



DOCUMENTO FUNCIONAL DEL PROYECTO

ALUMNO/GRUPO: Mario Alcaraz y Jon Bornás

1. Introducción y contexto

El proyecto consiste en el desarrollo de un videojuego 2D de plataformas protagonizado por un conejo. El objetivo principal del juego es recolectar zanahorias en diferentes niveles mientras se esquivan enemigos y obstáculos. Cada sesión de juego comienza con un número predeterminado de vidas y un contador de zanahorias, y finaliza automáticamente al perder todas las vidas o completar el nivel. No se guardan partidas ni se registran usuarios, asegurando que cada partida sea independiente y el jugador siempre inicie desde cero.

El proyecto tiene un carácter educativo: permite a los alumnos aprender programación, diseño de niveles, animaciones y mecánicas de juego. Aunque no se soluciona un problema real, el juego es útil para comprender cómo crear una experiencia interactiva y entretenida, centrándose en la jugabilidad inmediata.

El público objetivo es amplio: cualquier persona interesada en juegos 2D podrá jugar sin necesidad de registro ni conocimientos previos. El propósito del proyecto es doble: adquirir experiencia técnica y crear un producto tangible y funcional que muestre las habilidades aprendidas.

2. Análisis de requisitos

2.1. Requisitos funcionales (RF)

Código	Descripción del requisito funcional
RF1	El jugador puede iniciar una nueva partida desde el menú principal sin registro.
RF2	El jugador controla al conejo mediante teclado o gamepad con movimiento y salto.
RF3	Se muestra en pantalla el número de vidas disponibles.

RF4	Se muestra un contador de zanahorias recolectadas en el nivel.
RF5	Se detectan colisiones con enemigos u obstáculos, aplicando daño y efectos visuales.
RF6	La partida finaliza si el jugador pierde todas las vidas o completa el nivel.
RF7	Se muestra retroalimentación visual y sonora al recoger objetos o recibir daño.
RF8	El jugador puede pausar la partida y reanudarla sin pérdida de datos temporales.

2.2. Requisitos no funcionales (RNF)

Código	Descripción del requisito no funcional
RNF1	El juego será compatible con ordenadores de gama media y alta.
RNF2	El tiempo de carga inicial será menor a cinco segundos.
RNF3	La interfaz será clara, con botones grandes y texto legible.
RNF4	El juego mantendrá una tasa de 60 FPS para animaciones fluidas.
RNF5	Los datos de la partida se mantendrán solo mientras dure la sesión; al cerrar se reinician.

2.3. Restricciones

El juego se desarrollará únicamente con **Godot Engine** y **GScript**, sin bases de datos ni almacenamiento persistente. Se utilizará **GitHub** para control de versiones y copias de seguridad. Los recursos disponibles incluyen 7 meses de tiempo y ordenadores del centro y personales, limitando la complejidad de gráficos y animaciones.

3. Análisis de usuarios y roles

Rol	Descripción	Permisos principales
Jugador	Persona que inicia el juego y juega los niveles de forma estándar.	Iniciar y finalizar partidas, controlar al conejo, recolectar

		zanahorias, esquivar enemigos, pausar y reanudar partida.
Administrador / Desarrollador	Persona que realiza pruebas o ajustes en el juego durante el desarrollo.	Acceso a menús de depuración, cambiar niveles, activar/desactivar enemigos o efectos, reiniciar partidas.
Visitante	Usuario que abre el juego sin jugar activamente, puede explorar menús o configuraciones.	Consultar menú principal, ver pantallas de configuración y créditos, salir del juego.
Tester	Persona dedicada a probar niveles específicos para encontrar errores o evaluar jugabilidad.	Iniciar y finalizar partidas, controlar al conejo, ver estadísticas temporales, probar distintas mecánicas de niveles sin afectar datos persistentes.

4. Casos de uso / Escenarios de uso

Código	Nombre del caso de uso	Actor principal	Descripción	Resultado esperado
CU1	Iniciar partida	Jugador	El jugador selecciona "Jugar" en el menú principal.	Se carga el nivel inicial y empieza la partida.
CU2	Controlar personaje	Jugador	El jugador mueve al conejo y salta obstáculos.	El personaje responde en tiempo real.
CU3	Recoger zanahorias	Jugador	El jugador recolecta zanahorias distribuidas en el nivel.	Se actualiza el contador y se muestran efectos visuales y sonoros.

CU4	Superar nivel	Jugador	El jugador cumple los objetivos del nivel.	Se muestra pantalla de nivel completado y se carga siguiente nivel.
CU5	Finalizar partida	Jugador	El jugador pierde todas las vidas o completa el último nivel.	Se muestran estadísticas temporales y opción de volver al menú principal.

5. Modelo de datos o estructura de la información

Durante la sesión se manejan datos temporales:

- **Personaje:** posición, animaciones, velocidad, vidas.
- **Nivel:** número, distribución de zanahorias y enemigos, límites.
- **Zanahorias:** posición y estado de recolección.
- **Enemigos:** posición, patrón de movimiento y daño.
- **HUD:** vidas y zanahorias recolectadas.

Todos los datos se reinician al cerrar la partida.

6. Diseño de la interfaz

El diseño prioriza claridad, retroalimentación inmediata y facilidad de uso. Se contemplan **seis pantallas principales:**

Pantalla 1: Inicio

- Botones: Jugar, Configuración, Salir.
- Fondo animado con paisaje natural y partículas de zanahorias.
- Indicaciones de control visibles y música de bienvenida opcional.

Pantalla 2: Menú de juego / HUD

- Barra superior con vidas y zanahorias.
- Indicador de nivel actual.
- Mensajes temporales al recibir daño o recoger objetos.
- Animaciones y efectos visuales que refuerzan la jugabilidad.

Pantalla 3: Selección de nivel

- Mapa interactivo con iconos de niveles.
- Todos los niveles desbloqueados desde el inicio.
- Botones: Seleccionar nivel, Volver al menú.

Pantalla 4: Juego

- Escenarios variados: bosque, huerto, campo abierto.
- Obstáculos y enemigos simples: rocas, troncos, zorros, trampas.
- HUD en tiempo real con vidas y zanahorias.
- Animaciones y partículas al recoger objetos o recibir daño.
- Música y efectos ajustables desde Configuración.

Pantalla 5: Fin de nivel

- Mensajes: Nivel completado o Has perdido.
- Estadísticas temporales de la sesión.
- Botones: Reintentar, Siguiente nivel, Volver al menú principal.

Pantalla 6: Configuración

- Ajuste de volumen de música y efectos.
- Selección de controles (teclado o gamepad).
- Botón para volver al menú principal.

7. Planificación técnica

- **Lenguaje y entorno:** GDScript en Godot Engine.
- **Control de versiones:** GitHub.
- **Recursos gráficos:** GIMP y Krita para sprites y fondos.
- **Reparto de tareas:** ambos alumnos comparten programación, animaciones, diseño de niveles y pruebas.
- **Cronograma:** 1 mes planificación, 3 meses desarrollo base, 2 meses diseño de niveles, 1 mes pruebas y ajustes.

8. Análisis de riesgos

8.1 Identificación

- Falta de tiempo.
- Mala planificación.
- Problemas técnicos.
- Pérdida de datos.

8.2 Valoración y respuesta

Riesgo	Probabilidad	Impacto	Plan de prevención o contingencia
Falta de tiempo	Media	Alta	Planificación semanal de tareas, entregas parciales.
Mala planificación	Media	Alta	Reuniones de seguimiento y revisión continua.
Problemas técnicos	Media	Media	Pruebas de sistemas antes de implementación.

Pérdida de datos	Baja	Alta	Copias de seguridad frecuentes en GitHub.
------------------	------	------	---

9. Validación y criterios de éxito

El juego será exitoso si:

- Funciona sin errores y es estable.
- Los niveles son jugables y superables.
- Los controles responden correctamente.
- La interfaz es clara y comprensible.
- Cada partida comienza desde cero y finaliza correctamente.

Se realizarán pruebas funcionales, de usuario y de rendimiento para comprobar animaciones, sonidos, HUD y fluidez.

10. Conclusión

El sistema permitirá jugar niveles independientes, recolectando zanahorias y esquivando enemigos, con partidas temporales que no se guardan. Se usará Godot Engine y GDScript, con GitHub para control de versiones. El proyecto aporta experiencia práctica en programación, diseño de niveles y animaciones.

Próximos pasos:

- Implementar niveles y enemigos.
- Añadir animaciones y efectos visuales.
- Construir HUD completo y pantallas de interfaz.
- Realizar pruebas funcionales y de usuario.

Reflexión final:

“El análisis funcional proporciona una visión clara del proyecto, permitiendo iniciar el desarrollo de un videojuego divertido, educativo y totalmente jugable, donde cada partida es independiente y accesible para cualquier usuario.”