



# DOCUMENTO FUNCIONAL DEL PROYECTO

ALUMNO/GRUPO: Eric Aguilar Pulido

---

## 1. Introducción y contexto

*Objetivo:* Haré un juego de aventura y plataformas usando el motor gráfico de Godot Engine 4 desde cero, usando tutoriales y aprendiendo sobre la marcha el lenguaje de GDScript. Debe de quedar un buen juego.

---

## 2. Análisis de requisitos

### 2.1. Requisitos funcionales (RF)

*Qué debe hacer el sistema.*

Enumera las funciones principales, numeradas como RF1, RF2, etc.

Código	Descripción del requisito funcional
RF1	El programa deberá de ejecutarse bien sin bugs o problemas.
RF2	El usuario deberá de poder usar todas las funciones correctamente.
RF3	No deben de haber bugs visual, pérdida de rendimiento, lag, bajos FPS.

---

### 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 programa deberá de ejecutarse bien sin bugs o problemas.
RNF2	El usuario deberá de poder usar todas las funciones correctamente.
RNF3	No deben de haber bugs visual, pérdida de rendimiento, lag, bajos FPS.

---

### 2.3. Restricciones

*Condiciones o limitaciones del proyecto.*

- Lenguajes o tecnologías obligatorias: Godot Engine 4, GDScript
  - Recursos disponibles (tiempo, equipo, materiales): PC de clase y personal, herramientas gratuitas.
  - Dependencias o limitaciones técnicas: Solo se podrá jugar en un PC.
- 

## 3. Análisis de usuarios y roles

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

Rol	Descripción	Permisos principales
Usuario	Utiliza el sistema para jugar al videojuego.	Podrá jugar al juego con todas las funciones disponibles y desarrolladas.
IA	Podrá eliminar al jugador	Podrá eliminar al usuario cuando haya un momento de pelea o enfrentamiento

---

## 4. Casos de uso / Escenarios de uso

Selecciona de tres a cinco casos principales y descríbelos brevemente.

Código	Nombre del caso de uso	Actor principal	Descripción	Resultado esperado
CU1	Iniciar juego	Usuario	El usuario empieza a jugar	El juego empieza sin problemas
CU2	Leer contexto	Usuario	Leerá el contexto de la historia para poder entenderla correctamente	El usuario entenderá todo correctamente
CU3	Empezar a jugar	Usuario	Deberá de mover al personaje correctamente con todas sus mecánicas	Todo funcionara correctamente

## 5. Modelo de datos o estructura de la información

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

Ejemplo:

- Entidades: Usuario, Reserva, Recurso.
- Relaciones: Un usuario puede tener varias reservas, cada reserva corresponde a un recurso.

(Opcional: añadir diagrama ER o tabla resumen de campos.)

---

## 6. Diseño de la interfaz

Incluye bocetos o capturas de las pantallas principales y una breve descripción de su función.

Para cada pantalla, indica:

- Nombre de la pantalla.
- Funcionalidad principal.
- Casos de uso relacionados.

---

## 7. Planificación técnica

- Lenguajes y frameworks: Godot Engine 4
- Base de datos: -
- Herramientas de diseño o edición: Freesound, GNU, Itch.io.
- Reparto de tareas (si es en grupo): NO ES EN GRUPO
- Cronograma (puede incluir un diagrama de Gantt):

---

## 8. Análisis de riesgos

### 8.1. Identificación de riesgos

Ejemplos:

- Falta de tiempo o mala planificación.
- Problemas técnicos o de código programando el videojuego.
- Pérdida del progreso.
- Cambios en los requisitos.

### 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	Probar todo el programa varias veces.
Pérdida del progreso	Baja	Alta	Los datos están subidos a GitHub.

---

## 9. Validación y criterios de éxito

- Criterios de aceptación (qué debe cumplirse para darlo por válido).
  - Pruebas previstas (funcionales, de usuario, de rendimiento).
  - Indicadores de calidad o resultados esperados.
- 

## 10. Conclusión

Resume las decisiones principales tomadas durante el análisis:

- Qué funciones tendrá el sistema.
- Qué tecnologías se utilizarán.
- Qué valor aporta el proyecto.

Indica los próximos pasos:

- Preparar el entorno de desarrollo.
- Crear la estructura inicial del repositorio y la base de datos.
- Empezar la implementación de los casos de uso prioritarios.

Finaliza con una breve reflexión o frase de cierre, por ejemplo:

“El análisis funcional proporciona una visión clara y realista del proyecto. A partir de aquí, comienza la fase de desarrollo.”