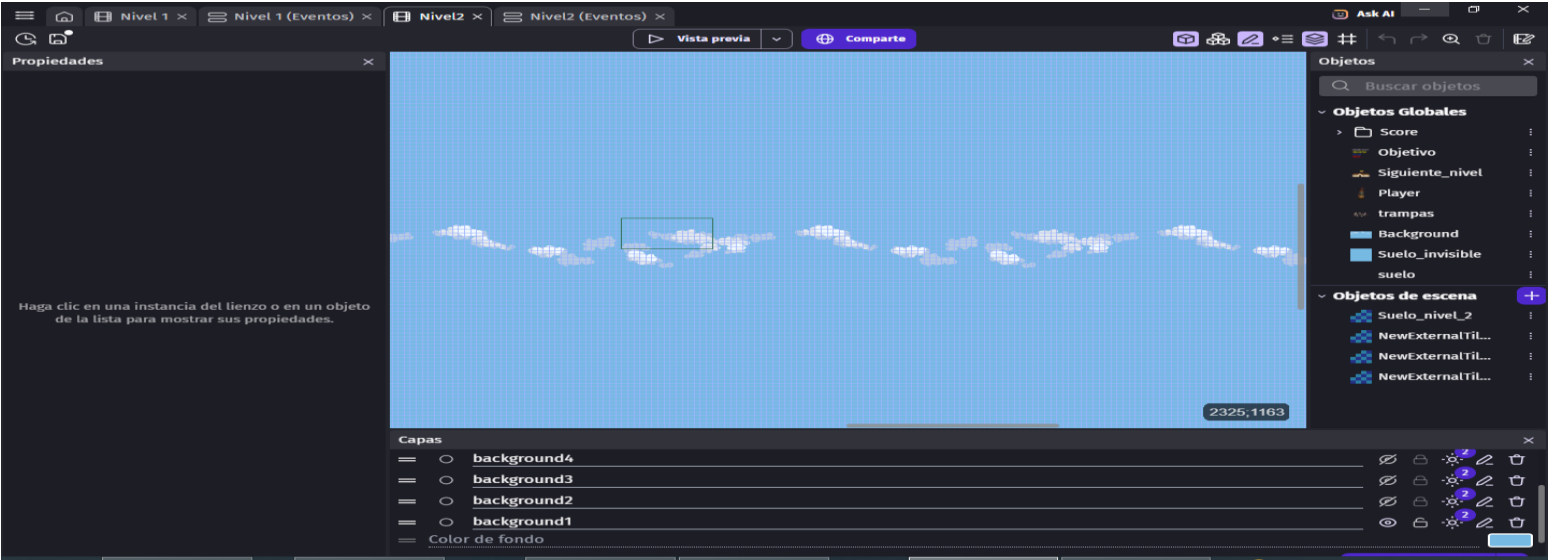
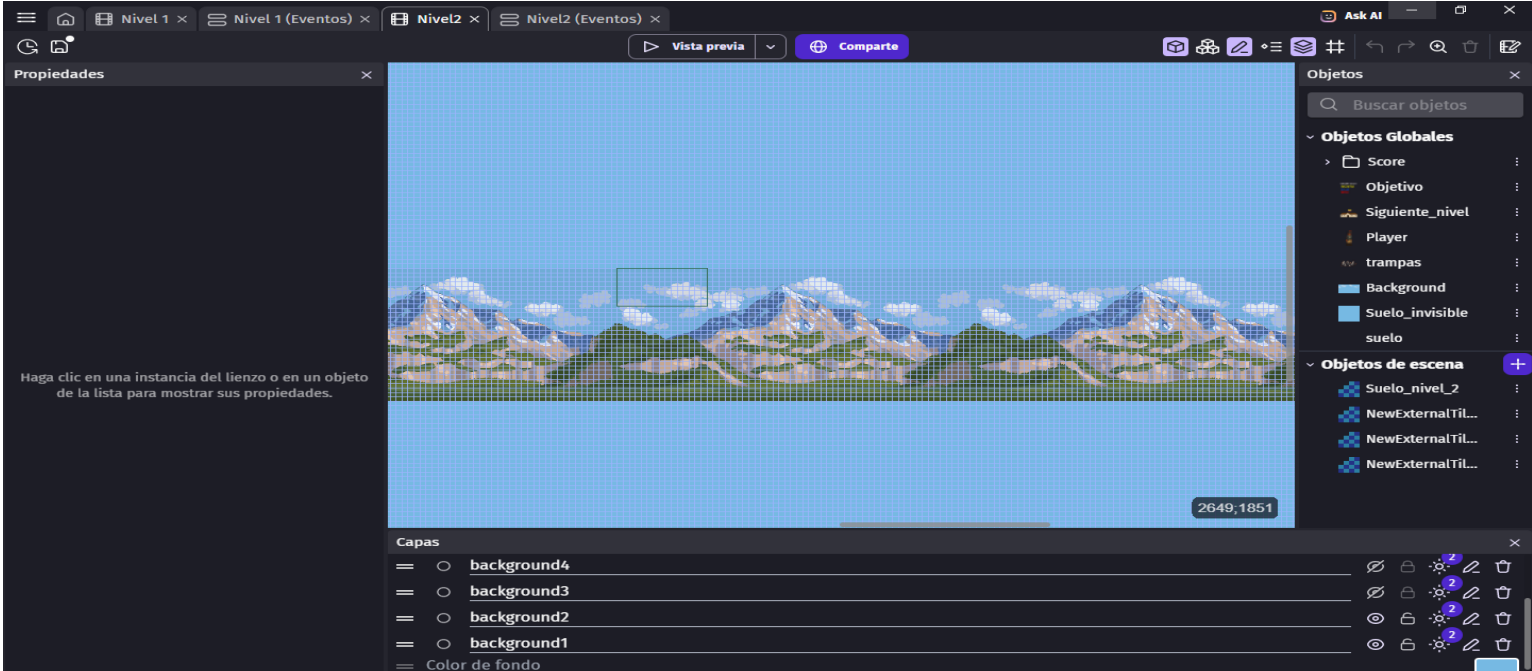
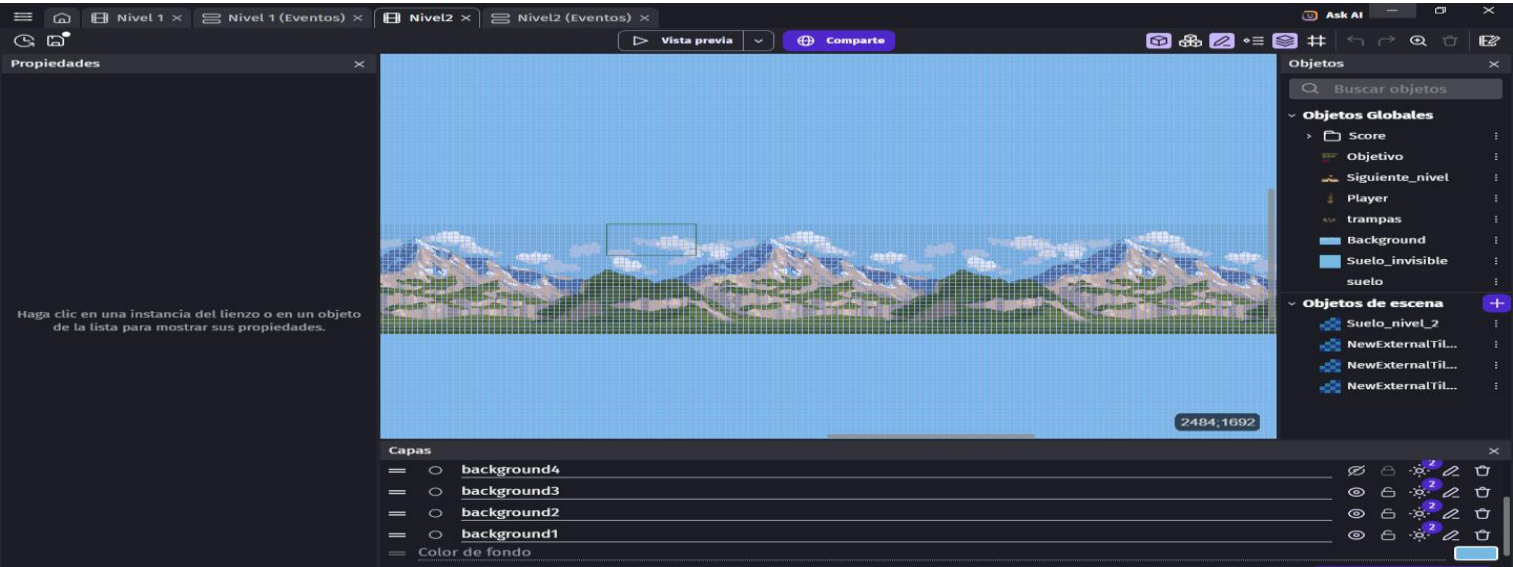
➤ Implementación

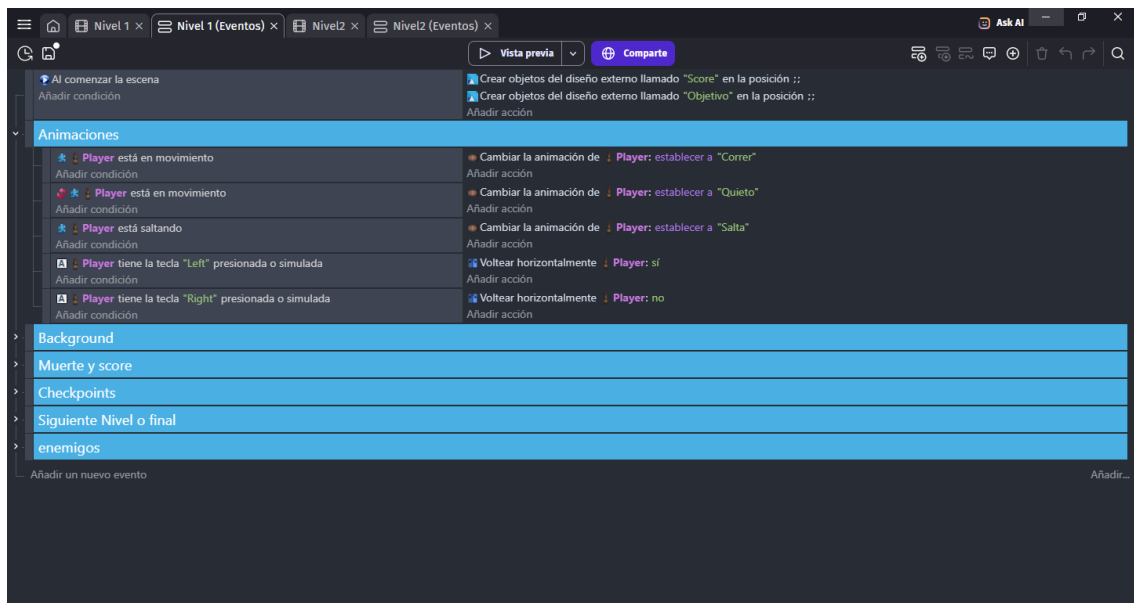
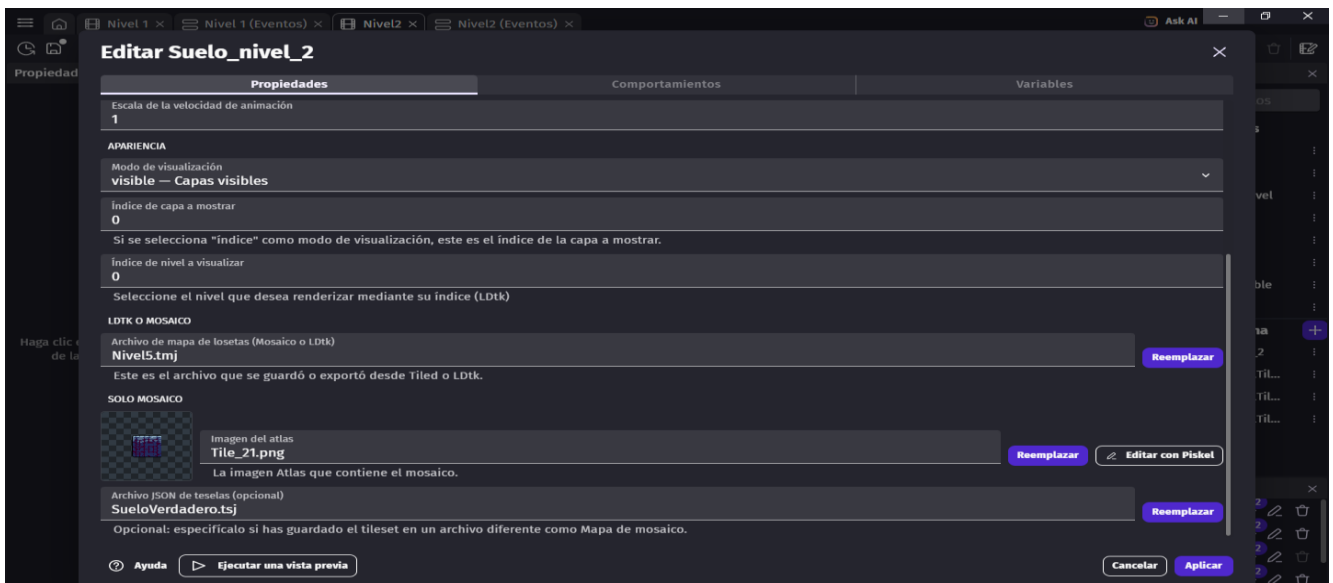
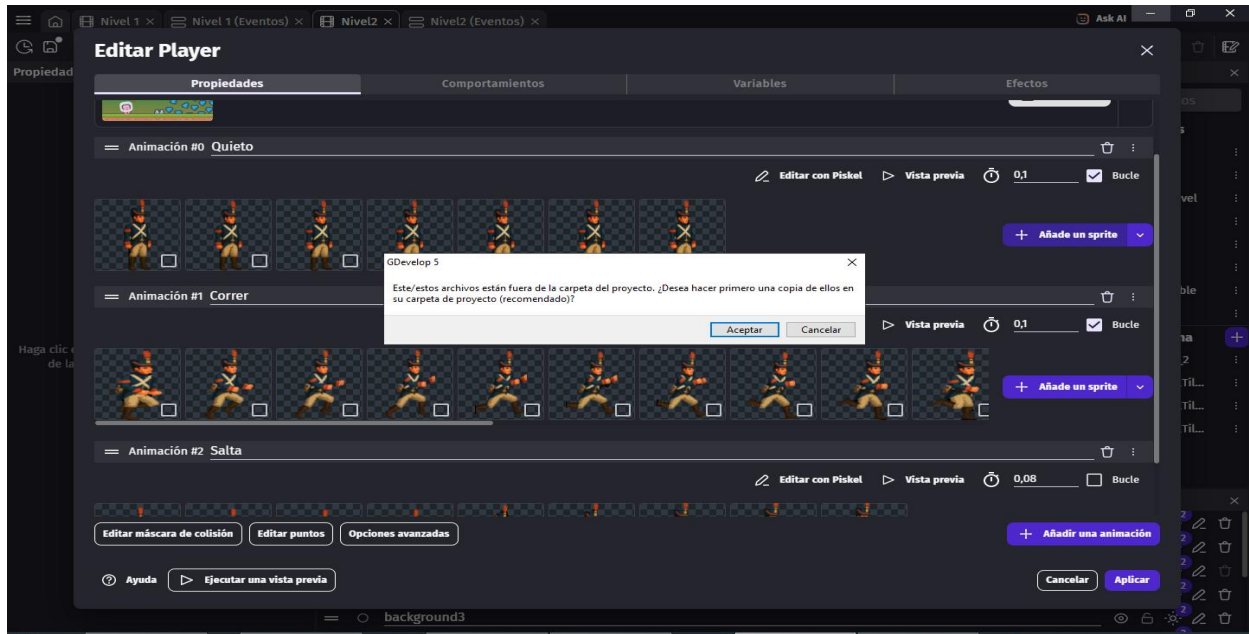
Construcción del videojuego en GDevelop.





Integración de sprites, fondos, sonidos y animaciones.





Configuración de variables, condiciones y eventos.

The screenshot shows the Unity Hierarchy and Console. The 'Animaciones' section is expanded, showing a list of conditions and actions. The 'Background' section is also expanded, showing a list of conditions and actions. A message box at the bottom indicates 'Proyecto guardado correctamente'.

Animaciones

Condición	Acción
Player está en movimiento	Cambiar la animación de Player: establecer a "Correr"
Player está en movimiento	Cambiar la animación de Player: establecer a "Quieto"
Player está saltando	Cambiar la animación de Player: establecer a "Salta"
Player tiene la tecla "Left" presionada o simulada	Voltear horizontalmente Player: sí
Player tiene la tecla "Right" presionada o simulada	Voltear horizontalmente Player: no

Background

Condición	Acción
LA CAMARA SIGUE AL JUGADOR	Centrar la cámara en Player (capa: Capa base)
	Cambiar la posición X de la cámara 0 (capa: Capa base): establecer a lerp(CameraX0, Player.X0, 0.01)
	Cambiar la posición Y de la cámara (capa: Capa base): establecer a lerp(CameraY0, Player.Y0, 0.01)
	Forzar límites de la cámara (izquierda: -3, arriba: -4 derecha: 1925, abajo: 1085, capa: Capa base)
Parallax	Cambiar la posición X de la cámara (capa: "background3"): establecer a CameraCenterX0*0.90
	Cambiar la posición X de la cámara (capa: "background2"): establecer a CameraCenterX0*0.75
	Cambiar la posición X de la cámara (capa: "background1"): establecer a CameraCenterX0*0.4
Bloqueo parallax vertical	Cambiar la posición Y de la cámara (capa: "background3"): establecer a CameraCenterY0
	Cambiar la posición Y de la cámara (capa: "background2"): establecer a CameraCenterY0
	Cambiar la posición Y de la cámara (capa: "background1"): establecer a CameraCenterY0

Proyecto guardado correctamente

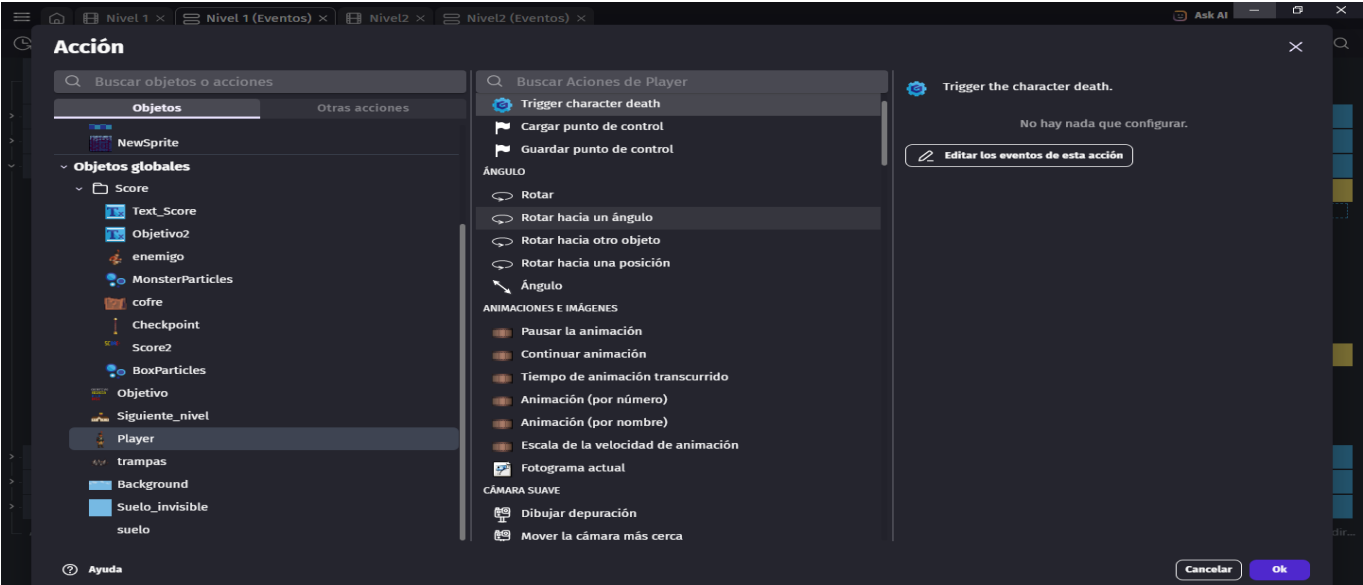
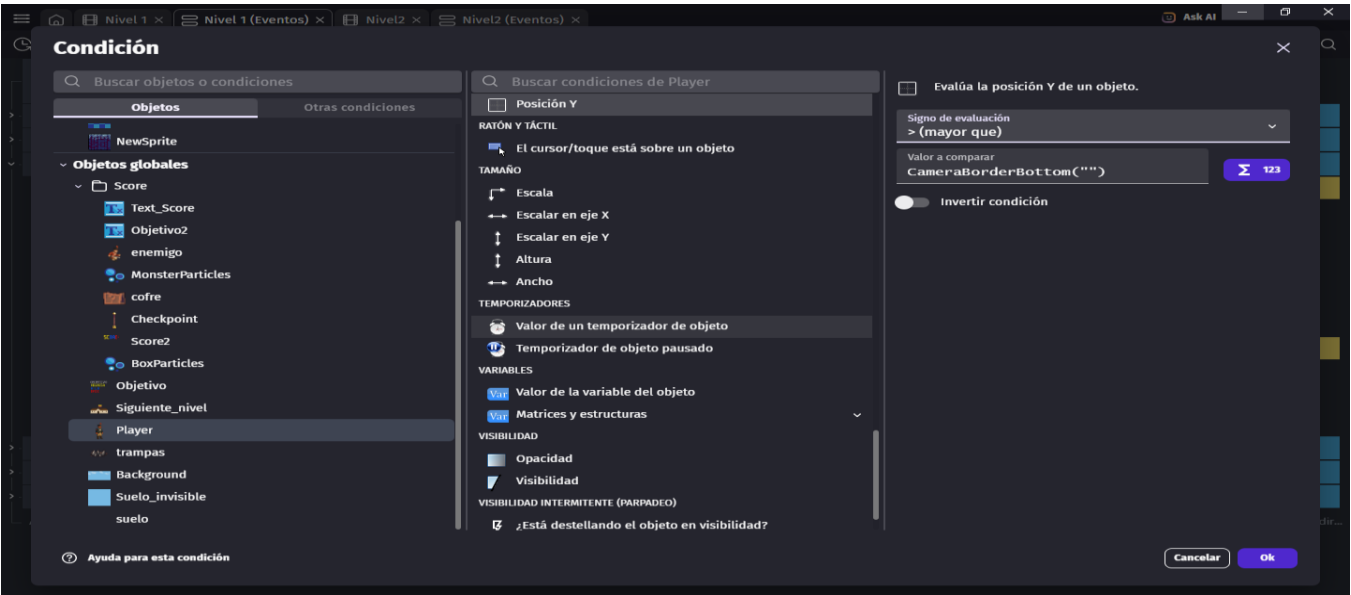
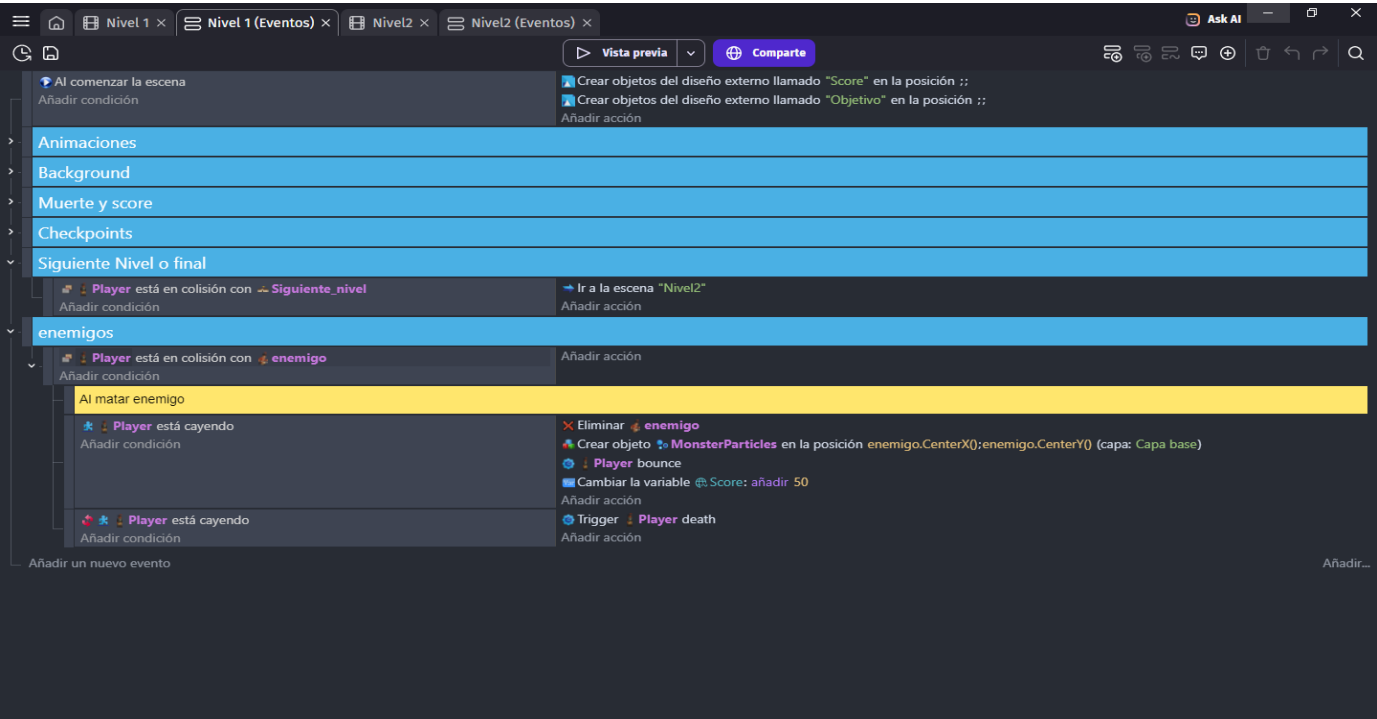
The screenshot shows the Unity Hierarchy and Console. The 'Muerte y score' section is expanded, showing a list of conditions and actions. The 'Checkpoints' section is also expanded, showing a list of conditions and actions.

Muerte y score

Condición	Acción
Actualizar score	Trigger Player death
La posición Y de Player > CameraBorderBottom("")	Crear objeto BoxParticles en la posición cofre.CenterX0;cofre.CenterY0 (capa: Capa base)
Player está en colisión con cofre	Eliminar cofre
	Cambiar la variable Score: añadir 100
	Cambiar el texto de Text_Score: establecer a Score
	Cambiar el texto de Text_Score: establecer a Score
Muerte	Reproducir el sonido rpg-sword-attack-combo19-388939.mp3 en el canal 2, vol: 100, bucle: no
Player está en colisión con trampas	Hacer que Player destelle (parpadee) durante 1 segundos
Ejecutar una vez	Espera 0.3 segundos
	Trigger Player death

Checkpoints

Condición	Acción
Al comenzar la escena	Guardar punto de control "Checkpoint" de Player a Player.X0 (eje x), Player.Y0 (eje y)
Si el jugador colisiona con el objeto de punto de control y este no está activo, se activará el siguiente evento.	Reproducir el sonido sonic-checkpoint.mp3, vol: 20, bucle: no
Player está en colisión con Checkpoint	
La animación de Checkpoint = "Activado"	
	Restablecer todos los puntos de control. Es necesario seleccionar "Seleccionar todo" porque la condición de colisión solo selecciona el punto de control que está en contacto con el jugador.
Elegir todas las instancias de Checkpoint	Cambiar la animación de Checkpoint: establecer a "Desactivado"
Activa el punto de control que la jugadora está tocando.	
	Guardar punto de control "Checkpoint" de Player a Checkpoint.X0 (eje x), Checkpoint.Y0 (eje y)
	Cambiar la animación de Checkpoint: establecer a "Activado"



➤ *Pruebas*

Registro de pruebas funcionales (lista de errores detectados y corregidos).

➤ 1er Error

Personaje no voltea cuando cambiamos de dirección

Solución: Declarar que cuando el objeto Player tiene la tecla left presionada acción “voltear horizontal” a player sí.

➤ 2do error

La cámara se queda totalmente quieta

Solución: Declaramos que la cámara se centre en el jugador y cambiamos la posición X e Y además de declarar límites a la cámara para que no se vea por debajo del suelo

➤ 3er Error

Player no muere

Solución: Cuando el jugador colisione con las trampas o cuando colisione con los enemigos siempre y cuando no este saltando entonces se reproduce un sonido de muerte, parpadea 1 segundo (Se debe instalar la extensión de parpadeo), esperar por 0.3 segundo y llamar a la acción death

➤ 4to Checkpoint no funciona

Solución: Se declara que cuando el player colisiona con el checkpoint, la animación cambie de desactivado a activado se reproduce sonido de activación, después se eligen todas las estancias de el checkpoint para que los demás checkpoint sean desactivados y por ultimo se declara que la acción sea guardar punto de control del player en las coordenadas X e Y del checkpoint.

Evaluación del aprendizaje logrado (mecánica educativa implementada).

La mecánica educativa del videojuego permite que el jugador aprenda sobre la Batalla del Pichincha de forma dinámica. A través de cofres informativos, mensajes dentro del nivel y elementos visuales relacionados al hecho histórico, el jugador recibe datos clave mientras juega. El aprendizaje se evidencia en que el jugador interactúa con estos elementos, recuerda información básica del suceso histórico y comprende su importancia mediante la progresión del nivel y la retroalimentación final.