Institut Puig Castellar
Projecte / Crèdit de síntesi
Document funcional

DOCUMENTO FUNCIONAL DEL PROYECTO

ALUMNO/GRUPO:___Jiayi Chen DAM 2nB_____

---

1. Introducción y contexto

Me basaré en un juego que desarrollé durante mi formación en SMX, al cual realizaré una profunda mejora y ampliación de funcionalidades para hacerlo más completo, variado y atractivo. Este proyecto está dirigido principalmente a jugadores jóvenes y de mediana edad, combinando elementos de **exploración, narrativa, disparos y humor**, con el objetivo de ofrecer una experiencia inmersiva que mantenga el interés y el disfrute del jugador a lo largo del juego.

En la versión anterior, intenté implementar un **modo multijugador en línea**, que permitía a varios jugadores interactuar, cooperar o competir dentro de un mismo entorno virtual. Este sistema se basaba inicialmente en el servicio de backend **Nakama**, pero debido a las limitaciones de desarrollo en ese momento, la funcionalidad no llegó a completarse del todo.

En la nueva versión, planeó **crear un servicio backend** para el proyecto: **una API** independiente desarrollada en **Python**, responsable de gestionar la lógica de red, la autenticación de usuarios, la sincronización de datos y las interacciones en tiempo real dentro del juego. El servidor mantendrá una comunicación eficiente con la **base de datos** para garantizar un rendimiento fluido y estable.

El núcleo del juego estará desarrollado con el **motor Godot**, una herramienta potente, de código abierto y muy flexible, que permite implementar sin dificultad tanto modos de un solo jugador como modos multijugador. En el apartado visual y de modelado, utilizaré **Blender** para crear personajes, escenarios y objetos en 3D de alta calidad, con el fin de desarrollar un estilo artístico único que refleje el tono humorístico y narrativo del juego.

Durante el proceso de desarrollo, emplearé **Git** para el control de versiones, lo que permitirá mantener la eficiencia y la seguridad en la expansión de funcionalidades y el mantenimiento del código. Además, el proyecto contará con una **plataforma web complementaria**, donde los jugadores podrán registrarse, consultar información sobre el juego y descargar el cliente.

En resumen, este proyecto tiene como objetivo ofrecer un **juego moderno, dinámico y con un fuerte componente social**, que combine la narrativa y la exploración con las ventajas de las tecnologías actuales de desarrollo de videojuegos y aplicaciones web, brindando a los jugadores una experiencia inmersiva llena de diversión e innovación.

2. Análisis de requisitos

2.1. Requisitos funcionales (RF)

Codigo	Descripción del requisito funcional	Tipo
RF1	Registro de nuevos usuarios mediante la introducción de credenciales básicas (nombre, correo y contraseña).	Gestión de usuarios
RF2	Autenticación de usuarios y acceso al sistema mediante credenciales válidas.	Gestión de usuarios
RF3	Validación de la unicidad del nombre de usuario durante el proceso de registro.	Validación
RF4	Cifrado y verificación segura de contraseñas antes de generar credenciales de acceso basadas en JWT.	Seguridad
RF5	Interacción del jugador con objetos del entorno para la obtención de ítems.	Inventario
RF6	Gestión completa del inventario, incluyendo la adición, uso y eliminación de objetos.	Inventario
RF7	Activación de efectos asociados al uso de objetos, como la recuperación de salud u otros estados.	Inventario
RF8	Obtención de objetos mediante recompensas, derrotas de enemigos o compras dentro del juego.	Inventario
RF9	Definición de atributos del personaje, incluyendo estadísticas, estados y niveles.	Datos
RF10	Comunicación entre cliente (Godot) y backend mediante APIs para operaciones CRUD.	Backend
RF11	Almacenamiento de datos de sesión tanto en el cliente local como en la base de datos.	Persistencia
RF12	Persistencia automática del progreso del jugador en la base de datos.	Persistencia
RF13	Visualización detallada de habilidades, incluyendo nivel, efecto y tiempo de reutilización.	Habilidades
RF14	Desbloqueo de habilidades condicionado a recursos disponibles o nivel del personaje.	Habilidades
RF15	Mejora progresiva de habilidades mediante incremento de nivel.	Habilidades
RF16	Integración de las habilidades dentro del sistema de combate del juego.	Combate
RF17	Aplicación de tiempos de reutilización (cooldown) tras el uso de habilidades.	Habilidades

RF18	Implementación de un sistema de combate contra enemigos con cálculo de daño y estados.	Combate
RF19	Interacción con NPCs para diálogo, eventos o futuras mecánicas de misión.	Interacción
RF20	Implementación de un sistema de progresión basado en experiencia y niveles.	Progresión
RF21	Definición de rutas evolutivas del personaje mediante un sistema genético.	Genético
RF22	Incorporación de un sistema de núcleo para mejoras avanzadas de habilidades y atributos.	Núcleo
RF23	Guardado y recuperación del estado del juego a través del backend.	Persistencia
RF24	Provisión de datos estáticos del juego (ítems, habilidades, enemigos) mediante APIs.	Datos
RF25	Inclusión de un sistema de tutorial para guiar a los nuevos jugadores.	UX
RF26	Aplicación de un mecanismo de desbloqueo progresivo de contenido basado en el avance del jugador.	Progresión
RF27	Creación y publicación de contenido dentro del sistema de comunidad.	Comunidad
RF28	Interacción social mediante comentarios y acciones de “me gusta” en publicaciones.	Comunidad
RF29	Gestión de relaciones sociales mediante seguimiento o eliminación de seguimiento entre usuarios.	Comunidad
RF30	Generación de notificaciones en respuesta a eventos sociales.	Comunidad
RF31	Participación de los usuarios en eventos organizados dentro de la comunidad.	Comunidad
RF32	Implementación de un sistema de check-in diario con recompensas asociadas.	Comunidad
RF33	Sincronización de datos entre el sistema de comunidad y el juego.	Integración
RF34	Envío de recompensas obtenidas en la comunidad al sistema de correo interno del juego.	Integración

2.2. Requisitos no funcionales (RNF)

Código	Categoría	Descripción del requisito no funcional
RNF1	Rendimiento	El tiempo de respuesta de las APIs backend no deberá superar los 500 ms en condiciones normales.
RNF2	Portabilidad	El sistema será compatible con Windows, Linux y macOS.

RNF3	Usabilidad	La interfaz de usuario deberá ser clara, intuitiva y consistente.
RNF4	Usabilidad	Todas las interacciones deberán proporcionar retroalimentación visual (confirmación, cancelación, carga).
RNF5	Usabilidad	Los mensajes del sistema deberán ser comprensibles y evitar terminología técnica innecesaria.
RNF6	Mantenibilidad	El código seguirá convenciones estándar de nomenclatura y documentación.
RNF7	Arquitectura	El sistema adoptará un diseño modular con bajo acoplamiento entre subsistemas.
RNF8	Escalabilidad	El sistema permitirá la incorporación de nuevos módulos sin cambios estructurales significativos.
RNF9	Fiabilidad	Los datos del usuario se guardarán automáticamente en el servidor para evitar pérdidas.
RNF10	Seguridad	Se implementará autenticación segura (JWT) y cifrado de contraseñas.
RNF11	Disponibilidad	Los servicios backend deberán garantizar alta disponibilidad y tolerancia a fallos.
RNF12	Consistencia	Se garantizará la coherencia entre datos locales y datos en la nube.
RNF13	Testabilidad	El sistema permitirá pruebas automatizadas.
RNF14	Despliegue	El sistema podrá desplegarse en entornos virtualizados y será adaptable a contenedores (Docker).

2.3. Restricciones

Condiciones o limitaciones del proyecto.

- Lenguajes o tecnologías obligatorias:
  - PostgreSQL
  - GDScript
  - Python o Go
  - Java
  - Git

- Recursos disponibles (tiempo, equipo, materiales):
  - Software Libre
  - Portatil
  - Hasta segunda semana de mayo

- Dependencias o limitaciones técnicas:
  - **Dependencias:**
    1. El cliente Godot se basa en la interfaz API proporcionada por Flask para conectarse a la base de datos
  - **Limitaciones:**

1. El número de desarrolladores es limitado y el ciclo de desarrollo es ajustado.
2. Los modelos exportados desde Blender deben ser compatibles con la versión Godot.
3. Eficiencia limitada en el modelado de escenas y personajes

---

3. Análisis de usuarios y roles

Rol	Descripción	Permisos principales
Administrador	El usuario de máxima autoridad del sistema, responsable de supervisar y mantener todo el sistema.	- Administrar cuentas de usuario (CRUD) - Administrar bibliotecas de objetos y habilidades - Ver registros y el estado del sistema
Usuario	Utiliza el sistema para iniciar sesión, interactuar con juegos y administrar mochilas y habilidades.	- Inicia sesión y crea una cuenta - Juega
Visitante	Prueba el juego y comprende la interfaz y el funcionamiento.	SIN Permiso

---

4. Casos de uso / Escenarios de uso

Código	Nombre del caso de uso	Actor principal	Descripción	Resultado esperado
CU 1	Registro de usuario	Visitante	El usuario se registra con correo, nombre y contraseña.	Usuario creado correctamente en la base de datos.
CU 2	Inicio de sesión	Usuario	Autenticación mediante credenciales y JWT.	Acceso concedido y datos del jugador cargados.
CU 3	Selección de personaje	Usuario	El jugador selecciona uno de los personajes disponibles.	Personaje activo asignado correctamente.
CU 4	Control del personaje	Usuario	Movimiento, interacción y acciones mediante teclado/ratón.	Personaje responde correctamente.
CU 5	Sistema de combate	Usuario	Combate contra enemigos mediante armas y habilidades.	Daño calculado y estado actualizado.
CU 6	Sistema de habilidades	Usuario	Uso y desbloqueo de habilidades en combate.	Habilidades ejecutadas correctamente.

CU 7	Sistema genético	Usuario	Selección de rutas evolutivas del personaje.	Atributos modificados según evolución.
CU 8	Sistema de núcleo	Usuario	Mejora avanzada de atributos y capacidades.	Potenciaciones aplicadas correctamente.
CU 9	Sistema de niveles	Usuario	Ganancia de experiencia y subida de nivel.	Nivel actualizado y recompensas desbloqueadas.
CU 10	Misiones y progreso	Usuario	Ejecución de misiones principales y secundarias.（aún no se ha implementado）	Progreso de historia actualizado.
CU 11	Inventario	Usuario	Gestión de objetos, armas y recursos.	Inventario sincronizado con backend.
CU 12	Equipamiento	Usuario	Equipar o cambiar armas y objetos.	Estado del personaje actualizado.
CU 13	Sistema de tutorial	Usuario	Introducción a mecánicas básicas del juego.	Progreso del tutorial registrado.
CU 14	Sistema de bloqueo (gate system)	Usuario	Restricción de contenido según progreso.	Acceso desbloqueado progresivamente.
CU 15	Guardado en la nube	Usuario	Persistencia de datos del jugador en backend.	Datos almacenados correctamente.
CU 16	Recuperación de partida	Usuario	Carga del estado anterior del jugador.	Estado restaurado correctamente.
CU 17	Cambios de escena	Usuario	Transición entre mapas y zonas del juego.	Escena cargada sin errores.
CU 18	Interacción con NPC	Usuario	Diálogo e interacción con personajes no jugables.	Eventos y diálogos activados.
CU 19	Sistema de IA de enemigos	Sistema	Control de enemigos mediante FSM/AI.	Enemigos reaccionan correctamente.
CU 20	Obtención de datos estáticos	Sistema	Carga de items, skills, genes desde API.	Datos sincronizados correctamente.

---

5. Modelo de datos o estructura de la información

Esta base de datos gestiona los datos de usuario, habilidades, objetos y estado de inicio de sesión, y es compatible con un modo de juego que incluye funciones como el desarrollo de personajes, la recolección de objetos y el aprendizaje de habilidades.

Las entidades principales son:

- Usuario
- Objetos de usuario
- Habilidades de Usuario
- Objetos
- Habilidades
- Sesiones

Relaciones entre entidades:

- Un usuario puede tener varios objetos y habilidades.
- Los objetos y habilidades de usuario se basan en tablas base correspondientes.
- Los usuarios pueden tener varias sesiones iniciadas.

Tabla de base de datos:

User

La tabla de usuarios gestiona la información básica del usuario:

Almacenar la información básica del usuario y el estado del personaje

- id
- username
- password
- email
- lv (Nivell)
- exp (Puntos de experiencia)
- health
- max_health
- created_at

Sessions:

La tabla de sesiones se utiliza para el estado de inicio de sesión del usuario y la gestión de sesiones:

Los usuarios pueden registrarse, iniciar sesión y cerrar sesión. El sistema registra el token, el período de validez y el estado de verificación de cada inicio de sesión a través de la tabla de Sesiones.

- id
- user_id
- token
- created_at
- expires_at
- is_verified
- last_login

User_items:

Registre los artículos y las cantidades que posee el personaje:

User_items registra los objetos específicos y las cantidades que posee un personaje, lo que permite mecanismos como la aparición y el uso de objetos.

- id
- user_id
- item_id
- qty

Items:

Define los atributos básicos de todos los objetos disponibles en el juego, como el tipo, la rareza y los efectos.

- id
- name
- type
- rarity
- desc
- effect **

User_skills:

Registre las habilidades que el personaje ha aprendido y su nivel actual.

- id
- user_id
- skill_id
- current_level

Skills:

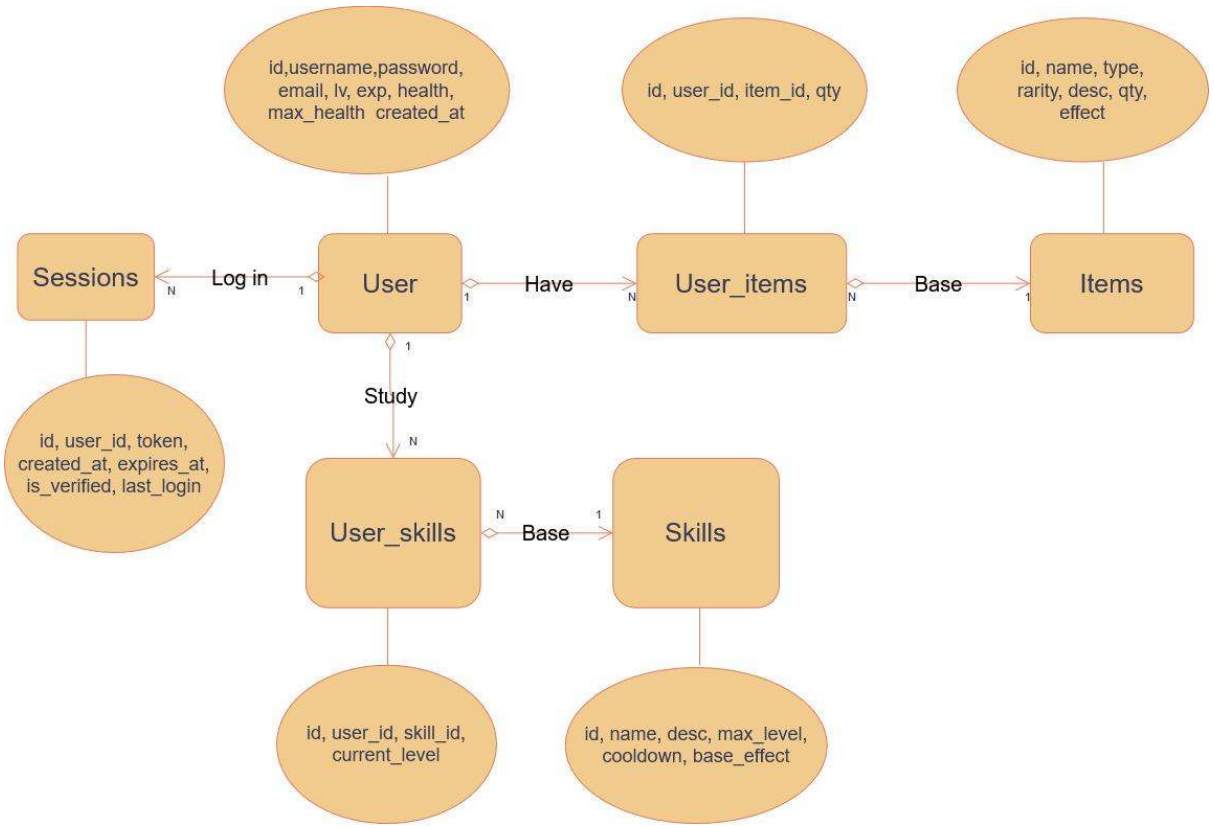
Define la información básica para todas las habilidades, como el nivel máximo, el tiempo de recarga y los efectos básicos.

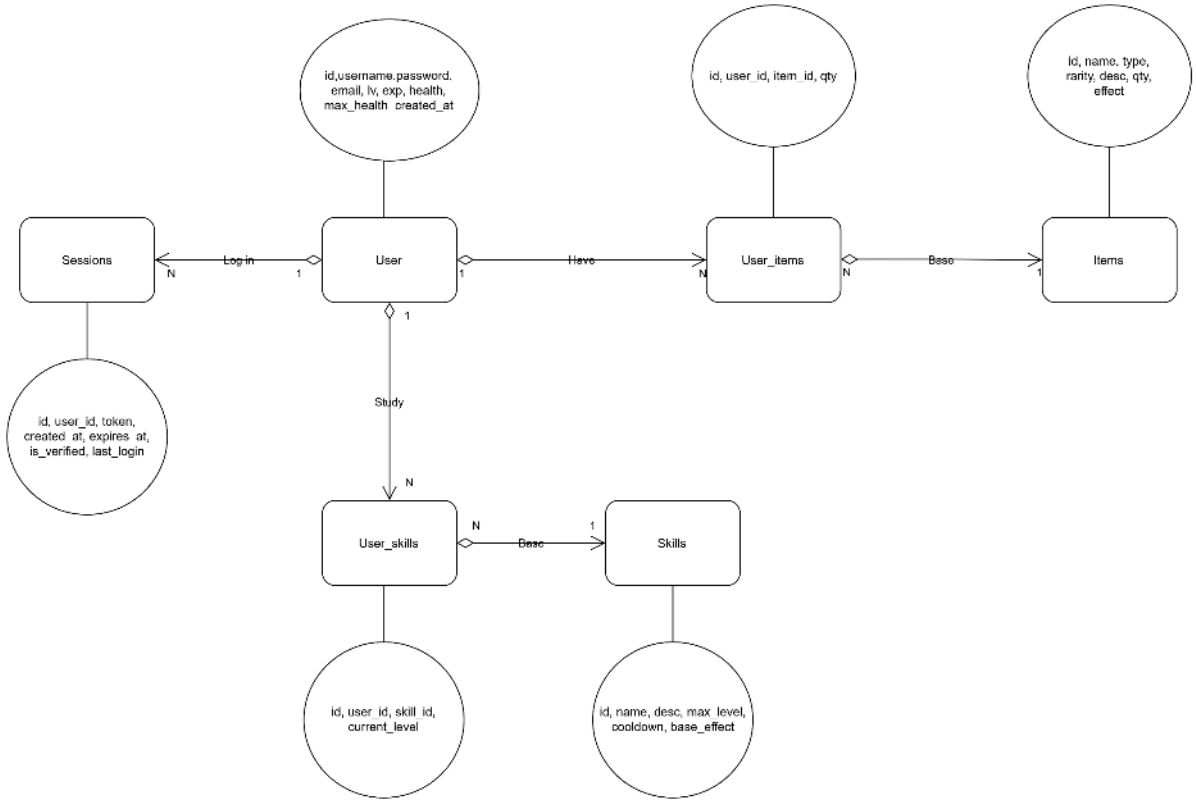
- id
- name
- desc
- max_level
- cooldown
- base_effect

E-R Model

DH - ER Model.jpg

https://drive.google.com/file/d/1EW82VXjNOvwEq62TtCI9BOWupPpJrm47/view?usp=sharing


id,username,password,
email, lv, exp, health,
max_health created_at
id, user_id, item_id, qty
id, name, type,
rarity, desc, qty,
effect
Sessions
Log in
User
Have
User_items
Base
Items
N
1
1
N
N
1
1
Study
N
id, user_id, token,
created_at, expires_at,
is_verified, last_login
User_skills
N
Base
1
Skills
id, user_id, skill_id,
current_level
id, name, desc, max_level,
cooldown, base_effect


id,username,password,
email, lv, exp, health,
max_health created_at
id, user_id, item_id, qty
id, name, type,
rarity, desc, qty,
effect
Sessions
Log in
User
Have
User_items
Base
Items
N
1
1
N
N
1
1
Study
N
id, user_id, token,
created_at, expires_at,
is_verified, last_login
User_skills
N
Base
1
Skills
id, user_id, skill_id,
current_level
id, name, desc, max_level,
cooldown, base_effect

6. Diseño de la interfaz

Dado que algunas interfaces aún no han alcanzado la etapa de diseño formal, algunos elementos de la interfaz de usuario aún se encuentran en borrador. Actualmente, los elementos principales de la interfaz de usuario se dividen en tres categorías:

- La interfaz de habilidades dentro de la interfaz de información del personaje
- Inventario del personaje
- Interfaz de inicio de sesión

La interfaz de inicio de sesión y la interfaz de inventario ya cuentan con prototipos básicos, mientras que la interfaz de habilidades, al estar en una etapa posterior, solo cuenta con un borrador y una imagen ideal generada por Gemini.

https://drive.google.com/file/d/1OZ0qSaryAeWiWbq5kd30cs6VSxx7nmyU/view?usp=sharing

NewInterface.png

https://drive.google.com/drive/folders/1L175udPOTQMCrfdiVpqC8qIuMIHwzw6e?usp=drive_link

Interface

---

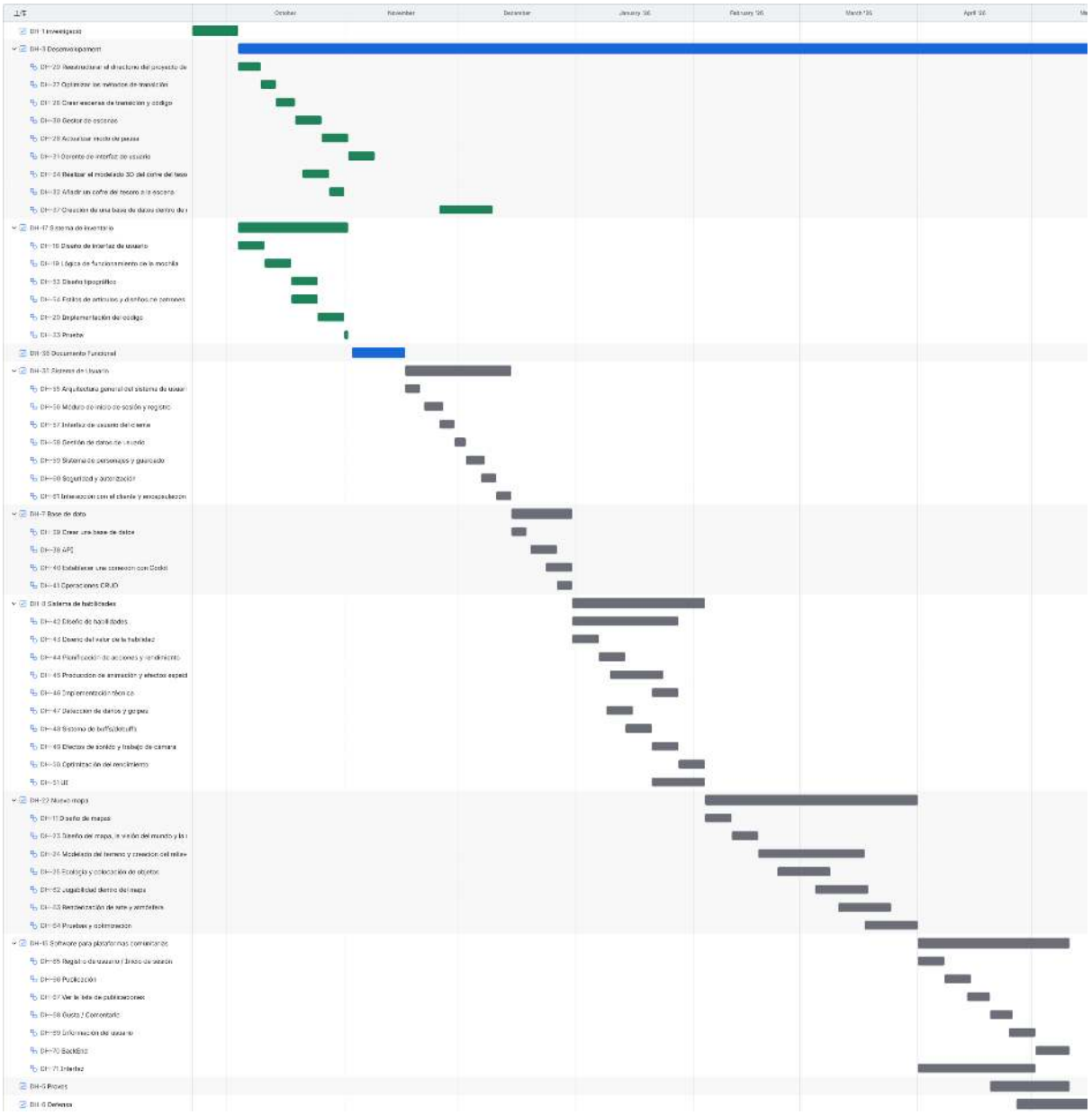
7. Planificación técnica

- Lenguajes y frameworks: Los principales lenguajes utilizados en mis juegos son **Python**, **GDScript** y posiblemente **Go**.
- Base de datos: El marco principal utilizado en mi juego es **Flask**, y **FastAPI** se utiliza como API de **backend**.
- Herramientas de diseño o edición: **Blender, Godot, PhotoPea y Capcut**
- Cronograma (puede incluir un diagrama de Gantt):

Data: 22 / 9 / 2025 - 15 / 5 / 2026

https://drive.google.com/file/d/1G2QqrJDthcSjr9v6cjLBnjqi5gnpDjxP/view?usp=sharing

desastre_humano_2025-10-23_06.26pm.png

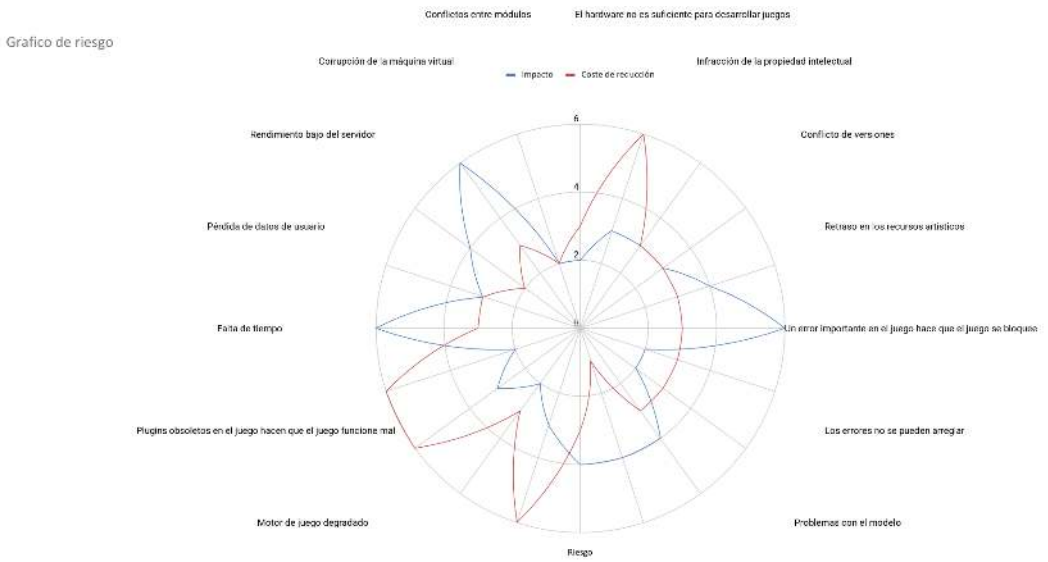


---

8. Análisis de riesgos

https://docs.google.com/spreadsheets/d/1dd_4t7PDwG-cTavYmoze2NgWcWGs4Tp1PhFGWoOBYJ0/edit?gid=421244783#gid=421244783

Riesgos de DH - Jiayi Chen



---

9. Validación y criterios de éxito

Plan del proyecto (si se dispone de tiempo suficiente):

El proyecto tiene previsto añadir los siguientes sistemas y funciones al juego:

- Sistema de habilidades
- Sistema de inventario
- Sistema de usuarios
- Base de datos
- Transición de escenas
- Una aplicación sencilla de foro

Pruebas planificadas para cada módulo y funcionalidad:

Sistema de inventario

- Poder abrir el inventario correctamente
- Poder interactuar con cofres en la escena y obtener objetos
- Poder hacer clic en los objetos
- Poder mostrar la información del objeto
- Poder mover los objetos
  Poder eliminar los objetos
- Poder buscar objetos
- Poder interactuar con los objetos
- Poder ordenar los objetos
- Poder cerrar el inventario correctamente

Sistema de habilidades

- Poder abrir la interfaz de habilidades
- Poder visualizar las habilidades
- Poder usar las habilidades
- Poder mejorar (subir de nivel) las habilidades
- Poder salir de la interfaz de habilidades

Sistema de usuarios

- Poder registrar un usuario
- Poder guardar la información del usuario
- Poder generar credenciales
- Poder iniciar sesión
- Poder vincular la información con la base de datos
- Poder cargar la información del usuario
- Poder guardar nuevamente la información del usuario
- Poder realizar operaciones CRUD en la base de datos

Transición de escenas

- Poder realizar correctamente la transición entre escenas
- Poder ejecutar la animación de transición
- Poder mostrar mensajes o palabras de aviso

Aplicación del foro

- Poder acceder a la aplicación

---

10. Conclusión

El sistema constará de los siguientes módulos principales:

- **Sistema de usuario**: registro de usuario, inicio de sesión, gestión de perfil, guardado del progreso y sincronización en la nube.
- **Sistema de habilidades**: árboles de habilidades personalizables con efectos de habilidades activas y pasivas.
- **Sistema de inventario**: gestión de la adquisición, el uso y el equipamiento de armas, objetos y recursos.
- **Núcleo del juego**: juego de disparos narrativo de mundo abierto que admite misiones no lineales y tramas ramificadas.
- **Capa de interacción de datos**: comunicación en tiempo real entre el backend de Python y el motor Godot para sincronizar los datos de los jugadores (información de usuario, habilidades, objetos, progreso, etc.).

Tecnologías utilizadas :

- **Backend**: Python, con la API construida utilizando Fast API.
- **Base de datos**: PostgreSQL.
- **Lenguaje de programación del juego**: GDScript + Godot.
- **Lenguaje de diseño:** Blender + shader
- **Control de versiones**: Git + GitHub.
- **Implementación y contenedorización**: Docker o empaquetado directo de Godot.

Valor del proyecto

Este proyecto combina una narrativa de mundo abierto con mecánicas de shooter, haciendo hincapié en una narración inmersiva y la progresión de los personajes.

Su valor fundamental reside en:

- Un sistema de progresión del jugador muy flexible con habilidades, elecciones narrativas, equipamiento, etc…
- Una alta integración técnica, con un backend en Python y comunicación en tiempo real con Godot, lo que garantiza datos estables y seguros.
- Una gran escalabilidad, que permite añadir futuras misiones, armas o módulos narrativos sin problemas.

Próximos pasos

Preparar el entorno de desarrollo:

- Configurar la máquina virtual.
- Crear un entorno virtual Python.
- Instalar y configurar la base de datos PostgreSQL.
- Configurar el proyecto Godot y el repositorio de código.

Crear la estructura del módulo y la base de datos para las nuevas funciones:

- Diseñar el modelo de datos básico (usuarios, habilidades, inventario).
- Redactar la documentación de la arquitectura del backend y las especificaciones de la interfaz.
- Establecer la estructura del directorio del proyecto.
- Compilar la documentación del proyecto.

Comenzar la implementación de las funciones de mayor prioridad:

- Sistema de registro y inicio de sesión de usuarios.
- Sincronización de datos fundamentales entre Godot y el backend.
- Implementación de prototipos de los sistemas de inventario y habilidades.