DOCUMENTO FUNCIONAL DEL PROYECTO

ALUMNO/GRUPO: Adrián López-Nathan Ruano

---

1. Introducción y contexto

Nuestro proyecto es un videojuego 2D de plataformas hecho con Godot Engine.
Queremos hacer un juego divertido y jugable que demuestre lo que hemos aprendido en programación, diseño y gestión de proyectos.

Problema o necesidad:
Muchos juegos de plataformas son de pago. Queremos crear un juego sencillo, entretenido y gratuito que se pueda jugar en PC.

Usuario o cliente final:
Personas que les gusten los juegos 2D, tanto casuales como estudiantes o centros educativos que quieran usarlo como ejemplo.

Solución propuesta:
Un juego de plataformas en 2D con mecánicas clásicas: saltar, recoger objetos, esquivar enemigos y llegar a la meta. Tendrá un estilo pixel art y será completo y optimizado.
Además, el juego tendrá un servidor que registre el tiempo que tarda cada jugador en completar los niveles, para poder comparar puntuaciones y tiempos.

---

2. Análisis de requisitos

2.1. Requisitos funcionales (RF)

Código	Qué hace el juego
RF1	**Permitir empezar una partida nueva.**
RF2	**Permitir mover al personaje (caminar, saltar, caer, atacar).**

RF3	**Detectar colisiones con enemigos y plataformas.**
RF4	**Recoger objetos que sumen puntos.**
RF5	**Mostrar vidas y puntuación en pantalla.**
RF6	**Si se pierden todas las vidas, mostrar “Game Over”.**
RF7	**Tener varios niveles con dificultad creciente.**
RF8	**Guardar la mejor puntuación localmente.**
RF9	**Enviar al servidor el tiempo que tarda el jugador en completar cada nivel.**
RF10	**Consultar desde el juego los mejores tiempos registrados en el servidor.**

2.2. Requisitos no funcionales (RNF)

Cómo debe comportarse el sistema.
Incluye aspectos como rendimiento, seguridad, compatibilidad o facilidad de uso.

Código	**Descripción del requisito no funcional**
RNF1	El juego devera ir a 60 fps
RNF2	Se debe poder jugar con mando

2.3. Restricciones

Condiciones o limitaciones del proyecto.

Tecnologías: Godot 4.x, GDScript, servidor propio (ej. Node.js, Python o similar) para registrar tiempos.

Recursos: 6 meses de curso, dos personas, ordenadores del aula y programas gratuitos (Krita, Aseprite, Audacity).

Limitaciones: El juego debe poder jugarse offline en su parte básica, pero enviar tiempos al servidor cuando haya conexión.

---

3. Análisis de usuarios y roles

Objetivo: identificar quién usará el sistema y qué podrá hacer.

Describe los distintos tipos de usuario, sus necesidades y sus permisos.

Rol	**Descripción**	**Permisos principales**
programadores	Gestionan los recursos y código.	Modificación de datos, código, administradores del servidor.
Usuario	Utiliza el sistema para realizar acciones básicas.	Crear, consultar y modificar sus propios datos.
jugador	Consulta información sin interactuar con el sistema	Sin permisos

---

4. Casos de uso / Escenarios de uso

Objetivo: mostrar cómo interactúan los usuarios con el sistema.

Selecciona de tres a cinco casos principales y descríbelos brevemente.

Código	Nombre del caso de uso	Actor principal	Descripción	Resultado esperado
CU1	Descargar el juego	usuario	El usuario descarga el juego	EL usuario tendrá acceso al juego
CU2	usar el juego	Usuario	El usuario podrá jugar al juego	el usuario podrá jugar al juego perfectamente
CU3	Cerrar el juego	Usuario	EL usuario podrá cerrar el juego sin problemas de congelamiento	EL usuario podrá cerrar el juego sin problemas

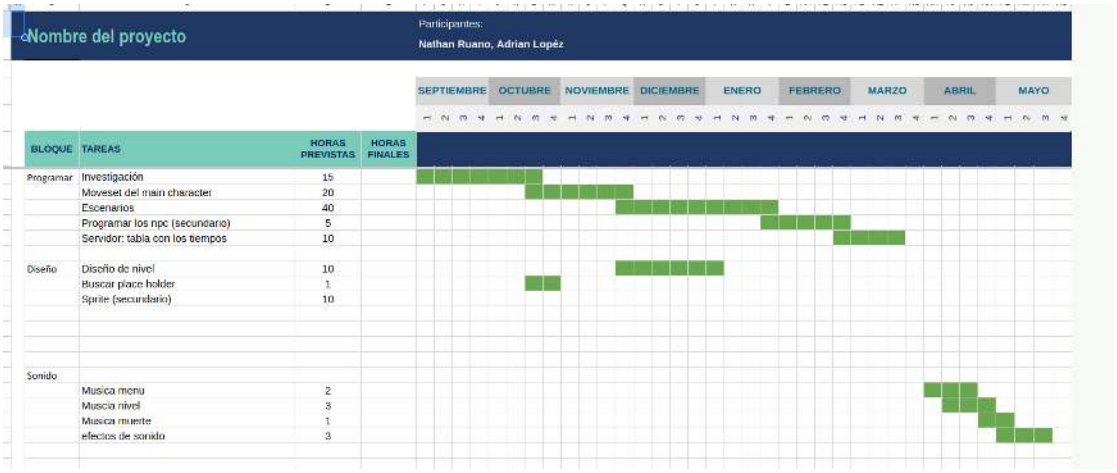
---

5. Modelo de datos o estructura de la información

Objetivo: representar la información que gestionará el sistema.

Incluye las entidades principales (tablas u objetos) y las relaciones entre ellas.

El juego guardará información del jugador, los niveles, los enemigos y los objetos. Cada jugador tendrá sus vidas, puntuación y tiempos por nivel. Los enemigos y objetos estarán definidos por su tipo, valor y posición. El servidor almacenará los tiempos que tarda cada jugador en completar los niveles para crear un ranking con las mejores marcas.



---

6. Diseño de la interfaz

Objetivo: visualizar la estructura y navegación del sistema antes de desarrollarlo.*

El juego tendrá un estilo pixel art clásico, con menús simples y controles por teclado . En el menú principal se podrá empezar una partida, ver el ranking de mejores tiempos o salir del juego. Durante la partida se mostrará el personaje, las plataformas, los enemigos, los puntos y el tiempo que lleva jugando. Al completar un nivel aparecerá una pantalla de victoria donde se enviará el tiempo al servidor, mientras que si el jugador pierde todas las vidas se mostrará una pantalla de “Game Over” con la opción de reiniciar o volver al menú. También habrá una pantalla de ranking donde se podrán ver los mejores tiempos guardados en el servidor. El estilo general del juego usará colores vivos, gráficos pixelados, música ligera y sonidos simples, todo pensado para que la experiencia sea clara, fluida y agradable de jugar.

---

7. Planificación técnica

Objetivo: planificar el desarrollo del proyecto.

Indica las tecnologías y herramientas que se utilizarán, y cómo se organizará el trabajo.

Lenguajes y frameworks: GDScript y C#

Base de datos: MySql

Herramientas de diseño o edición: Krita, Godot

Reparto de tareas (si es en grupo): Trello

Cronograma (puede incluir un diagrama de Gantt): Diagrama de Grantt

---

8. Análisis de riesgos

Objetivo: identificar posibles problemas y cómo se afrontarán.

8.1. Identificación de riesgos

Durante el desarrollo del juego pueden aparecer varios problemas que afecten al tiempo o al funcionamiento del proyecto.
Los riesgos principales son los siguientes:

- Falta de tiempo o mala planificación.
- Problemas técnicos con Godot o el servidor.
- Pérdida de datos del proyecto.
- Archivos dañados o incompatibilidades.
- Que algún miembro del grupo no pueda continuar.
- Fallos en la conexión o en el envío de datos al servidor.

8.2. Valoración y respuesta

Clasifica cada riesgo según su probabilidad e impacto, e indica cómo se mitigará.

Riesgo	**Probabilidad**	**Impacto**	**Plan de prevención o contingencia**
Falta de tiempo	Alta	Alta	Dividir tareas y fijar entregas intermedias.
Problemas técnicos	Media	Media	Copias de seguridad y comprobaciones antes de programar.
Pérdida de datos	Baja	Alta	Hacer copias de seguridad semanales y subir constantemente los datos a github
Fallo del servidor	Media	Alta	Reiniciar y actualizar el servidor constantemente y tener preparado un re instalador en caso de fallo.

---

9. Validación y criterios de éxito

Objetivo: definir cómo sabremos que el proyecto funciona correctamente.

Criterios de aceptación:

El juego se puede abrir sin errores y funciona correctamente.

El jugador puede moverse, saltar, recoger objetos y pasar niveles.

Al menos 3 niveles jugables y con enemigos.

El juego mantiene una tasa estable de 60 FPS.

Cuando el jugador termina un nivel, su tiempo se envía al servidor y se guarda correctamente.

El ranking de mejores tiempos se puede consultar desde el juego.

Pruebas previstas:

Pruebas funcionales: comprobar que todas las mecánicas funcionan (movimiento, colisiones, puntuación, envío de datos).

Pruebas de servidor: verificar que el tiempo de los jugadores se registra y se muestra sin errores.

Pruebas de usuario: hacer que otras personas jueguen y vean si entienden cómo se juega y si todo responde bien.

Pruebas de rendimiento: revisar que no haya caídas de FPS ni pantallas que tarden en cargar.

Indicadores de calidad:

Sin errores ni cierres inesperados.

El juego es fluido y entretenido.

Los datos del servidor son correctos y actualizados.

---

10. Conclusión

Objetivo: cerrar el análisis y preparar la siguiente fase.

Con este documento tenemos una visión clara del juego 2D de plataformas con servidor para registrar tiempos.

Ya sabemos las funciones principales, las herramientas que usaremos y cómo organizaremos el trabajo durante los seis meses.

El siguiente paso será configurar el entorno de Godot, crear la estructura inicial del servidor y empezar con las mecánicas básicas del juego (movimiento, salto, colisiones y envío de datos).

Nuestro objetivo es entregar un juego completo, divertido y bien hecho, que refleje todo lo aprendido durante el curso.