



**Tecnológico  
de Monterrey**

**Análisis de Requerimientos y Actualización de  
Base de Datos**

**TC2005B.501**

Dr. Octavio Navarro Hinojosa

Dr. Esteban Castillo Juarez

Dr. Gilberto Echeverría Furio

David Medina Domínguez A01783155

Juan Pablo Cruz Rodriguez A01783208

Angel Adrian Afonso Castellanos A01782545

Daniel Soult Gómez A01782985

## **Requerimientos de Software y Metodología Ágil:**

Los requerimientos de software son descripciones detalladas de las funciones y características que se necesitan para un sistema o aplicación de software. Estos requerimientos pueden incluir aspectos como la funcionalidad, la usabilidad, el rendimiento, la seguridad y otros aspectos técnicos y no técnicos. La importancia de los requerimientos de software radica en que proporcionan una guía clara y detallada para el diseño, desarrollo, implementación y prueba de un sistema o aplicación de software.

## **Historias de Usuario:**

Las historias de usuario son una técnica de gestión de requisitos en el desarrollo de software que se utiliza para describir las características y funcionalidades que se desean implementar en un producto o sistema. Las historias de usuario son descripciones breves y simples de una funcionalidad desde la perspectiva del usuario, que se escriben en un lenguaje natural y se enfocan en las necesidades y objetivos del usuario.

La importancia de las historias de usuario radica en que ayudan a los equipos de desarrollo de software a comprender las necesidades de los usuarios y a construir un producto que cumpla con sus expectativas. Las historias de usuario proporcionan una descripción clara y concisa de las funcionalidades requeridas, lo que ayuda a los desarrolladores a concentrarse en los detalles relevantes y a evitar el desperdicio de tiempo y esfuerzo en funcionalidades innecesarias.

ID Historia de Usuario	Como<Rol>	Quiero<Eventos>	Para<Objetivo>	Prioridad 1-5	Estimación	Estado
1	Socio ... ▾	que el juego sea RPG	establecer ciertas mecánicas y características	5	240h	Finalizado
2	Socio ... ▾	que el juego sea atractivo	el usuario le de ganas de jugar de nuevo	5	240h	Finalizado

ID Historia de Usuario (Esteban)	Como<Rol>	Quiero<Eventos>	Para<Objetivo>	Prioridad 1-5	Estimación	Estado
1	Produ... ▾	que se use una base de datos relacional	para consultar los datos de forma eficaz y optimizada	5	240h	Finalizado
2	Produ... ▾	se proteja la integridad de los datos	para evitar datos incorrectos, duplicados, faltantes o alterados	5	240h	Finalizado
3	Produ... ▾	usar llaves de optimización	para una búsqueda en la tabla de forma rápida	4	240h	Finalizado
4	Produ... ▾	normalizar la base de datos	reducir la redundancia de los datos y aumentar la integridad de los mismos	4	240h	Finalizado
5	Produ... ▾	utilizar una API	para conectar la página web, el juego y la base de datos	4	240h	Finalizado
6	Produ... ▾	usar datos atómicos para la tablas	desglosar la información a un punto fácil de entender	3	240h	Finalizado
7	Produ... ▾	implementar un nivel introductorio	reconocer los controles o historia del juego	3	240h	Finalizado
8	Produ... ▾	controles o mecánicas intuitivas	para facilitar el aprendizaje del jugador	4	240h	Finalizado
9	Produ... ▾	opción de inicio de sesión	para llevar un registro de los datos del usuario	3	240h	Finalizado

ID Historia de Usuario (Gilberto)	Como<Rol>	Quiero<Eventos>	Para<Objetivo>	Prioridad 1-5	Estimación	Estado
1	Produ... ▾	que el juego tenga un objetivo	para darle una razón al jugador para completar el juego	5	240h	Finalizado
2	Produ... ▾	implementar mecánica de crecimiento	para darle la sensación al jugador de estar mejorando en el juego	4	240h	Finalizado
3	Produ... ▾	usar máquinas de estados para la toma de decisiones	para una mejor visualización de los posibles caminos	4	240h	Finalizado

ID Historia de Usuario (Octavio)	Como<Rol>	Quiero<Eventos>	Para<Objetivo>	Prioridad 1-5	Estimación	Estado
1	Produ... ▾	que la página web sea segura	para mantener los datos del usuario	5	240h	Finalizado
2	Produ... ▾	que tenga un interfaz de usuario intuitiva	para que se le facilite al usuario ubicarse y encontrar cada apartado	3	240h	Finalizado
3	Produ... ▾	que tenga sonido y gráficos de calidad	para una experiencia inmersiva y gratificante	2	240h	Finalizado
4	Produ... ▾	que el juego sea desafiante	para mantener el interés y el compromiso	5	240h	Finalizado

ID Historia de Usuario (Octavio)	Como<Rol>	Quiero<Eventos>	Para<Objetivo>	Prioridad 1-5	Estimación	Estado
			del jugador			
5	Produ...	implementar una función de registro	para guardar la información y la partida del jugador	3	240h	Finalizado

## Product Backlog:

El Product Backlog es una lista ordenada de requerimientos o funcionalidades que se desean incluir en un producto o sistema en desarrollo. El Product Backlog es un componente clave de la metodología ágil de desarrollo de software, y se utiliza para gestionar los requisitos del proyecto y planificar el trabajo a realizar en cada iteración.

Durante un periodo de 10 semanas realizamos algunos ajustes a ciertos requerimientos funcionales y no funcionales, para una mejor comprensión y agilidad del proyecto. A continuación se muestra los requerimientos funcionales y no funcionales que permanecieron intactos o sin ningún cambio:

- **Funcionales**
  - Socio Formador
    - **RPG**
      - Variedad de personajes
      - Sistema de combate
      - Progresion de Personaje
      - Historia y Misiones
      - Sistema de Inventario
      - Jugabilidad Intuitiva
    - **Juego Atractivo**
      - Gráficos de Calidad
      - Sonidos de Calidad
      - Historia y Personajes Interesantes
      - Retos Desafiantes
  - Product Owner Esteban Castillo Juarez
    - **Utilizar una base de datos relacional**
      - Almacenamiento y Organización de los datos
      - Integridad de los datos
      - Seguridad de los datos

- Escalabilidad
- **Usar restricciones de integridad**
  - Acceso y Recuperación de los Datos
  - Llaves Primarias y Foráneas
  - Acciones en Cascada
  - Reglas de Actualización
  - Restricción de Valores de Columnas
- **Normalizar la base de datos**
  - Eliminación de Redundancia
  - Reducción de Errores y Ambigüedad
  - Simplificación de Tablas
  - Flexibilidad y Escalabilidad
  - Aumentar la Capacidad de Análisis
- **Mantener segura la base de datos**
  - Autenticacion y Autorizacion
  - Protección de Datos Confidenciales
  - Registro de Acciones
  - Restricción de Acceso
  - Copias de Seguridad
  - Cumplimiento de Regulaciones
- **Implementar una API**
  - Acceso a Datos y Funcionalidad
  - Interfaz Estandarizada
  - Pruebas Y Monitoreo
  - Documentacion Clara
- **Uso de datos atómicos**
  - Integridad de los Datos
  - Unicidad de los Datos
  - Accesibilidad de los datos
  - Escalabilidad de los Datos
  - Compatibilidad de los Datos
  - Identificación de los Datos
- **Implementación de tutorial**
  - Presentación de los Controles y la Historia
  - Interactividad con el Jugador
  - Duración corta
- **Controles intuitivos**
  - Facilidad de Uso
  - Responsividad Instantánea
  - Accesibilidad Amplia
- **Inicio de sesión**
  - Verificación de la Cuenta
  - Recordar la sesión
  - Restablecer contraseña

- Accesibilidad
- Product Owner Gilberto Echeverria
  - **Juego con objetivo**
    - Reglas del juego
    - Niveles
    - Enemigo u Obstáculos
  - **Crecimiento de personaje**
    - Progresión en el juego
    - Sistema de experiencia
    - Crecimiento de habilidad y atributos
    - Dificultad del juego
  - **Máquinas de estados para decisiones**
    - Estados y Transiciones
    - Eventos
    - Acciones y Condiciones
    - Tolerancia a Fallos
- Product Owner Octavio Navarro
  - **Seguridad de la web**
    - Autenticacion y Autorizacion
    - Encriptación de Datos
    - Prevención de Ataques
    - Monitoreo de Actividades
    - Actualización y Parches de Seguridad
  - **Interfaz de usuario**
    - Diseño Amigable e Intuitivo
    - Navegación Clara
    - Compatibilidad de los Dispositivos
    - Seguridad
  - **Efectos especiales**
    - Variedad de Efectos
    - Calidad de Efectos
    - Coherentes
    - Sincronizados
  - **Juego desafiante**
    - Dificultad escalable
    - Puntos de Control
    - Habilidades Especiales
    - Inteligencia Artificial Desafiante
  - **Función de registro**
    - Información de la Cuenta del Jugador
    - Creación de Perfil o Cuenta
    - Estadísticas y Clasificación

- Características Adicionales
- **No Funcionales**

- Paleta de Colores Oscura
- Presentación Amigable de Controles
- Controles del Teclado
- Historia de los Backrooms
- Una Sola Dificultad
- Diagramas y Esquemas
- Datos guardados en un Servidor en Línea
- Usar Claves Foráneas en cada Tabla
- Música y Ambiente Tétricos
- Pixel Art 16x16
- Creación de cinemáticas
- Temática de horror
- Elección de personajes
- Animaciones
- Escenas de transición
- Guardado de varias partidas

## Sprint Backlog:

El Sprint Backlog es una lista de tareas o actividades específicas que el equipo de desarrollo debe completar durante un Sprint en particular. El Sprint Backlog se crea a partir del Product Backlog y se utiliza para planificar el trabajo que se realizará durante el Sprint. El Sprint Backlog se actualiza diariamente durante el Sprint y se utiliza como una herramienta para el seguimiento del progreso del equipo y para asegurar que se cumplan los objetivos del Sprint.

En el apartado de sprints se mantuvo igual. Sin embargo, debido algunos asuntos relacionados con el tiempo de trabajo decidimos replantear ,durante el transcurso de las semanas, el plazo de los sprints al igual que la cantidad de los mismos:

- **Sprint: Organización Inicial y Creación de Niveles (Introducción, Lobby)**
  - Juan Pablo Cruz Rodríguez
- **Sprint: Creación de Niveles (1 Hospital , 2 Bosque)**
  - David Medina Domínguez
- **Sprint: Ajustes de niveles anteriores**
  - Angel Adrian Alfonso

- **Sprint: Reuniones de metodología SCRUM , Presentaciones y Debugging**
  - Daniel Soult Gomez
- **Sprint: Reuniones Finales , Expo Ingenierías y Entrega de Videojuego**
  - Juan Pablo Cruz Rodríguez

## Trello: Inicial

**Sprint BackLog**

- Organización Inicial y Creación de Niveles (Introducción, Lobby)
- Creación de Niveles (1 Hospital, 2 Bosque)
- Ajustes y Creación de Nivel Final
- Reuniones Finales, Presentaciones y Debugging

**Historia de Usuario**

El juego debe ser un RPG  
El juego debe ser atractivo (Jugabilidad, Originalidad y Vendible)  
El juego debe de producir un sentimiento al jugador de volver a jugar el producto.

**Product BackLog Funcionales**

- Variedad de personajes
- Sistema de combate
- Progresión de personaje
- Historia y misiones
- Sistema de inventario
- Jugabilidad intuitiva
- Graficos de calidad
- Sonido de Calidad

**Product BackLog No Funcionales**

- El juego tiene una paleta de colores oscuras.
- Presentación de controles
- Historia de Backrooms
- Una dificultad desafiante
- Esquemas relación entidad
- Datos guardados en un servidor en línea privado
- Simplificar los elementos en tablas
- Usar llaves foráneas en cada tabla

**Lista de tareas**

Introducción a nuestro juego  
Diseño de Nivel de Introducción y Nivel de Lobby

**En proceso**

Efectos de Sonido

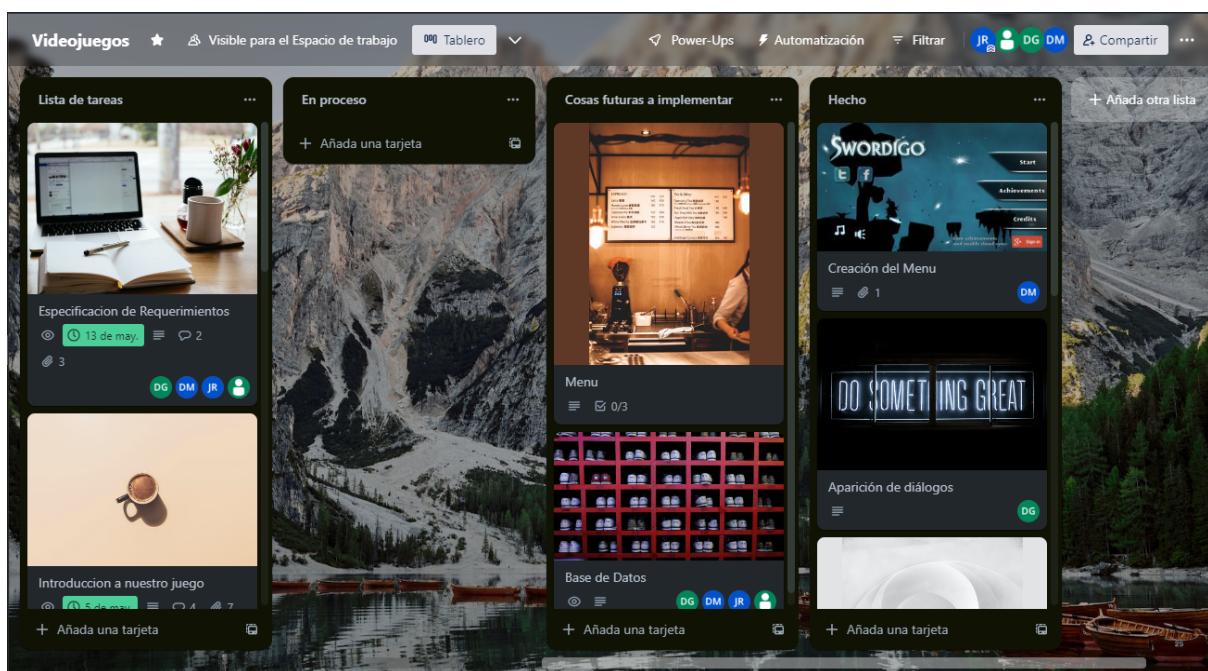
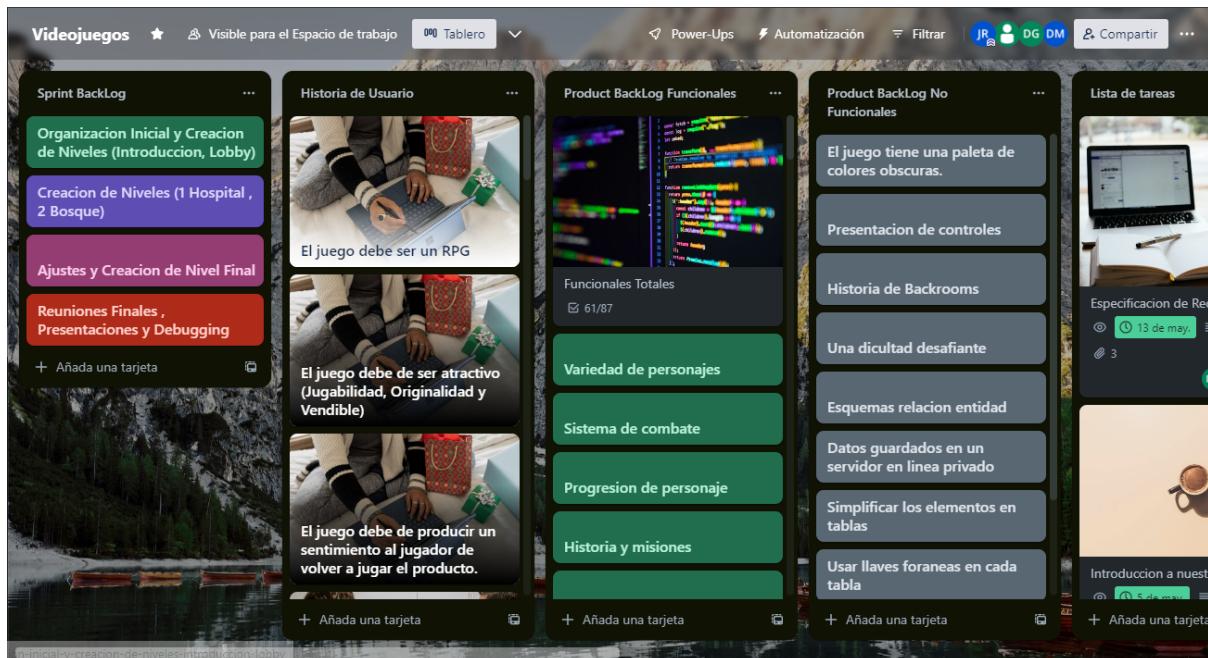
**Cosas futuras a implementar**

Menu  
Base de Datos

**Hecho**

Creación del Menú

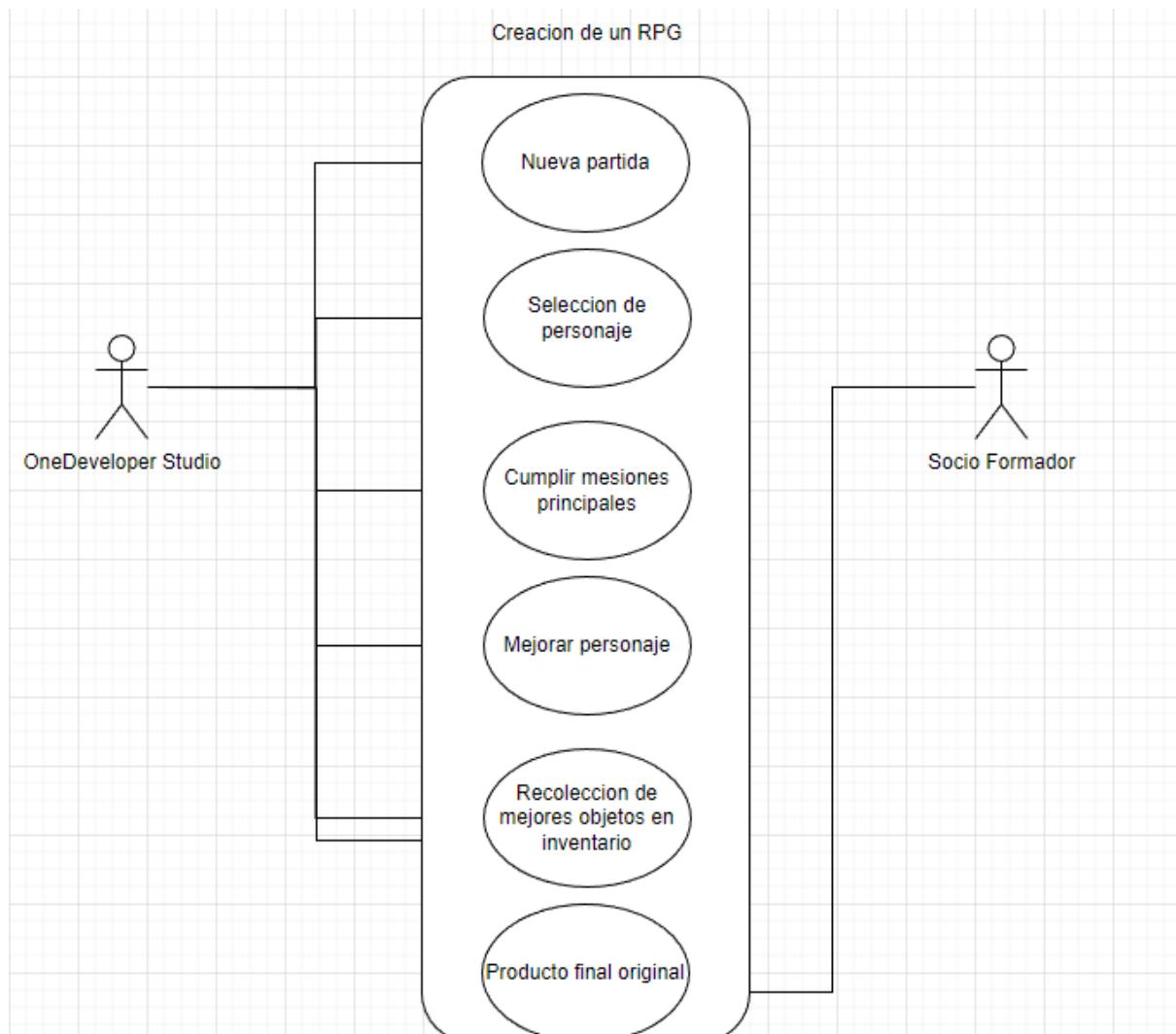
## Trello Final

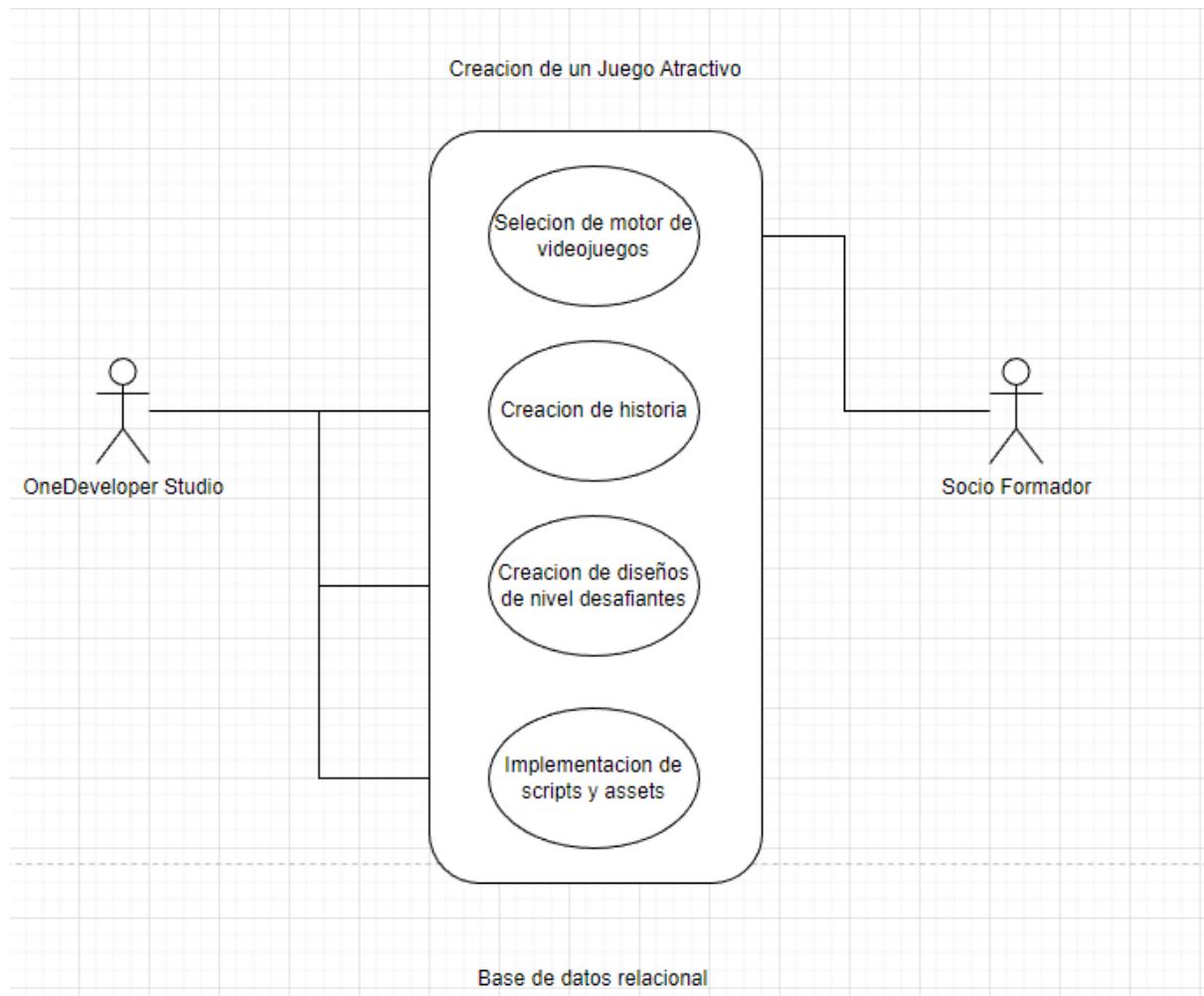


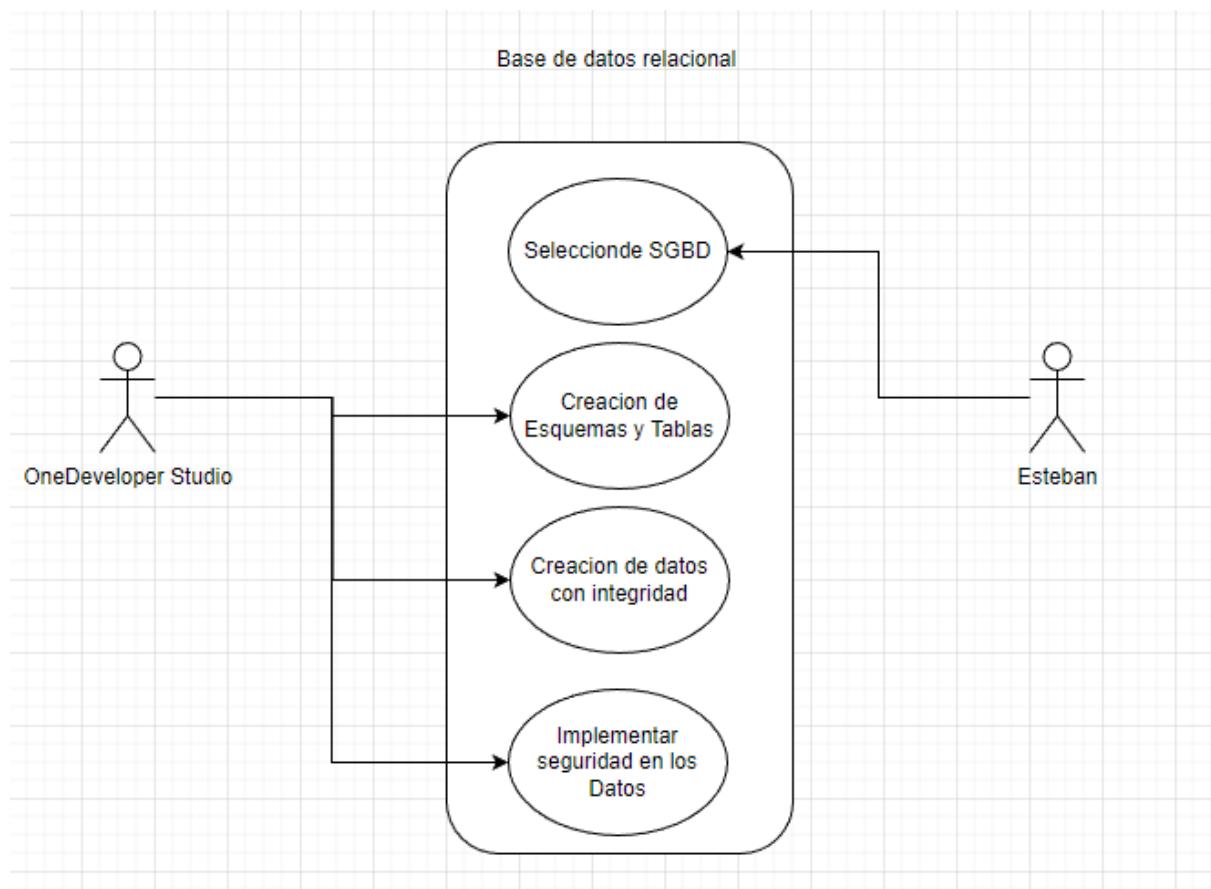
## Trello: Espacio de Metodología Ágil SCRUM

### Diagramas de Casos de Uso:

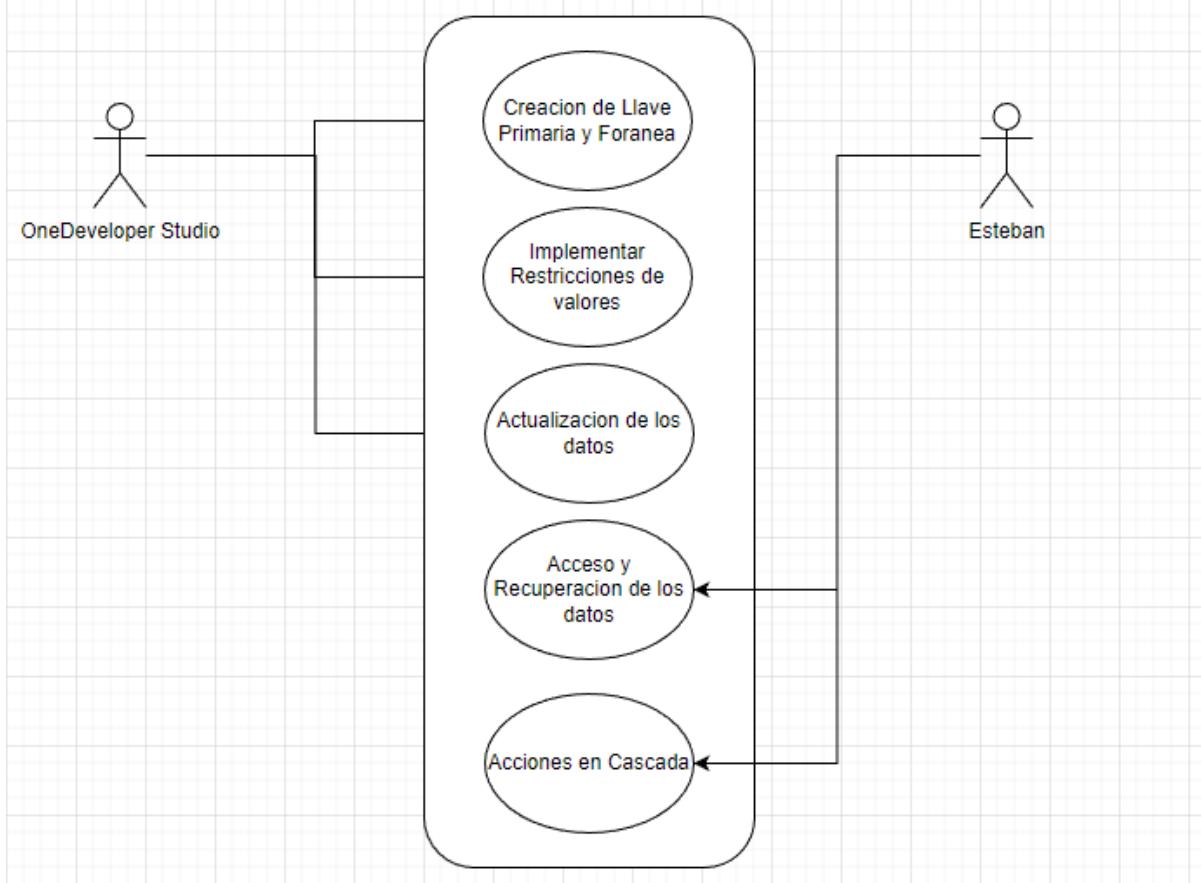
Los diagramas de caso de uso son una herramienta gráfica utilizada en la ingeniería de software para representar las funcionalidades del sistema desde la perspectiva del usuario. Se utilizan para describir cómo los usuarios interactúan con el sistema y para identificar los requisitos funcionales del sistema.



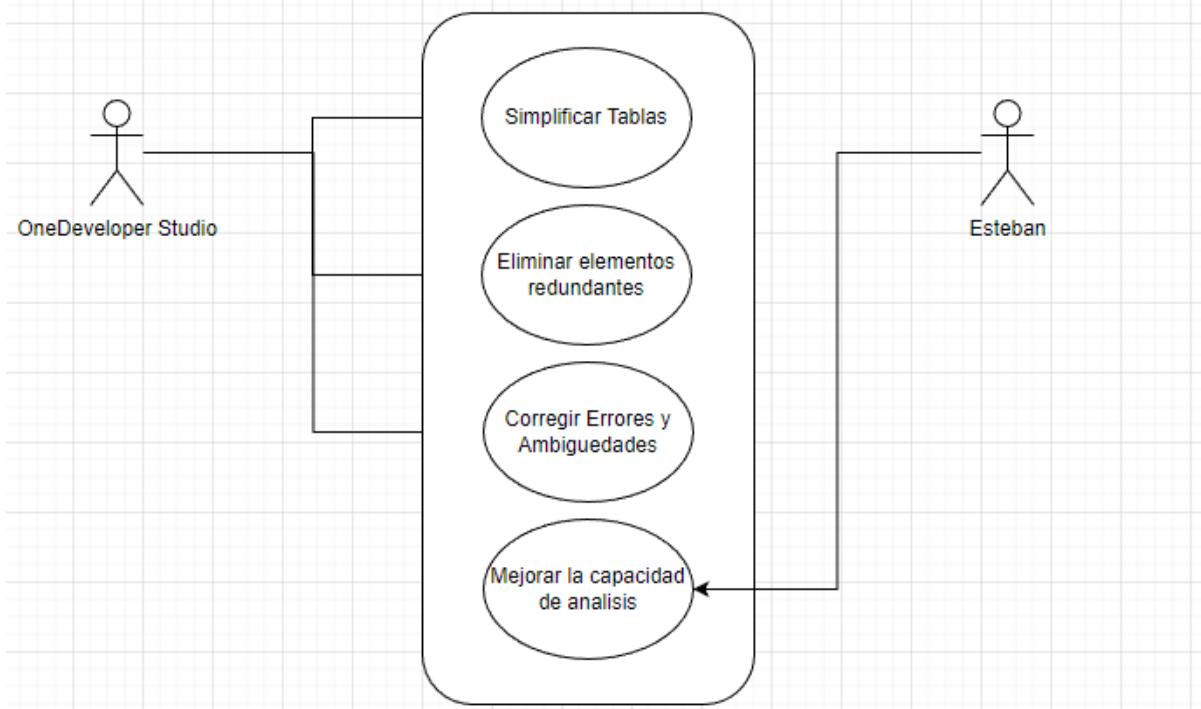




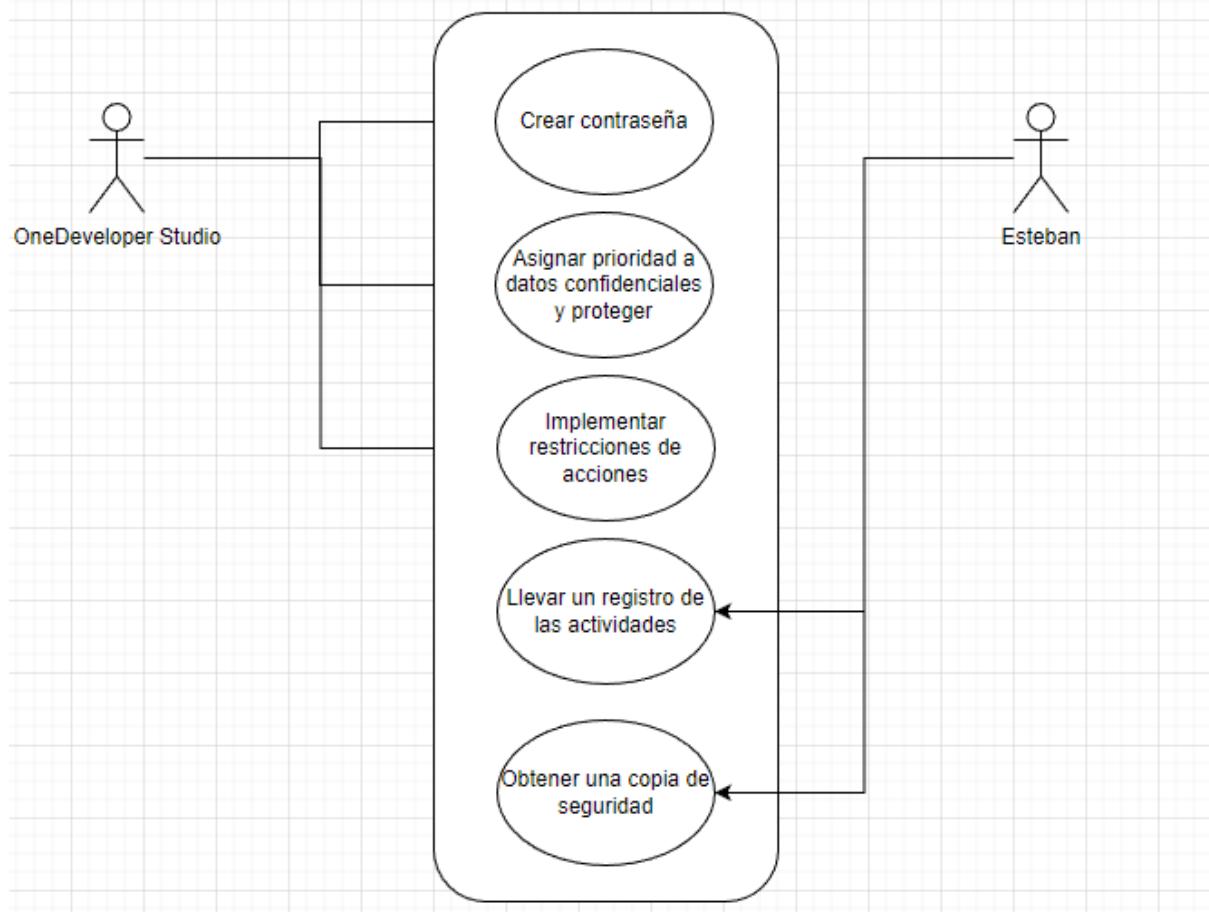
### Implementar Restricciones de Integridad

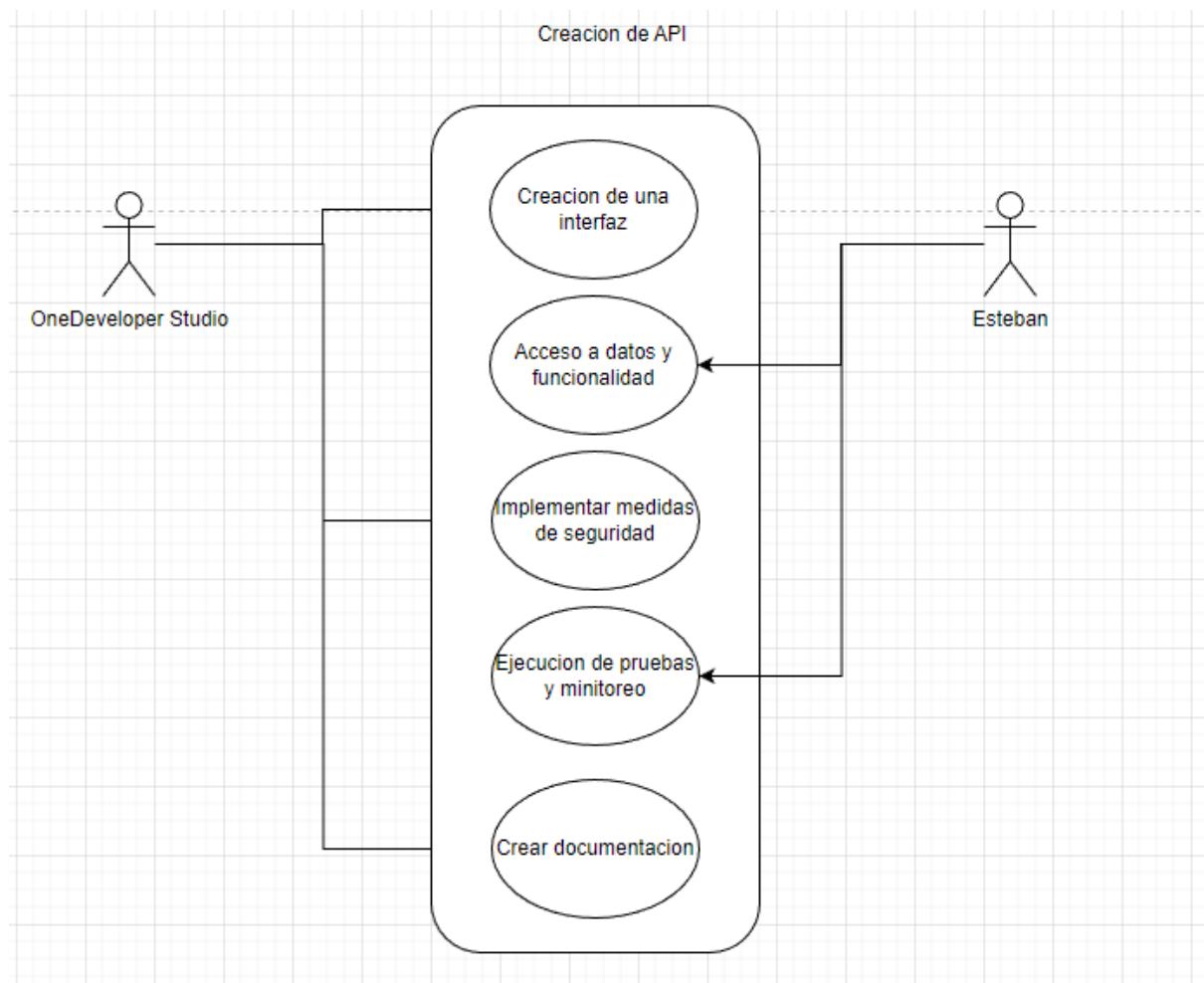


### Normalizacion de la base de datos

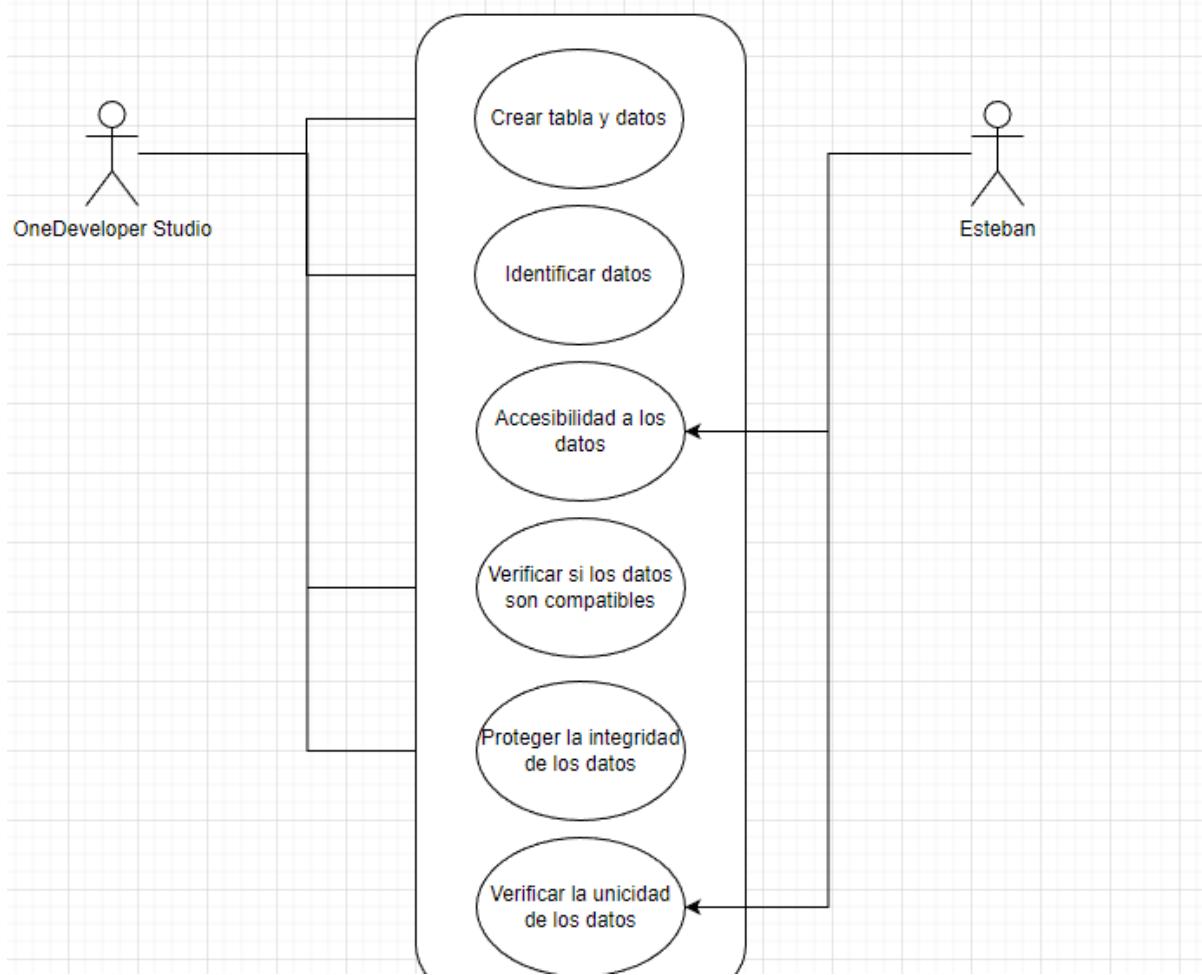


## Seguridad de la base de datos

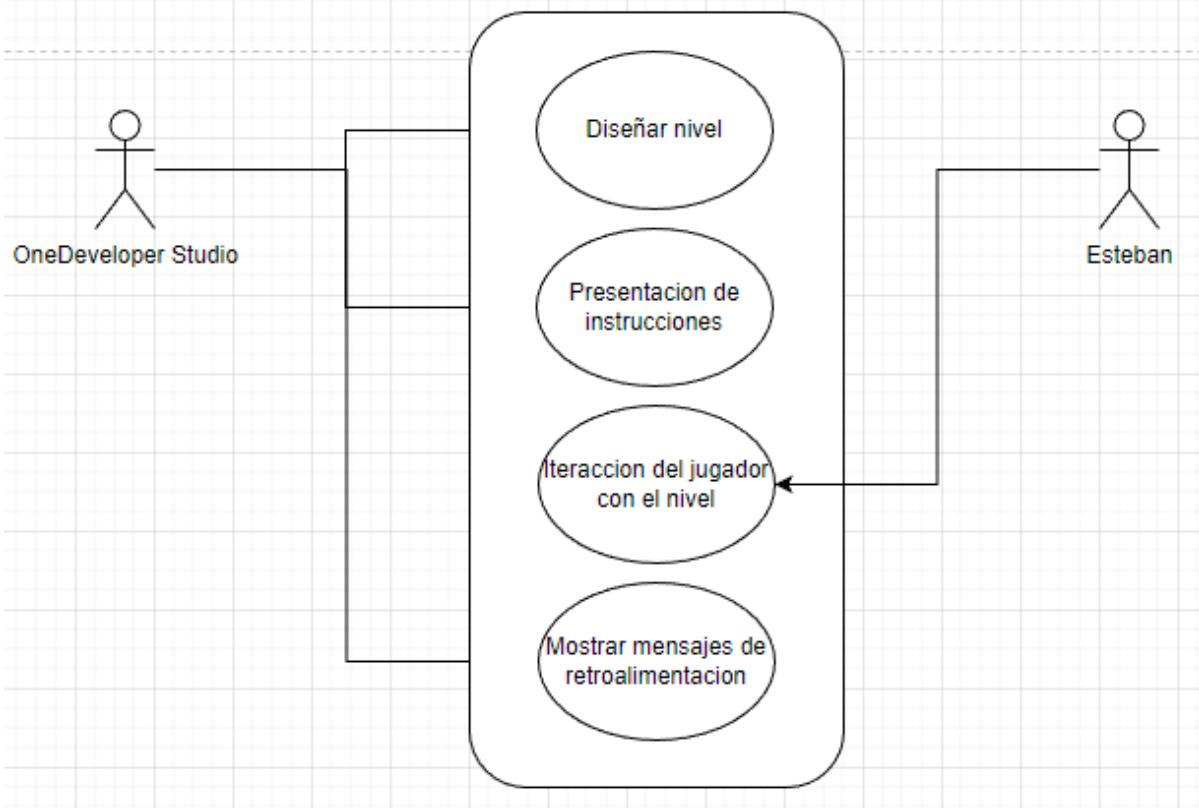




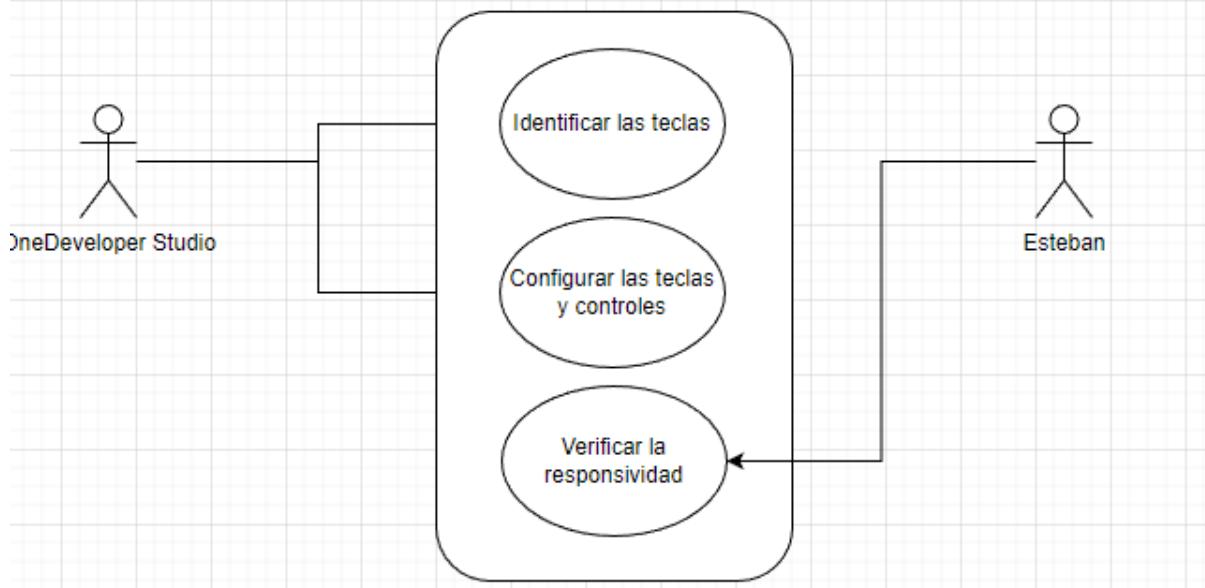
### Creacion de datos atomicos

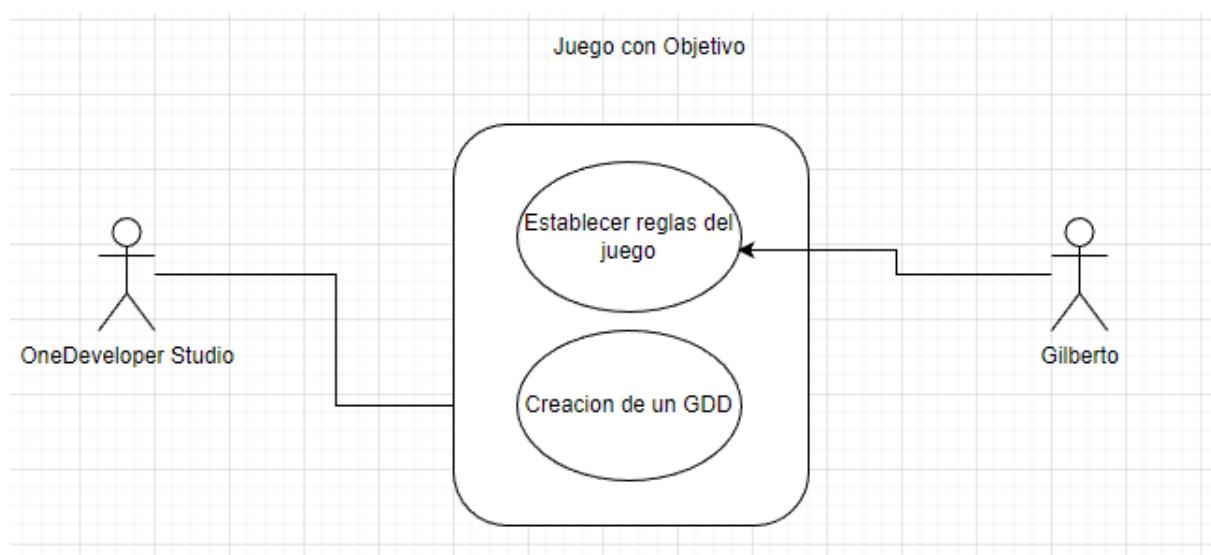
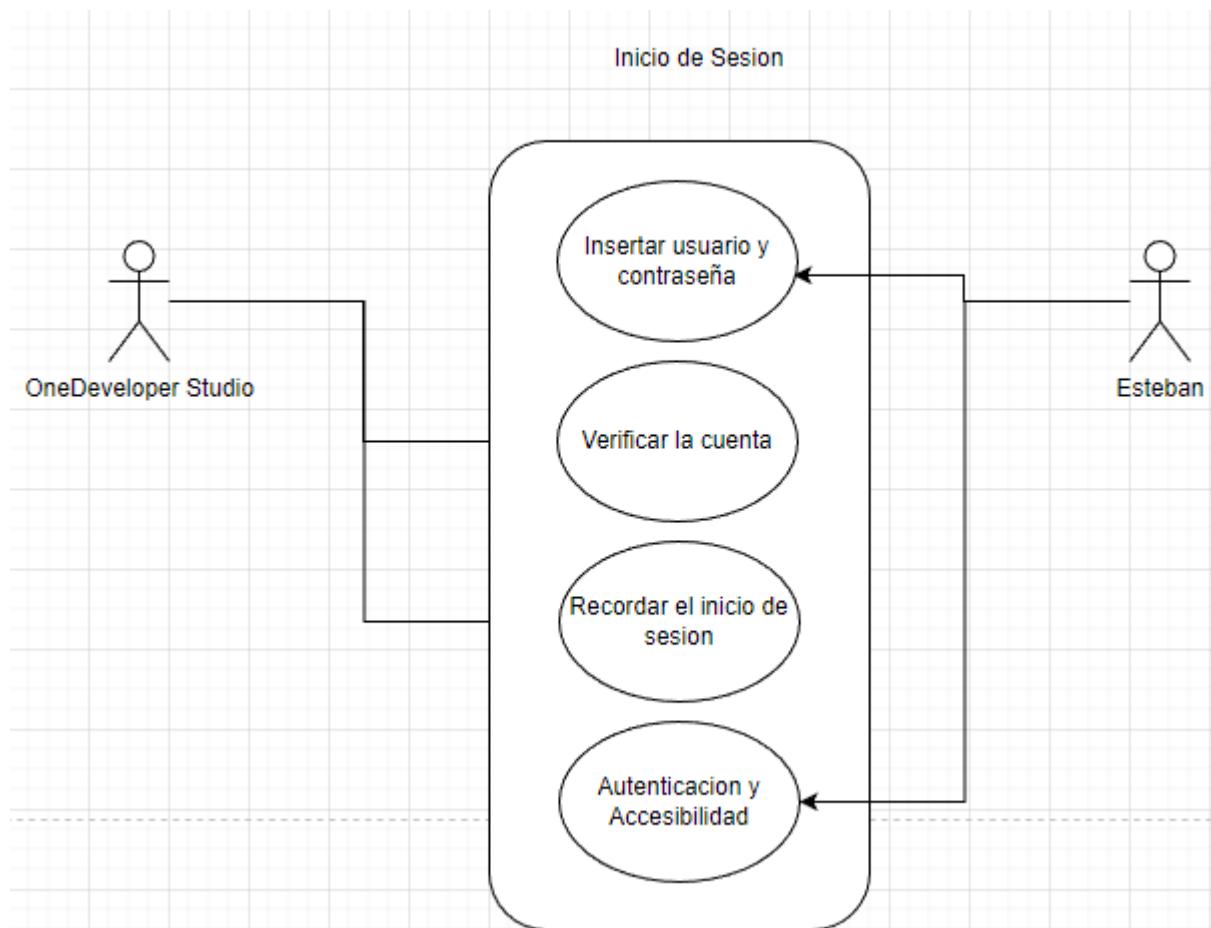


### Creacion de Tutorial

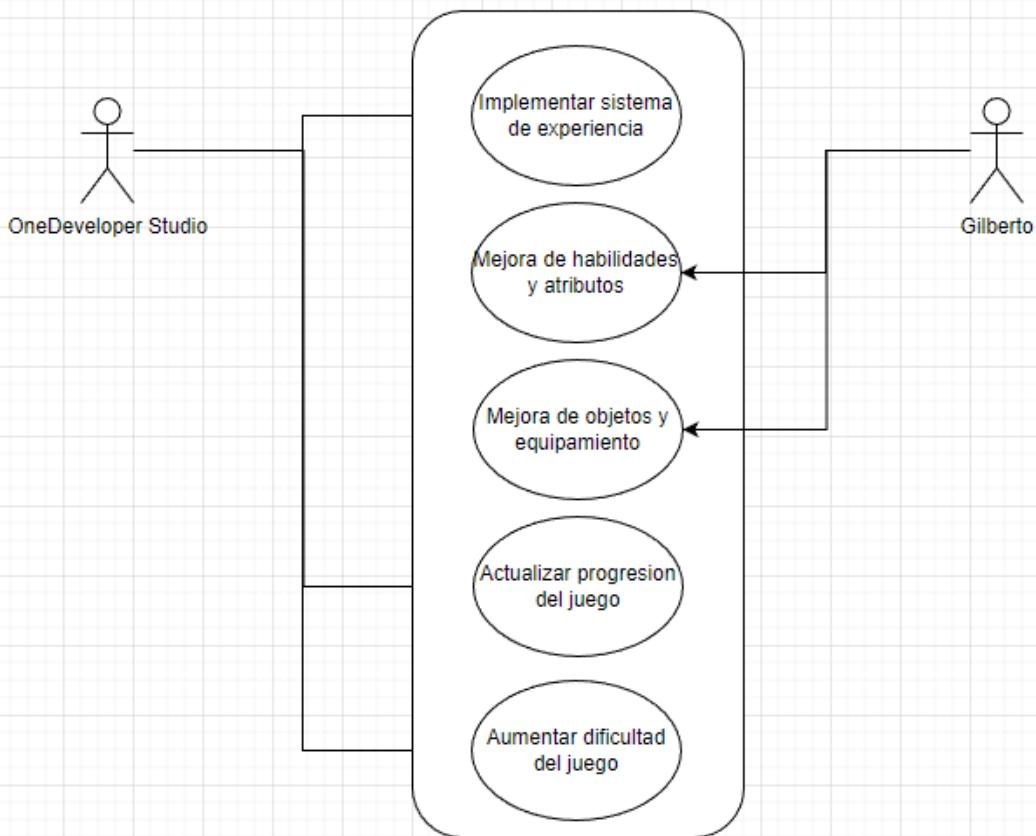


### Implementacion de Controles Intuitivos

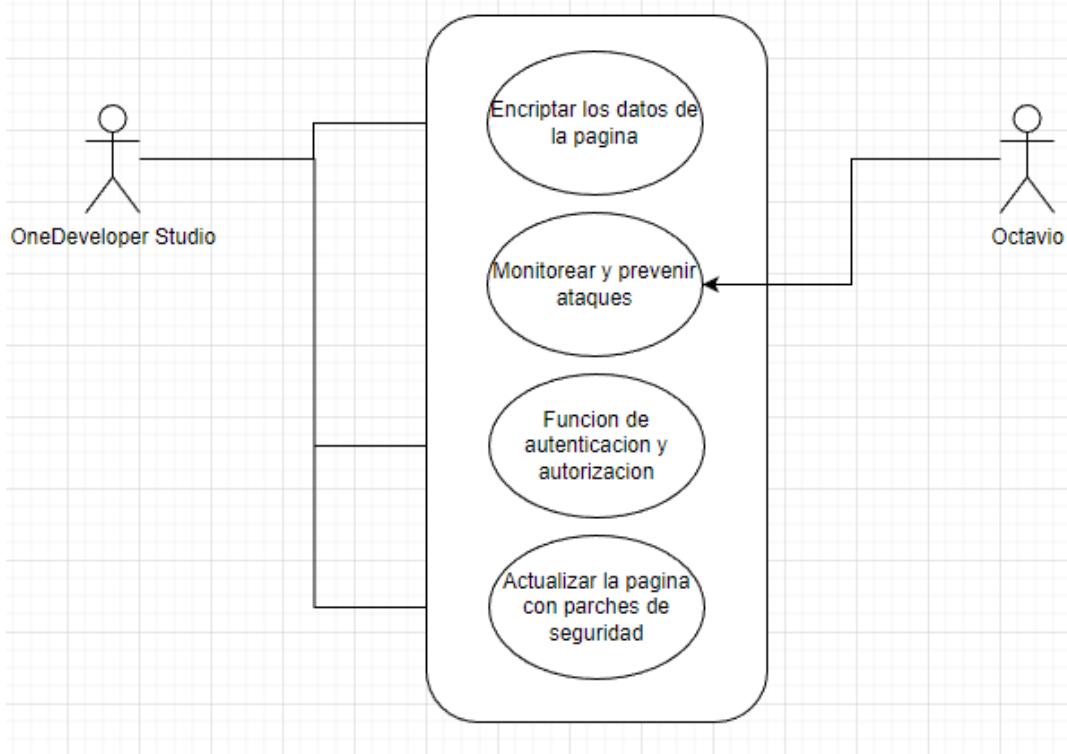


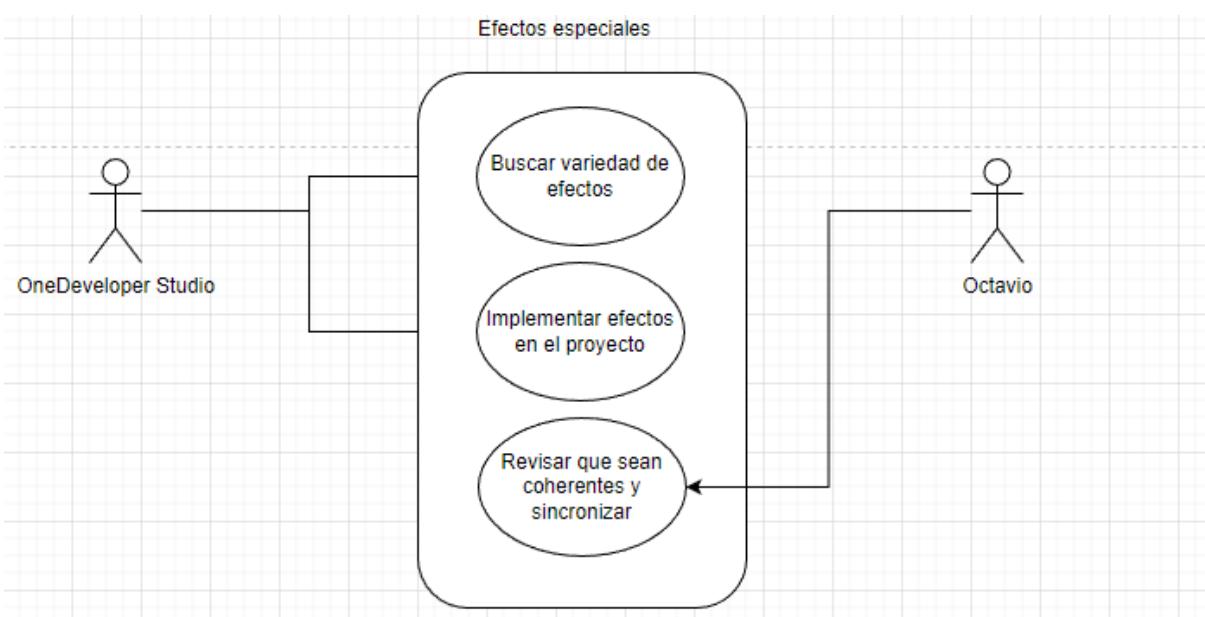
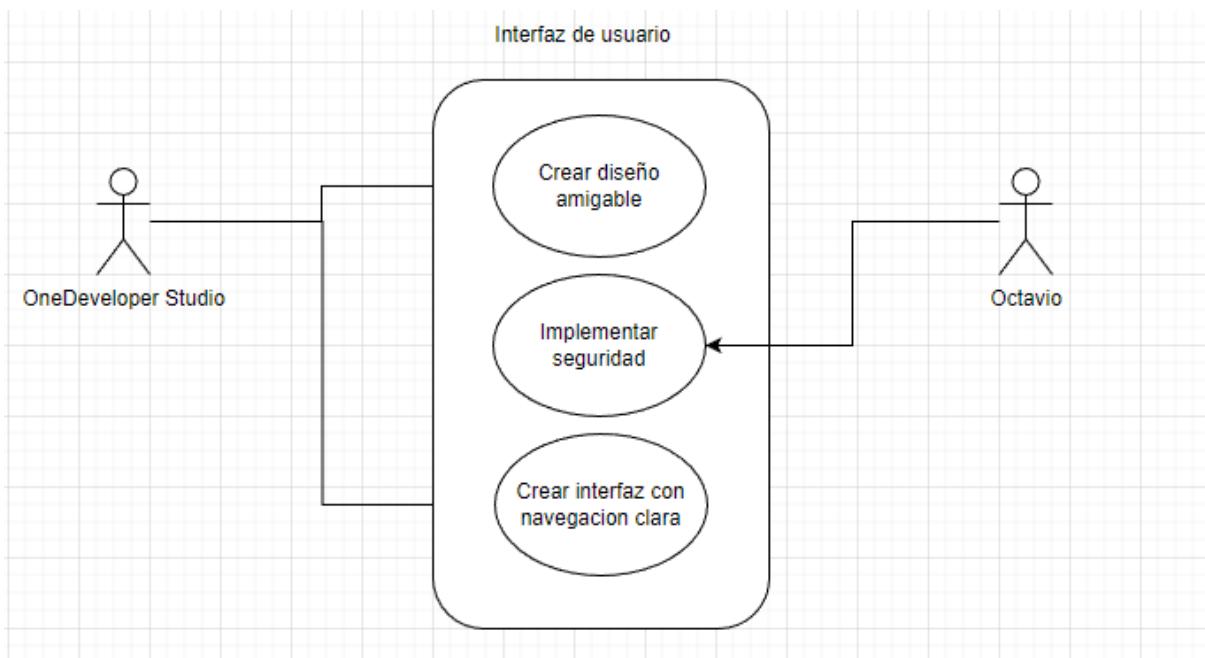


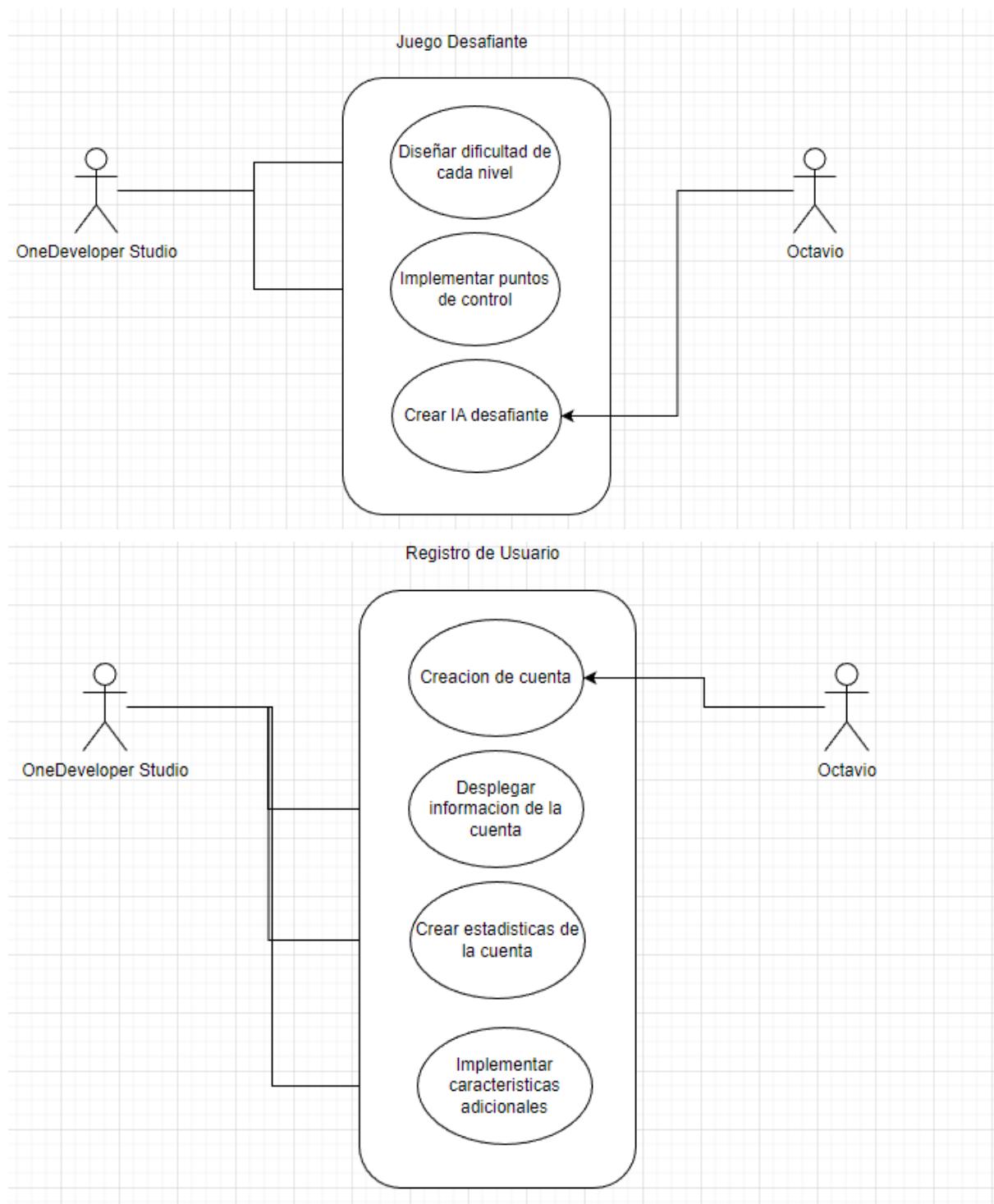
### Mecanica Crecimiento de Personaje



### Seguridad de la web







### Diagramas de caso de uso

### **Tablas Descriptivas:**

Las tablas descriptivas, también conocidas como tablas de casos de uso o tablas de especificación, son herramientas utilizadas en la ingeniería de software para describir de manera estructurada y detallada los casos de uso de un sistema o software. Estas tablas proporcionan información clara y concisa sobre los actores,

los disparadores del caso, los flujos de ejecución y las postcondiciones de cada caso de uso.

<b>Caso de Uso</b>	Inicio de Nueva Partida
<b>Descripción</b>	Este caso de uso describe el inicio de una nueva partida en el videojuego. El jugador comienza una sesión de juego desde cero.
<b>Actores</b>	Jugador
<b>Disparador</b>	El jugador decide iniciar una nueva partida.
<b>Precondiciones</b>	El juego está en modo de inicio y el jugador ha seleccionado la opción de comenzar una nueva partida.
<b>Postcondiciones</b>	Se inicia una nueva partida y se carga el estado inicial del juego.
<b>Flujo de Ejecución:</b>	
1.	El jugador selecciona la opción "Nueva Partida" en el menú principal del juego.
2.	El juego muestra una pantalla de confirmación para asegurarse de que el jugador desea comenzar una nueva partida.
3.	El jugador confirma su decisión y el juego procede a reiniciar todos los elementos necesarios para una nueva partida.
4.	El juego carga el estado inicial, incluyendo la ubicación del jugador, su inventario vacío y cualquier configuración predeterminada necesaria.
5.	El jugador es presentado con una introducción o una secuencia inicial para establecer el contexto de la historia, si corresponde.
6.	El jugador obtiene el control del personaje y puede comenzar a jugar en

	la nueva partida.
7.	El caso de uso finaliza.

<b>Caso de Uso</b>	Selección de Personaje
<b>Descripción</b>	Este caso de uso describe el proceso de selección de personaje al inicio de un videojuego. El jugador elige al personaje que controlará durante el juego.
<b>Actores</b>	Jugador
<b>Disparador</b>	El jugador decide iniciar una nueva partida.
<b>Precondiciones</b>	El juego está en modo de selección de personaje y se han cargado los personajes disponibles..
<b>Postcondiciones</b>	El jugador selecciona un personaje y pasa al siguiente paso del juego.
<b>Flujo de Ejecución:</b>	
1.	El jugador accede a la pantalla de inicio o menú principal del juego.
2.	El juego muestra la opción "Nuevo Juego" o "Iniciar Partida" ..
3.	El jugador selecciona la opción "Nuevo Juego" o "Iniciar Partida".
4.	El juego muestra la pantalla de selección de personaje con una lista de personajes disponibles..
5.	El jugador examina las opciones de personaje y sus características.
6.	El jugador elige un personaje seleccionándolo en la pantalla..
7.	El juego confirma la selección del personaje y muestra un mensaje de confirmación.
8.	El jugador acepta la confirmación y el

	juego guarda la selección del personaje.
9.	El juego carga el escenario o nivel inicial del juego con el personaje seleccionado.
10.	El jugador obtiene el control del personaje y puede comenzar a jugar.
11.	El caso de uso finaliza.

<b>Caso de Uso</b>	Misión Principal
<b>Descripción</b>	Este caso de uso describe el inicio de una misión principal en un videojuego. El jugador se embarca en una tarea específica como parte de la historia principal del juego.
<b>Actores</b>	Jugador, Personajes No Jugadores (PNJ) relacionados con la misión.
<b>Disparador</b>	El jugador alcanza un punto en la historia del juego que activa una nueva misión principal..
<b>Precondiciones</b>	El juego está en modo de misión principal y se ha cargado la misión correspondiente con sus objetivos y personajes involucrados.
<b>Postcondiciones</b>	El jugador comienza la misión principal y se establece el estado de la misión.
<b>Flujo de Ejecución:</b>	
1.	El jugador se encuentra en un área del juego o recibe una comunicación que indica el inicio de una nueva misión principal.
2.	El juego muestra una notificación o mensaje en pantalla que informa al jugador sobre la nueva misión y sus objetivos.
3.	El jugador revisa los detalles de la misión, incluyendo los objetivos, el

	contexto de la misión y las recompensas potenciales.
4.	El jugador acepta iniciar la misión principal y confirma su participación.
5.	El juego actualiza el estado de la misión, marcándose como "activa" y estableciendo los objetivos iniciales.
6.	El jugador recibe indicaciones o pistas sobre la ubicación o acciones requeridas para progresar en la misión principal.
7.	El jugador explora el mundo del juego, interactúa con personajes no jugadores (PNJ) relevantes y cumple los objetivos de la misión principal.
8.	Durante la misión, el jugador puede enfrentar desafíos, resolver acertijos, participar en combates u otras actividades relacionadas con la misión.
9.	El jugador progresá en la misión principal al completar los objetivos intermedios y avanzar hacia el objetivo final.
10.	Una vez que el jugador ha cumplido con éxito los requisitos de la misión principal, el juego muestra una notificación de completado y proporciona las recompensas correspondientes..
11.	El juego actualiza el estado de la misión a "completada" y permite
12.	El caso de uso finaliza.

<b>Caso de Uso</b>	Mejorar al Personaje
<b>Descripción</b>	Este caso de uso describe el proceso de mejora del personaje del jugador en un videojuego. El jugador realiza acciones específicas para aumentar las habilidades, características o niveles

	del personaje.
<b>Actores</b>	Jugador
<b>Disparador</b>	El jugador decide mejorar al personaje después de alcanzar un punto específico en el juego, acumular experiencia, obtener nuevos recursos o cumplir ciertos requisitos.
<b>Precondiciones</b>	El jugador decide mejorar al personaje después de alcanzar un punto específico en el juego, acumular experiencia, obtener nuevos recursos o cumplir ciertos requisitos..
<b>Postcondiciones</b>	El personaje del jugador se mejora según las acciones realizadas, y se actualizan las estadísticas, habilidades o características correspondientes.
<b>Flujo de Ejecución:</b>	
1.	El jugador accede a una pantalla o menú específico para mejorar al personaje.
2.	El juego muestra las opciones y recursos disponibles para la mejora, como puntos de experiencia, habilidades desbloqueables, objetos especiales, etc.
3.	El jugador revisa las opciones y decide cómo desea mejorar al personaje.
4.	El jugador selecciona una opción de mejora, como asignar puntos de experiencia a características específicas, desbloquear una nueva habilidad o equipar un objeto mejorado.
5.	El juego verifica si el jugador cumple los requisitos necesarios para realizar la mejora seleccionada.
6.	Si los requisitos se cumplen, el juego realiza los cambios correspondientes en las estadísticas, habilidades o características del personaje.

7.	El juego muestra una confirmación de la mejora realizada y actualiza la interfaz del jugador para reflejar los cambios..
8.	El jugador puede repetir el proceso de mejora seleccionando nuevas opciones o recursos disponibles.
9.	El jugador puede guardar los cambios realizados en la mejora del personaje.
10.	El jugador puede continuar jugando con el personaje mejorado y aprovechar las ventajas o beneficios obtenidos.
11.	El caso de uso finaliza.

<b>Caso de Uso</b>	Recolección de Objetos
<b>Descripción</b>	Este caso de uso describe el proceso de recolección de objetos por parte del jugador en un videojuego. El jugador busca, recoge y almacena diferentes tipos de objetos durante su progreso en el juego.
<b>Actores</b>	Jugador
<b>Disparador</b>	El jugador encuentra un objeto visible en el entorno del juego o interactúa con un elemento específico que contiene objetos recolectables..
<b>Precondiciones</b>	El juego está en modo de exploración o interacción, y existen objetos recolectables en el entorno del jugador.
<b>Postcondiciones</b>	El jugador recolecta el objeto, que se agrega a su inventario o se utiliza de alguna manera en el juego.
<b>Flujo de Ejecución:</b>	
1.	El jugador explora el entorno del juego o interactúa con elementos específicos, como cofres, cajas, enemigos derrotados, etc.

2.	El juego muestra un objeto recolectable en el entorno o en la recompensa obtenida al interactuar con el elemento.
3.	El jugador decide recolectar el objeto y realiza la acción correspondiente, como acercarse al objeto o interactuar con él.
4.	El juego verifica si el jugador cumple los requisitos necesarios para recolectar el objeto.
5.	Si los requisitos se cumplen, el juego realiza los cambios necesarios en el inventario del jugador y agrega el objeto recolectado.
6.	El juego muestra una confirmación de la recolección del objeto y actualiza la interfaz del jugador para reflejar los cambios en el inventario.
7.	El jugador puede repetir el proceso de recolección al encontrar nuevos objetos en el entorno del juego o al interactuar con elementos adicionales.
8.	El jugador puede utilizar los objetos recolectados de diversas formas, como equiparlos, utilizarlos como herramientas o recursos, combinarlos con otros objetos, etc.
9.	El jugador puede gestionar su inventario, descartando objetos no deseados, vendiéndolos o intercambiándose con otros personajes no jugadores (PNJ) en el juego.
10.	El caso de uso finaliza.

<b>Caso de Uso</b>	Entrega Final
<b>Descripción</b>	Este caso de uso describe el proceso de entrega final de un videojuego. El equipo de desarrollo completa y entrega el producto final del videojuego a los usuarios o clientes.

<b>Actores</b>	Equipo de Desarrollo, Usuarios o Clientes
<b>Disparador</b>	El desarrollo del videojuego ha llegado a su etapa final y está listo para ser lanzado o entregado a los usuarios o clientes.
<b>Precondiciones</b>	El desarrollo del videojuego ha alcanzado su estado final, todas las características y funcionalidades han sido implementadas y se han realizado las pruebas y correcciones necesarias.
<b>Postcondiciones</b>	El videojuego finalizado se entrega a los usuarios o clientes para su uso o distribución.
<b>Flujo de Ejecución:</b>	
1.	El equipo de desarrollo realiza las últimas pruebas y correcciones para garantizar que el videojuego esté libre de errores y cumpla con los requisitos establecidos.
2.	El equipo de desarrollo prepara los elementos necesarios para la entrega final, como la compilación final del videojuego, los recursos adicionales, los documentos de usuario y cualquier otro componente requerido.
3.	El equipo de desarrollo crea una versión final del videojuego que se ajusta a los estándares y formatos requeridos para su distribución o uso.
4.	El equipo de desarrollo genera la documentación necesaria, como manuales de usuario, guías de instalación, notas de lanzamiento, etc.
5.	El equipo de desarrollo realiza una revisión final de todos los elementos y documentos para asegurarse de que estén completos y listos para su entrega.
6.	El videojuego finalizado y los documentos relacionados se entregan a

	los usuarios o clientes según los acuerdos establecidos, ya sea a través de descarga digital, distribución física o mediante una plataforma específica.
7.	Los usuarios o clientes reciben el videojuego finalizado y los documentos asociados, y procede a instalarlo, utilizarlo o distribuirlo según sea necesario.
8.	El equipo de desarrollo puede realizar un seguimiento posterior a la entrega para recopilar comentarios, realizar actualizaciones o proporcionar soporte adicional según sea necesario.
9.	El caso de uso finaliza.

**Caso de Uso:** Seleccionar un Motor de Creación de Videojuegos

**Descripción:** Este caso de uso describe el proceso de selección de un motor de creación de videojuegos para el desarrollo de un nuevo proyecto. El equipo de desarrollo evalúa diferentes motores y toma una decisión basada en los requisitos y objetivos del proyecto.

**Actores:** Equipo de Desarrollo

**Disparador:** El equipo de desarrollo inicia un nuevo proyecto de videojuego y necesita seleccionar un motor de creación adecuado para su desarrollo.

**Precondiciones :**El equipo de desarrollo ha definido los requisitos y objetivos del proyecto.

**Postcondiciones :**El equipo de desarrollo ha seleccionado un motor de creación de videojuegos para el proyecto.

| Flujo de Ejecución | |

| 1. El equipo de desarrollo investiga y recopila información sobre diferentes motores de creación de videojuegos disponibles en el mercado. |

| 2. El equipo de desarrollo define los requisitos técnicos y funcionales del proyecto, incluyendo la plataforma de destino, el tipo de juego, las capacidades requeridas, el presupuesto, etc. |

| 3. El equipo de desarrollo compara los motores de creación de videojuegos disponibles en función de los requisitos y objetivos establecidos. |

| 4. El equipo de desarrollo evalúa las características, funcionalidades, facilidad de uso, documentación, soporte de la comunidad y otros factores relevantes de cada motor. |

| 5. El equipo de desarrollo discute y analiza las opciones disponibles, considerando los pros y contras de cada motor en relación con los requisitos del proyecto. |

| 6. El equipo de desarrollo realiza pruebas de concepto o prototipos utilizando los motores de creación de videojuegos seleccionados, para evaluar su idoneidad y rendimiento en función de los requisitos del proyecto. |

| 7. El equipo de desarrollo toma una decisión final sobre el motor de creación de videojuegos que

mejor se adapte a los requisitos y objetivos del proyecto.
8. El equipo de desarrollo documenta la elección del motor de creación de videojuegos y los factores que influyeron en la decisión.
9. El equipo de desarrollo comunica la selección del motor de creación de videojuegos a todos los miembros del equipo y a otras partes interesadas relevantes.
10. El equipo de desarrollo comienza a trabajar con el motor de creación de videojuegos seleccionado, iniciando el proceso de desarrollo del proyecto.
11. El caso de uso finaliza.

Caso de Uso: Creación de la Historia
Descripción: Este caso de uso describe el proceso de creación de la historia de un videojuego. Incluye la definición de los actores involucrados, los disparadores que inician la historia y el flujo de ejecución para el desarrollo de la trama.
Actores: Diseñador de la historia, Diseñador de niveles, Artista gráfico, Músico
Disparador: Inicio del desarrollo del videojuego
Precondiciones: Se han definido los objetivos y el concepto general del videojuego.
Postcondiciones: Se ha creado la estructura de la historia y los elementos narrativos del videojuego.
Flujo de Ejecución:
1. El equipo de desarrollo se reúne para discutir y definir los elementos clave de la historia, como el género, la ambientación y los personajes principales.
2. El diseñador de la historia trabaja en la creación del argumento general del videojuego, estableciendo los puntos de inicio y fin, así como los eventos principales que sucederán durante el juego.
3. El diseñador de la historia define los actores involucrados en la trama, incluyendo los protagonistas, antagonistas y personajes secundarios.
4. El diseñador de la historia crea los disparadores que iniciarán la historia, como eventos específicos, interacciones del jugador o desbloqueo de niveles.
5. El diseñador de la historia trabaja en el desarrollo de la estructura narrativa, incluyendo la división en niveles, misiones o capítulos, y la progresión de la trama a lo largo del juego.
6. El diseñador de niveles colabora con el diseñador de la historia para crear niveles o escenarios que se ajusten a la trama y promuevan el desarrollo de la historia.
7. El artista gráfico y el músico colaboran en la creación de los elementos visuales y sonoros que complementen la historia, como ilustraciones, animaciones y música de fondo.
8. El equipo de desarrollo realiza revisiones y ajustes en la historia, incorporando comentarios y sugerencias para mejorar la narrativa y el flujo del juego.
9. Una vez finalizada la creación de la historia, se integra en el videojuego, asegurándose de que los eventos, diálogos y elementos narrativos se ejecuten correctamente durante el juego.
10. El caso de uso finaliza.

Caso de Uso: Diseño de Niveles
--------------------------------

Descripción: Este caso de uso describe el proceso de diseño de niveles para un videojuego. Incluye la creación de escenarios jugables, colocación de elementos interactivos y configuración de desafíos.
Actores: Diseñador de niveles, Artista gráfico, Músico
Disparador: Inicio del desarrollo del videojuego
Precondiciones: Se han definido los objetivos y la estructura de la historia del videojuego.
Postcondiciones: Se ha creado un nivel jugable con elementos interactivos y desafíos.
Flujo de Ejecución:
1. El diseñador de niveles revisa los conceptos y requisitos del videojuego, comprendiendo el género, la mecánica y las limitaciones técnicas.
2. El diseñador de niveles crea un boceto o plano inicial del nivel, definiendo la disposición general del escenario y las áreas jugables.
3. El diseñador de niveles coloca los elementos ambientales, como terrenos, plataformas, obstáculos y elementos decorativos, para crear el entorno visual del nivel.
4. El diseñador de niveles incorpora elementos interactivos, como interruptores, puertas, enemigos y objetos colecciónables, para proporcionar desafíos y mecánicas de juego.
5. El diseñador de niveles ajusta el equilibrio del nivel, asegurando que los desafíos sean adecuados para el jugador y promuevan una experiencia gratificante.
6. El artista gráfico trabaja en la creación de los elementos visuales del nivel, como texturas, modelos 3D y efectos visuales, para mejorar la estética y el ambiente del entorno.
7. El músico colabora en la creación de la música y los efectos de sonido que complementen la experiencia del nivel, aportando inmersión y emociones al jugador.
8. El equipo de desarrollo realiza pruebas y ajustes en el nivel, iterando sobre el diseño y solucionando problemas de jugabilidad o rendimiento.
9. Una vez finalizado el diseño del nivel, se integra en el videojuego, asegurándose de que la transición entre niveles sea fluida y se mantenga la coherencia de la historia.
10. El caso de uso finaliza.

Caso de Uso: Implementación de Assets y Scripts
Descripción: Este caso de uso describe el proceso de implementación de assets (recursos visuales y sonoros) y scripts (código) en un videojuego. Incluye la integración de elementos gráficos, sonoros y lógicos para su correcto funcionamiento en el juego.
Actores: Programador, Artista gráfico, Músico
Disparador: Finalización del diseño de niveles
Precondiciones: Se han diseñado los niveles y se han definido los assets y scripts necesarios.
Postcondiciones: Los assets y scripts se han integrado en el videojuego, funcionando correctamente en su contexto correspondiente.
Flujo de Ejecución:
1. El programador revisa los requisitos y especificaciones técnicas del videojuego, comprendiendo las plataformas de destino y las capacidades del motor o entorno de desarrollo.
2. El artista gráfico entrega los assets visuales al programador, como texturas, modelos 3D, animaciones y efectos visuales, en el formato adecuado para su implementación.

3. El músico entrega los archivos de audio al programador, como música de fondo, efectos de sonido y diálogos, en el formato adecuado para su implementación.
4. El programador integra los assets en el videojuego, asignándoles a los objetos, personajes o escenarios correspondientes según el diseño de niveles.
5. El programador implementa los scripts necesarios para controlar el comportamiento de los objetos, la lógica del juego y las interacciones entre elementos, utilizando el lenguaje de programación adecuado.
6. El programador realiza pruebas y ajustes en la implementación de assets y scripts, solucionando errores o inconsistencias en el funcionamiento del juego.
7. El artista gráfico y el músico colaboran con el programador para realizar mejoras visuales o sonoras, como optimización de rendimiento, ajustes de volumen o efectos especiales.
8. El equipo de desarrollo realiza pruebas exhaustivas del videojuego, verificando que los assets y scripts se integren correctamente y funcionen de manera esperada.
9. Una vez finalizada la implementación de assets y scripts, se procede a la fase de refinamiento y pulido del juego, ajustando detalles y mejorando la experiencia del jugador.
10. El caso de uso finaliza.

#### Caso de Uso: Selección de un SGBD

**Descripción:** Este caso de uso describe el proceso de selección de un Sistema de Gestión de Bases de Datos (SGBD) adecuado para un proyecto o aplicación. Incluye la evaluación de requisitos, características y opciones disponibles para determinar el SGBD más apropiado.

**Actores:** Equipo de desarrollo, Administrador de bases de datos

**Disparador:** Inicio del proyecto o necesidad de una nueva base de datos

**Precondiciones:** Se han identificado los requisitos de almacenamiento y manipulación de datos del proyecto.

**Postcondiciones:** Se ha seleccionado un SGBD apropiado para el proyecto.

#### Flujo de Ejecución:

1. El equipo de desarrollo analiza los requisitos del proyecto en términos de volumen de datos, rendimiento, disponibilidad, escalabilidad, seguridad y características específicas.
2. El equipo de desarrollo investiga y evalúa diferentes SGBD disponibles en el mercado, considerando factores como el modelo de datos, el lenguaje de consulta, la compatibilidad con la plataforma de desarrollo y la comunidad de soporte.
3. El equipo de desarrollo compara las características y funcionalidades de los SGBD considerados, teniendo en cuenta aspectos como la escalabilidad, la seguridad, la replicación de datos y el rendimiento.
4. El equipo de desarrollo consulta con el administrador de bases de datos para obtener su opinión y recomendaciones sobre los SGBD evaluados.
5. Basándose en el análisis y las evaluaciones, el equipo de desarrollo selecciona el SGBD más adecuado para el proyecto, teniendo en cuenta los requisitos y restricciones específicas.
6. El equipo de desarrollo inicia la implementación y configuración del SGBD seleccionado para su integración en el proyecto.
7. El caso de uso finaliza.

### Caso de Uso: Creación de Esquemas y Tablas

Descripción: Este caso de uso describe el proceso de creación de esquemas y tablas en un SGBD. Incluye la definición de la estructura de la base de datos y la creación de tablas para almacenar los datos de manera organizada.

Actores: Administrador de bases de datos

Disparador: Configuración inicial de la base de datos o necesidad de nuevas tablas

Precondiciones: El SGBD ha sido seleccionado y está instalado y configurado correctamente.

Postcondiciones: Se han creado los esquemas y tablas necesarios en la base de datos.

Flujo de Ejecución:

1. El administrador de bases de datos define los esquemas que se utilizarán para organizar las tablas, considerando la estructura lógica y las relaciones entre los datos.
2. El administrador de bases de datos crea las tablas dentro de los esquemas definidos, especificando los nombres de las columnas, los tipos de datos, las restricciones y las relaciones con otras tablas.
3. El administrador de bases de datos configura los índices, claves primarias y claves foráneas en las tablas, para garantizar la integridad y eficiencia de los datos.
4. El administrador de bases de datos realiza pruebas para verificar la correcta creación de los esquemas y tablas, asegurándose de que se ajusten a los requisitos del proyecto.
5. El administrador de bases de datos documenta la estructura de la base de datos, incluyendo los esquemas, tablas, columnas y relaciones, para futuras referencias y mantenimiento.
6. El caso de uso finaliza.

### Caso de Uso: Integración de los Datos de las Tablas

Descripción: Este caso de uso describe el proceso de integración de los datos en las tablas de una base de datos. Incluye la inserción, actualización y eliminación de registros para mantener la consistencia de los datos.

Actores: Administrador de bases de datos, Aplicación o usuarios autorizados

Disparador: Adición, modificación o eliminación de datos en la aplicación

Precondiciones: Se han creado las tablas y esquemas necesarios en la base de datos.

Postcondiciones: Los datos se han integrado correctamente en las tablas de la base de datos.

Flujo de Ejecución:

1. El administrador de bases de datos o la aplicación autorizada recibe los datos a ser integrados en la base de datos.
2. El administrador de bases de datos o la aplicación autorizada realiza la validación y limpieza de los datos, asegurándose de que cumplan con los formatos y restricciones establecidos en las tablas.
3. El administrador de bases de datos o la aplicación autorizada ejecuta las sentencias SQL correspondientes para insertar, actualizar o eliminar los registros en las tablas.
4. El administrador de bases de datos o la aplicación autorizada realiza pruebas y verifica la correcta integración de los datos, verificando que los registros se hayan añadido, actualizado o eliminado según lo esperado.
5. El administrador de bases de datos o la aplicación autorizada registra los cambios realizados en

un registro de auditoría o bitácora, para fines de seguimiento y control de los datos.

6. El caso de uso finaliza.

#### Caso de Uso: Implementación de la Seguridad de los Datos

**Descripción:** Este caso de uso describe el proceso de implementación de la seguridad de los datos en un SGBD. Incluye la protección de los datos contra accesos no autorizados, asegurando la confidencialidad, integridad y disponibilidad de la información.

**Actores:** Administrador de bases de datos, Administrador de seguridad

**Disparador:** Configuración inicial de la base de datos o necesidad de mejorar la seguridad de los datos

**Precondiciones:** Los usuarios y roles de acceso han sido definidos en el SGBD.

**Postcondiciones:** La seguridad de los datos se ha implementado según las políticas y requerimientos establecidos.

#### Flujo de Ejecución:

1. El administrador de bases de datos identifica los requisitos de seguridad, incluyendo la clasificación de los datos, los niveles de acceso y los controles necesarios.
2. El administrador de bases de datos configura los permisos y privilegios de los usuarios y roles, restringiendo el acceso a las tablas y datos sensibles.
3. El administrador de bases de datos implementa medidas de cifrado para proteger los datos confidenciales en reposo y en tránsito.
4. El administrador de bases de datos establece políticas de contraseñas seguras y realiza auditorías periódicas para garantizar el cumplimiento de las medidas de seguridad.
5. El administrador de seguridad revisa y valida la implementación de la seguridad, realizando pruebas de penetración y evaluaciones de riesgos.
6. El administrador de bases de datos y el administrador de seguridad realizan monitoreo y mantenimiento continuo de la seguridad de los datos, aplicando actualizaciones y parches según sea necesario.
7. El caso de uso finaliza.

#### Caso de Uso: Creación de Llaves Primarias y Foráneas

**Descripción:** Este caso de uso describe el proceso de creación de llaves primarias y foráneas en una base de datos relacional. Incluye la definición de las restricciones de clave y la configuración de las relaciones entre las tablas.

**Actores:** Administrador de bases de datos

**Disparador:** Diseño de la estructura de la base de datos o creación de nuevas tablas

**Precondiciones:** Las tablas y columnas necesarias han sido creadas en la base de datos.

**Postcondiciones:** Se han establecido las llaves primarias y foráneas en las tablas correspondientes.

#### Flujo de Ejecución:

1. El administrador de bases de datos identifica las columnas que servirán como llaves primarias en cada tabla, generalmente seleccionando una columna única y relevante para identificar de manera única cada registro.
2. El administrador de bases de datos configura las llaves primarias en las tablas, especificando la

columna o conjunto de columnas correspondientes.

3. El administrador de bases de datos identifica las relaciones entre las tablas, determinando qué columnas en una tabla se relacionan con las llaves primarias de otra tabla.

4. El administrador de bases de datos configura las llaves foráneas en las tablas relacionadas, especificando la columna o conjunto de columnas correspondientes y la tabla a la que se hace referencia.

5. El administrador de bases de datos verifica la correcta creación de las llaves primarias y foráneas, asegurándose de que las restricciones sean aplicadas correctamente y las relaciones se establezcan sin problemas.

6. El caso de uso finaliza.

#### Caso de Uso: Restricciones de Valores

**Descripción:** Este caso de uso describe el proceso de aplicación de restricciones de valores en una base de datos. Incluye la definición de reglas y condiciones que los datos deben cumplir para mantener la integridad y consistencia de la base de datos.

**Actores:** Administrador de bases de datos

**Disparador:** Configuración inicial de la base de datos o necesidad de aplicar restricciones a los datos

**Precondiciones:** Las tablas y columnas necesarias han sido creadas en la base de datos.

**Postcondiciones:** Las restricciones de valores se han aplicado a las columnas correspondientes.

**Flujo de Ejecución:**

1. El administrador de bases de datos identifica las columnas que requerirán restricciones de valores para asegurar la integridad de los datos.

2. El administrador de bases de datos define las restricciones de valores para cada columna, estableciendo condiciones como rangos numéricos, valores permitidos o requerimientos de formato.

3. El administrador de bases de datos configura las restricciones en las columnas correspondientes, asegurándose de que las reglas sean aplicadas cuando se inserten o actualicen los datos.

4. El administrador de bases de datos realiza pruebas para verificar que las restricciones de valores funcionen correctamente, impidiendo la inserción de datos inválidos o que no cumplan con las condiciones establecidas.

5. El caso de uso finaliza.

#### Caso de Uso: Actualización de los Datos

**Descripción:** Este caso de uso describe el proceso de actualización de los datos en una base de datos. Incluye la modificación de registros existentes para reflejar cambios o correcciones en la información almacenada.

**Actores:** Administrador de bases de datos, Usuarios autorizados

**Disparador:** Necesidad de actualizar información existente en la base de datos

**Precondiciones:** Los datos a actualizar están almacenados en la base de datos y se tiene acceso y permisos suficientes.

**Postcondiciones:** Los registros actualizados reflejan los cambios realizados en la información almacenada.

**Flujo de Ejecución:**

1. El administrador de bases de datos o el usuario autorizado identifica los registros que requieren actualización, basándose en criterios como fecha, estado o condiciones específicas.
2. El administrador de bases de datos o el usuario autorizado ejecuta una sentencia SQL de actualización, especificando los valores nuevos o corregidos para las columnas correspondientes.
3. El sistema de gestión de bases de datos realiza la actualización de los registros, modificando los valores de las columnas según las instrucciones proporcionadas.
4. El administrador de bases de datos o el usuario autorizado verifica la correcta actualización de los datos, realizando consultas o visualizando los registros actualizados.
5. El caso de uso finaliza.

#### Caso de Uso: Recuperación y Acceso de los Datos

**Descripción:** Este caso de uso describe el proceso de recuperación y acceso de los datos almacenados en una base de datos. Incluye consultas y operaciones de búsqueda para obtener la información deseada.

**Actores:** Usuarios autorizados

**Disparador:** Necesidad de obtener información específica de la base de datos

**Precondiciones:** Los datos están almacenados en la base de datos y el usuario tiene acceso y permisos suficientes.

**Postcondiciones:** El usuario obtiene la información requerida de la base de datos.

**Flujo de Ejecución:**

1. El usuario define los criterios de búsqueda o consulta, especificando las columnas, condiciones y filtros necesarios para obtener los datos deseados.
2. El usuario ejecuta una sentencia SQL de consulta, utilizando comandos como SELECT, WHERE y JOIN para acceder a la información requerida.
3. El sistema de gestión de bases de datos procesa la consulta y recupera los datos correspondientes según los criterios especificados.
4. El usuario analiza los resultados obtenidos, revisando la información relevante y realizando operaciones adicionales si es necesario.
5. El caso de uso finaliza.

#### Caso de Uso: Acciones en Cascada

**Descripción:** Este caso de uso describe el efecto de las acciones en cascada en una base de datos, donde los cambios realizados en una tabla afectan automáticamente a las tablas relacionadas.

**Actores:** Administrador de bases de datos

**Disparador:** Modificación o eliminación de registros en una tabla principal

**Precondiciones:** Las tablas relacionadas y las restricciones de clave foránea están configuradas en la base de datos.

**Postcondiciones:** Las acciones en cascada se han aplicado a las tablas relacionadas según las configuraciones establecidas.

**Flujo de Ejecución:**

1. El administrador de bases de datos define las acciones en cascada que se deben aplicar a las tablas relacionadas cuando se realicen cambios en una tabla principal. Estas acciones pueden ser

actualizaciones o eliminaciones en cascada.

2. El administrador de bases de datos configura las acciones en cascada en las restricciones de clave foránea correspondientes, especificando las tablas y columnas afectadas.

3. Cuando se realiza una modificación o eliminación en la tabla principal, el sistema de gestión de bases de datos aplica automáticamente las acciones en cascada a las tablas relacionadas.

4. El administrador de bases de datos verifica que las acciones en cascada se hayan ejecutado correctamente, revisando los cambios realizados en las tablas relacionadas.

5. El caso de uso finaliza.

#### Caso de Uso: Simplificación de las Tablas

**Descripción:** Este caso de uso describe el proceso de simplificación de las tablas en una base de datos. Consiste en identificar y eliminar columnas innecesarias o redundantes para mejorar la eficiencia y el rendimiento del sistema.

**Actores:** Administrador de bases de datos

**Disparador:** Identificación de columnas innecesarias o redundantes en las tablas

**Precondiciones:** Las tablas contienen columnas que no son relevantes o que se superponen en términos de información.

**Postcondiciones:** Las tablas se simplifican y contienen solo las columnas necesarias para almacenar la información requerida.

**Flujo de Ejecución:**

1. El administrador de bases de datos analiza las tablas existentes y examina las columnas para identificar aquellas que no son relevantes o que se superponen en términos de información.

2. El administrador de bases de datos determina las columnas que se pueden eliminar sin afectar la integridad de los datos o la funcionalidad del sistema.

3. El administrador de bases de datos modifica las tablas, eliminando las columnas identificadas como innecesarias o redundantes.

4. El administrador de bases de datos verifica que las tablas se hayan simplificado correctamente y que las columnas eliminadas no afecten la funcionalidad del sistema.

5. El caso de uso finaliza.

#### Caso de Uso: Eliminación de Elementos Redundantes

**Descripción:** Este caso de uso describe el proceso de eliminación de elementos redundantes en una base de datos. Consiste en identificar y eliminar registros o datos duplicados que no aportan valor adicional a la información almacenada.

**Actores:** Administrador de bases de datos

**Disparador:** Identificación de registros o datos duplicados en las tablas

**Precondiciones:** Las tablas contienen registros o datos que son duplicados y no aportan valor adicional a la información almacenada.

**Postcondiciones:** Los elementos redundantes se eliminan de las tablas, manteniendo solo un registro o dato válido.

**Flujo de Ejecución:**

1. El administrador de bases de datos analiza las tablas y busca registros o datos duplicados

utilizando consultas SQL o herramientas de detección de duplicados.

2. El administrador de bases de datos identifica los registros o datos duplicados y determina cuál de ellos se mantendrá como válido.

3. El administrador de bases de datos elimina los registros o datos duplicados de las tablas, manteniendo solo el registro o dato válido.

4. El administrador de bases de datos verifica que los elementos redundantes se hayan eliminado correctamente y que la integridad de los datos no se vea afectada.

5. El caso de uso finaliza.

#### Caso de Uso: Corrección de Errores y Ambigüedades

**Descripción:** Este caso de uso describe el proceso de corrección de errores y ambigüedades en una base de datos. Consiste en identificar y solucionar problemas de datos incorrectos, incompletos o ambiguos para garantizar la integridad y precisión de la información almacenada.

**Actores:** Administrador de bases de datos

**Disparador:** Identificación de errores o ambigüedades en los datos almacenados

**Precondiciones:** Las tablas contienen datos incorrectos, incompletos o ambiguos que afectan la integridad y precisión de la información.

**Postcondiciones:** Los errores y ambigüedades se corrigen, asegurando la integridad y precisión de los datos almacenados.

#### Flujo de Ejecución:

1. El administrador de bases de datos realiza un análisis exhaustivo de los datos almacenados, identificando posibles errores o ambigüedades.

2. El administrador de bases de datos verifica la validez y coherencia de los datos, realizando consultas y comparando la información con fuentes externas o estándares establecidos.

3. El administrador de bases de datos corrige los errores identificados, actualizando los registros o datos incorrectos o incompletos.

4. El administrador de bases de datos resuelve las ambigüedades existentes, aclarando la información almacenada y asegurando que sea comprensible y precisa.

5. El administrador de bases de datos verifica que los errores y ambigüedades se hayan corregido correctamente y que la integridad y precisión de los datos se hayan restaurado.

6. El caso de uso finaliza.

#### Caso de Uso: Mejora de la Capacidad de Análisis de la Tabla

**Descripción:** Este caso de uso describe el proceso de mejora de la capacidad de análisis de una tabla en una base de datos. Consiste en agregar columnas adicionales o realizar transformaciones en los datos existentes para facilitar el análisis y la generación de informes.

**Actores:** Administrador de bases de datos

**Disparador:** Identificación de la necesidad de una mayor capacidad de análisis en una tabla

**Precondiciones:** La tabla carece de columnas adicionales o transformaciones que permitan un análisis más detallado de los datos.

**Postcondiciones:** Se agregan columnas adicionales o se realizan transformaciones en los datos para mejorar la capacidad de análisis de la tabla.

**Flujo de Ejecución:**

1. El administrador de bases de datos identificar la necesidad de una mayor capacidad de análisis en una tabla específica. Esto puede deberse a la falta de información relevante o a la necesidad de realizar cálculos o agregaciones adicionales.
2. El administrador de bases de datos determina las columnas adicionales que se deben agregar o las transformaciones que se deben aplicar a los datos existentes para satisfacer las necesidades de análisis.
3. El administrador de bases de datos modifica la estructura de la tabla, agregando las columnas adicionales o realizando las transformaciones necesarias en los datos existentes.
4. El administrador de bases de datos verifica que la capacidad de análisis de la tabla se haya mejorado correctamente, realizando consultas y generando informes que aprovechen las nuevas columnas o transformaciones.
5. El caso de uso finaliza.

**Caso de Uso: Creación de una Contraseña para una Base de Datos**

**Descripción:** Este caso de uso describe el proceso de creación de una contraseña para proteger el acceso a una base de datos. Se establece una contraseña segura que los usuarios deben proporcionar al intentar acceder a la base de datos.

**Actores:** Administrador de bases de datos, usuarios

**Disparador:** Configuración inicial de la base de datos o necesidad de establecer una contraseña segura.

**Precondiciones:** La base de datos no tiene una contraseña establecida o se requiere una contraseña más segura.

**Postcondiciones:** Se establece una contraseña segura para proteger el acceso a la base de datos.

**Flujo de Ejecución:**

1. El administrador de bases de datos decide establecer una contraseña para proteger el acceso a la base de datos.
2. El administrador de bases de datos genera una contraseña segura que cumple con los requisitos de complejidad, como la longitud, combinación de caracteres y uso de mayúsculas y minúsculas.
3. El administrador de bases de datos configura la base de datos para requerir la contraseña al intentar acceder.
4. Los usuarios deben proporcionar la contraseña al intentar acceder a la base de datos.
5. El sistema verifica la contraseña proporcionada por el usuario y permite el acceso si es correcta.
6. El caso de uso finaliza.

**Caso de Uso: Asignación de Prioridad a los Datos**

**Descripción:** Este caso de uso describe el proceso de asignación de prioridad a los datos en una base de datos. Consiste en clasificar los datos según su importancia y asignarles una prioridad adecuada para su gestión y protección.

**Actores:** Administrador de bases de datos

**Disparador:** Identificación de la necesidad de clasificar y priorizar los datos en la base de datos.

**Precondiciones:** La base de datos contiene datos que requieren diferentes niveles de prioridad y

gestión.

Postcondiciones: Los datos se clasifican y se les asigna una prioridad adecuada para su gestión y protección.

Flujo de Ejecución:

1. El administrador de bases de datos analiza los datos almacenados en la base de datos y determina su importancia y criticidad para el sistema.
2. El administrador de bases de datos clasifica los datos en diferentes categorías o niveles de prioridad, como alta, media y baja, según su importancia y el impacto en el sistema en caso de pérdida o compromiso.
3. El administrador de bases de datos establece políticas y procedimientos para la gestión y protección de los datos en función de su nivel de prioridad.
4. Los datos se gestionan y protegen de acuerdo con las políticas establecidas, asegurando un tratamiento adecuado según su prioridad.
5. El caso de uso finaliza.

Caso de Uso: Implementación de Restricciones de Acciones

Descripción: Este caso de uso describe la implementación de restricciones de acciones en una base de datos. Consiste en establecer reglas y restricciones para controlar las acciones que pueden realizar los usuarios sobre los datos almacenados.

Actores: Administrador de bases de datos, usuarios

Disparador: Necesidad de controlar y restringir las acciones que los usuarios pueden realizar en la base de datos.

Precondiciones: Los usuarios tienen ciertos privilegios y se requiere una mayor seguridad y control sobre las acciones que pueden realizar.

Postcondiciones: Se establecen restricciones y reglas que limitan las acciones que los usuarios pueden realizar en la base de datos.

Flujo de Ejecución:

1. El administrador de bases de datos identifica las acciones que deben restringirse o controlarse en la base de datos. Esto puede incluir la modificación de datos sensibles, la eliminación de registros o la ejecución de ciertas consultas.
2. El administrador de bases de datos establece restricciones y reglas que limitan las acciones permitidas a los usuarios. Esto puede incluir la configuración de permisos y privilegios, la creación de reglas de validación o la implementación de disparadores.
3. Los usuarios intentan realizar acciones que están restringidas por las reglas establecidas.
4. El sistema verifica las restricciones y permite o deniega las acciones solicitadas según las reglas establecidas.
5. El caso de uso finaliza.

Caso de Uso: Registro de Actividades

Descripción: Este caso de uso describe el registro de actividades en una base de datos. Consiste en mantener un registro detallado de las acciones realizadas por los usuarios en la base de datos, permitiendo la auditoría y el seguimiento de eventos importantes.

Actores: Sistema de registro, administrador de bases de datos

Disparador: Realización de acciones por parte de los usuarios en la base de datos

Precondiciones: Se requiere un registro detallado de las acciones realizadas por los usuarios en la base de datos.

Postcondiciones: Se registran de forma detallada las actividades realizadas por los usuarios en la base de datos.

Flujo de Ejecución:

1. El sistema de registro captura y registra las acciones realizadas por los usuarios en la base de datos, como consultas, inserciones, actualizaciones o eliminaciones de datos.
2. El administrador de bases de datos puede acceder al registro de actividades y revisar las acciones realizadas por los usuarios.
3. El registro de actividades permite la auditoría y el seguimiento de eventos importantes, facilitando la identificación de posibles problemas, errores o actividades sospechosas.
4. El caso de uso finaliza.

Caso de Uso: Creación de una Copia de Seguridad

Descripción: Este caso de uso describe el proceso de creación de una copia de seguridad de una base de datos. Consiste en realizar una copia de los datos almacenados en la base de datos con el fin de protegerlos contra pérdidas o daños.

Actores: Administrador de bases de datos

Disparador: Necesidad de proteger los datos almacenados en la base de datos contra pérdidas o daños.

Precondiciones: Se requiere realizar copias de seguridad periódicas de los datos almacenados en la base de datos.

Postcondiciones: Se crea una copia de seguridad de los datos almacenados en la base de datos.

Flujo de Ejecución:

1. El administrador de bases de datos determina la frecuencia y el método de creación de copias de seguridad. Esto puede incluir copias de seguridad completas, incrementales o diferenciales.
2. El administrador de bases de datos inicia el proceso de creación de una copia de seguridad de la base de datos.
3. Los datos almacenados en la base de datos se copian y se guardan en un medio de almacenamiento seguro, como un disco duro externo o una unidad de cinta.
4. El administrador de bases de datos verifica la integridad de la copia de seguridad para asegurarse de que los datos se hayan copiado correctamente.
5. La copia de seguridad se guarda en un lugar seguro y se mantiene actualizada de acuerdo con la política de copias de seguridad establecida.
6. En caso de pérdida o daño en la base de datos, se puede restaurar la copia de seguridad para recuperar los datos.
7. El caso de uso finaliza.

Caso de Uso: Creación de una Interfaz de una API

Descripción: Este caso de uso describe el proceso de creación de una interfaz de programación de aplicaciones (API). Consiste en diseñar y desarrollar una interfaz que permita a los usuarios

interactuar con un sistema a través de servicios web.

Actores: Desarrollador de la API, usuarios de la API

Disparador: Necesidad de proporcionar una forma estandarizada y estructurada para que los usuarios interactúen con el sistema a través de servicios web.

Precondiciones: Se requiere un diseño claro de la funcionalidad y los servicios que se ofrecerán a través de la API.

Postcondiciones: Se crea una interfaz de API que expone los servicios y la funcionalidad del sistema.

Flujo de Ejecución:

1. El desarrollador de la API define los servicios y la funcionalidad que se ofrecerán a través de la API.

2. Se selecciona un protocolo de comunicación, como REST o SOAP, para implementar la API.

3. Se definen los endpoints de la API, que representan los diferentes recursos y operaciones que se pueden acceder a través de la interfaz.

4. Se establecen los métodos de autenticación y autorización para controlar el acceso a los servicios de la API.

5. Se implementan los mecanismos de respuesta, como el formato de los datos devueltos y los códigos de estado HTTP.

6. Se documenta la interfaz de la API, incluyendo la descripción de los servicios, los parámetros esperados y las posibles respuestas.

7. Los usuarios de la API pueden interactuar con el sistema a través de las llamadas a los servicios expuestos en la interfaz.

8. El caso de uso finaliza.

#### Caso de Uso: Acceso de los Datos desde la API

Descripción: Este caso de uso describe el proceso de acceso a los datos desde una API. Consiste en permitir que los usuarios accedan y manipulen los datos almacenados en un sistema a través de servicios web.

Actores: Usuarios de la API

Disparador: Necesidad de acceder y manipular los datos almacenados en un sistema a través de servicios web.

Precondiciones: Se requiere una conexión a la API y los permisos adecuados para acceder a los datos.

Postcondiciones: Los usuarios pueden acceder y manipular los datos almacenados en el sistema a través de los servicios de la API.

Flujo de Ejecución:

1. Los usuarios autentican su identidad y obtienen los permisos adecuados para acceder a los datos a través de la API.

2. Los usuarios realizan una llamada a los servicios de la API para acceder a los datos requeridos.

3. La API valida la solicitud y verifica los permisos de acceso del usuario.

4. La API recupera los datos solicitados del sistema de almacenamiento correspondiente.

5. La API devuelve los datos al usuario en el formato especificado, como JSON o XML.

- |  |
|--|
| 6. Los usuarios pueden manipular los datos según sus necesidades, como crear, leer, actualizar o eliminar registros. |
| 7. Los cambios realizados en los datos a través de la API se reflejan en el sistema de almacenamiento subyacente.    |
| 8. El caso de uso finaliza.  |

#### Caso de Uso: Implementación de Medidas de Seguridad de la API

**Descripción:** Este caso de uso describe la implementación de medidas de seguridad en una API. Consiste en proteger la API y los datos contra posibles amenazas y garantizar la confidencialidad, integridad y disponibilidad de los recursos.

**Actores:** Desarrollador de la API, usuarios de la API

**Disparador:** Necesidad de garantizar la seguridad de la API y los datos que maneja.

**Precondiciones:** Se requiere una evaluación de los riesgos y amenazas potenciales a la seguridad de la API.

**Postcondiciones:** Se implementan medidas de seguridad adecuadas para proteger la API y los datos.

#### Flujo de Ejecución:

1. El desarrollador de la API identifica los posibles riesgos y amenazas a la seguridad de la API.
2. Se implementan mecanismos de autenticación para verificar la identidad de los usuarios de la API.
3. Se establecen mecanismos de autorización para controlar los privilegios de acceso a los servicios y datos de la API.
4. Se implementa el cifrado de datos para garantizar la confidencialidad de la información transmitida a través de la API.
5. Se aplican técnicas de validación y saneamiento de datos para prevenir ataques como la inyección de código.
6. Se monitorea y registra la actividad de la API para detectar y responder a posibles incidentes de seguridad.
7. Se establecen políticas de seguridad y se realizan evaluaciones periódicas para mantener y mejorar la seguridad de la API.
8. El caso de uso finaliza.

#### Caso de Uso: Ejecución de Pruebas y Monitoreo de la API

**Descripción:** Este caso de uso describe el proceso de ejecución de pruebas y monitoreo de una API. Consiste en verificar el correcto funcionamiento de la API y monitorear su rendimiento y disponibilidad.

**Actores:** Equipo de desarrollo, administrador de la API

**Disparador:** Necesidad de garantizar la calidad y rendimiento de la API durante su funcionamiento.

**Precondiciones:** La API ha sido implementada y está lista para ser probada y monitoreada.

**Postcondiciones:** Se verifica el correcto funcionamiento y rendimiento de la API y se toman medidas correctivas si es necesario.

#### Flujo de Ejecución:

1. Se definen los casos de prueba que cubren diferentes escenarios y funcionalidades de la API.

2. Se ejecutan las pruebas de la API, comprobando el cumplimiento de los requisitos y la correcta respuesta a las diferentes solicitudes.
3. Se registra y analiza el resultado de las pruebas, identificando posibles problemas o errores en la API.
4. Se monitorea el rendimiento de la API, registrando métricas como la latencia, el tiempo de respuesta y la utilización de recursos.
5. Se detectan y resuelven posibles cuellos de botella o problemas de rendimiento de la API.
6. Se realiza un seguimiento continuo del monitoreo de la API, evaluando su rendimiento y disponibilidad en tiempo real.
7. Se toman medidas correctivas si se detectan problemas o se violan los acuerdos de nivel de servicio (SLA).
8. El caso de uso finaliza.

#### Caso de Uso: Documentación de la API

**Descripción:** Este caso de uso describe el proceso de documentación de una API. Consiste en proporcionar información detallada sobre los servicios, endpoints, parámetros y respuestas de la API para facilitar su uso y comprensión por parte de los usuarios.

**Actores:** Desarrollador de la API, usuarios de la API

**Disparador:** Necesidad de brindar documentación clara y completa sobre la API y sus funcionalidades.

**Precondiciones:** La API ha sido implementada y probada correctamente.

**Postcondiciones:** Se genera y publica la documentación de la API para su acceso por parte de los usuarios.

#### Flujo de Ejecución:

1. El desarrollador de la API genera una documentación que describe los servicios, endpoints, parámetros y respuestas de la API.
2. Se documentan los posibles errores y códigos de estado que pueden devolver los servicios de la API.
3. Se proporcionan ejemplos y casos de uso para ilustrar el uso correcto de la API.
4. Se publica la documentación en un formato accesible, como una página web o un archivo PDF.
5. Los usuarios de la API acceden a la documentación para comprender y utilizar adecuadamente los servicios de la API.
6. La documentación se actualiza regularmente para reflejar cambios o mejoras en la API.
7. El caso de uso finaliza.

#### Caso de Uso: Creación de Tablas y sus Datos

**Descripción:** Este caso de uso describe el proceso de creación de tablas en una base de datos y la inserción de datos en dichas tablas.

**Actores:** Administrador de la base de datos

**Disparador:** Necesidad de almacenar datos en una estructura organizada y consultable.

**Precondiciones:** La base de datos está disponible y se ha diseñado el esquema de las tablas.

**Postcondiciones:** Las tablas se crean y se insertan los datos correspondientes.

**Flujo de Ejecución:**

1. El administrador de la base de datos define el esquema de las tablas, especificando los nombres de las tablas y los campos correspondientes.
2. Se crea cada tabla en la base de datos, estableciendo las columnas, tipos de datos y restricciones necesarias.
3. Se insertan los datos en las tablas, proporcionando los valores correspondientes para cada campo.
4. Se verifica que los datos se hayan insertado correctamente y se reflejan en las tablas.
5. El caso de uso finaliza.

**Caso de Uso: Identificación del Valor de los Datos**

**Descripción:** Este caso de uso describe el proceso de identificar el valor de los datos almacenados en una base de datos.

**Actores:** Usuarios de la base de datos

**Disparador:** Necesidad de conocer y utilizar el valor de los datos almacenados en la base de datos.

**Precondiciones:** La base de datos contiene los datos relevantes para el usuario.

**Postcondiciones:** El usuario accede y utiliza el valor de los datos según sea necesario.

**Flujo de Ejecución:**

1. El usuario identifica los datos de interés dentro de la base de datos.
2. El usuario realiza consultas o ejecuta operaciones para acceder a los datos específicos.
3. Se obtienen los valores de los datos requeridos y se utilizan según las necesidades del usuario.
4. El caso de uso finaliza.

**Caso de Uso: Accesibilidad de los Datos**

**Descripción:** Este caso de uso describe cómo acceder a los datos almacenados en una base de datos de forma segura y controlada.

**Actores:** Usuarios de la base de datos

**Disparador:** Necesidad de acceder a los datos almacenados en la base de datos.

**Precondiciones:** Los usuarios tienen los privilegios adecuados para acceder a los datos.

**Postcondiciones:** Los usuarios pueden acceder a los datos de manera segura y controlada.

**Flujo de Ejecución:**

1. Los usuarios autentican su identidad para acceder a la base de datos.
2. Se verifican los privilegios de los usuarios para determinar qué datos pueden acceder y qué operaciones pueden realizar.
3. Los usuarios realizan consultas o ejecutan operaciones permitidas para acceder a los datos requeridos.
4. Los datos solicitados se devuelven al usuario de manera segura y controlada.
5. El caso de uso finaliza.

#### Caso de Uso: Verificación de los Datos que sean Compatibles

Descripción: Este caso de uso describe el proceso de verificación de los datos para garantizar su compatibilidad con las restricciones y reglas definidas en la base de datos.

Actores: Base de datos

Disparador: Inserción o actualización de datos en la base de datos.

Precondiciones: Se han definido restricciones y reglas para los datos en la base de datos.

Postcondiciones: Los datos se verifican y se asegura su compatibilidad con las restricciones y reglas definidas.

Flujo de Ejecución:

1. Se ingresan o actualizan datos en la base de datos.
2. La base de datos verifica que los datos cumplan con las restricciones de tipo, tamaño, formato, etc. definidas para los campos correspondientes.
3. Si los datos son compatibles con las restricciones, se aceptan y se almacenan en la base de datos.
4. Si los datos no son compatibles, se rechazan y se emite un mensaje de error indicando la causa del rechazo.
5. El caso de uso finaliza.

#### Caso de Uso: Protección de la Integridad de los Datos

Descripción: Este caso de uso describe cómo se protege la integridad de los datos almacenados en una base de datos, evitando la corrupción o pérdida de los mismos.

Actores: Base de datos

Disparador: Acciones que pueden afectar la integridad de los datos, como inserción, actualización o eliminación.

Precondiciones: Los datos almacenados en la base de datos están en un estado válido y coherente.

Postcondiciones: Se garantiza la integridad de los datos almacenados en la base de datos.

Flujo de Ejecución:

1. Se definen restricciones y reglas para los datos en la base de datos, como llaves primarias, llaves foráneas, restricciones de unicidad, etc.
2. Se implementan mecanismos de validación y verificación de los datos para asegurar su coherencia y consistencia.
3. Se establecen procedimientos de respaldo y recuperación de datos para prevenir la pérdida o corrupción de los mismos.
4. Se realizan acciones de mantenimiento periódicas para detectar y corregir posibles problemas de integridad de los datos.
5. El caso de uso finaliza.

#### Caso de Uso: Verificación de la Unicidad de los Datos

**Descripción:** Este caso de uso describe el proceso de verificación de la unicidad de los datos en una base de datos, evitando la duplicación o repetición de información.

**Actores:** Base de datos

**Disparador:** Inserción o actualización de datos en la base de datos.

**Precondiciones:** Se han definido restricciones de unicidad en la base de datos.

**Postcondiciones:** Los datos se verifican y se asegura su unicidad en la base de datos.

**Flujo de Ejecución:**

1. Se ingresan o actualizan datos en la base de datos.
2. La base de datos verifica que los datos no violen las restricciones de unicidad definidas para los campos correspondientes.
3. Si los datos son únicos, se aceptan y se almacenan en la base de datos.
4. Si los datos ya existen y violan la restricción de unicidad, se rechazan y se emite un mensaje de error indicando la duplicación.
5. El caso de uso finaliza.

**Caso de Uso: Diseño de un Tutorial**

**Descripción:** Este caso de uso describe el proceso de diseño de un tutorial en un videojuego, que tiene como objetivo guiar al jugador y enseñarle las mecánicas del juego.

**Actores:** Diseñador del juego, Jugador

**Disparador:** Inicio de una nueva partida o activación del tutorial por parte del jugador.

**Precondiciones:** El juego tiene un tutorial diseñado y disponible para ser activado.

**Postcondiciones:** El jugador recibe instrucciones y guía sobre las mecánicas del juego a través del tutorial.

**Flujo de Ejecución:**

1. El jugador inicia una nueva partida o activa el tutorial desde el menú del juego.
2. El juego muestra una introducción al tutorial, explicando su propósito y cómo funcionará.
3. El jugador recibe instrucciones paso a paso sobre las mecánicas del juego, como controles, movimientos, interacciones, etc.
4. El tutorial presenta ejemplos y ejercicios prácticos para que el jugador practique las técnicas aprendidas.
5. El jugador avanza a través de las secciones del tutorial, siguiendo las instrucciones y completando los desafíos propuestos.
6. El tutorial proporciona retroalimentación inmediata al jugador sobre su desempeño y le brinda consejos para mejorar.
7. El jugador finaliza el tutorial una vez que ha completado todas las secciones o decide abandonarlo en cualquier momento.
8. El caso de uso finaliza.

**Caso de Uso: Presentación de las Instrucciones**

**Descripción:** Este caso de uso describe cómo se presentan las instrucciones al jugador durante el tutorial de un videojuego.

Actores: Tutorial, Jugador

Disparador: Inicio de una nueva sección del tutorial o acción específica del jugador.

Precondiciones: El jugador se encuentra en una sección del tutorial que requiere instrucciones.

Postcondiciones: El jugador recibe instrucciones claras y concisas para completar una tarea o desafío específico.

Flujo de Ejecución:

1. El jugador llega a una nueva sección del tutorial que requiere instrucciones.
2. El tutorial muestra un mensaje en pantalla o utiliza elementos visuales para resaltar los pasos a seguir.
3. El jugador lee las instrucciones proporcionadas y las comprende.
4. El tutorial espera a que el jugador realice la acción correspondiente según las instrucciones dadas.
5. El tutorial proporciona retroalimentación inmediata al jugador sobre su desempeño y le brinda consejos adicionales si es necesario.
6. El jugador completa la tarea o desafío de acuerdo con las instrucciones.
7. El caso de uso finaliza.

Caso de Uso: Interacción del Tutorial con el Jugador

Descripción: Este caso de uso describe cómo el tutorial interactúa con el jugador durante su progreso.

Actores: Tutorial, Jugador

Disparador: Acciones del jugador, avance en el tutorial.

Precondiciones: El jugador se encuentra en una sección del tutorial en curso.

Postcondiciones: El jugador recibe retroalimentación, consejos y guía para mejorar su desempeño en el juego.

Flujo de Ejecución:

1. El jugador realiza una acción o avanza en el tutorial.
2. El tutorial monitorea la actividad del jugador y analiza su desempeño.
3. El tutorial proporciona retroalimentación inmediata al jugador, elogiando los logros y corrigiendo los errores.
4. El tutorial ofrece consejos y sugerencias para mejorar el rendimiento del jugador en el juego.
5. El jugador continúa siguiendo las instrucciones y realizando las acciones requeridas.
6. El caso de uso finaliza

Caso de Uso: Retroalimentación del Juego hacia el Jugador

Descripción: Este caso de uso describe cómo el juego proporciona retroalimentación al jugador durante el tutorial.

Actores: Tutorial, Jugador

Disparador: Acciones del jugador, avance en el tutorial.

Precondiciones: El jugador se encuentra en una sección del tutorial en curso.

**Postcondiciones:** El jugador recibe retroalimentación sobre su desempeño y se le ofrece consejos y guía para mejorar.

**Flujo de Ejecución:**

1. El jugador realiza una acción o avanza en el tutorial.
2. El juego evalúa el desempeño del jugador y analiza su progreso.
3. El juego proporciona retroalimentación inmediata al jugador, elogiando los logros y corrigiendo los errores.
4. El juego ofrece consejos y sugerencias para mejorar el rendimiento del jugador en el juego.
5. El jugador continúa siguiendo las instrucciones y realizando las acciones requeridas.
6. El caso de uso finaliza.

**Caso de Uso: Identificación de las Teclas para el Videojuego**

**Descripción:** Este caso de uso describe el proceso de identificar las teclas utilizadas en el videojuego.

**Actores:** Jugador, Diseñador del juego

**Disparador:** Diseño de los controles del juego.

**Precondiciones:** El juego se encuentra en desarrollo y se está diseñando la asignación de teclas.

**Postcondiciones:** Las teclas necesarias para jugar el videojuego están identificadas.

**Flujo de Ejecución:**

1. El diseñador del juego decide las acciones y funciones que serán controladas mediante teclas.
2. El diseñador asigna una tecla específica para cada acción o función del juego.
3. El diseñador documenta y comunica las teclas asignadas a los jugadores.
4. Los jugadores aprenden las teclas asignadas y se familiarizan con ellas.
5. Durante el juego, los jugadores utilizan las teclas correspondientes para controlar y realizar acciones en el juego.
6. El caso de uso finaliza.

**Caso de Uso: Configuración de las Teclas y los Controles**

**Descripción:** Este caso de uso describe cómo se configuran las teclas y los controles del videojuego según las preferencias del jugador.

**Actores:** Jugador

**Disparador:** Acceso a la configuración de controles en el menú del juego.

**Precondiciones:** El jugador desea personalizar la asignación de teclas y los controles del juego.

**Postcondiciones:** Las teclas y los controles del juego se configuran según las preferencias del jugador.

**Flujo de Ejecución:**

1. El jugador accede al menú de configuración de controles en el juego.
2. El juego muestra una lista de acciones y funciones del juego junto con las teclas o botones asignados actualmente.

3. El jugador selecciona una acción o función y elige una nueva tecla o botón para asignar.
4. El jugador repite el paso 3 para todas las acciones o funciones que desee personalizar.
5. El jugador guarda la configuración de controles personalizada.
6. Durante el juego, el jugador utiliza las teclas y los controles personalizados para controlar y realizar acciones en el juego.
7. El caso de uso finaliza.

#### Caso de Uso: Verificación de la Responsividad

**Descripción:** Este caso de uso describe cómo se verifica la responsividad de las teclas y los controles del videojuego.

**Actores:** Jugador

**Disparador:** Acción del jugador utilizando las teclas o los controles del juego.

**Precondiciones:** El juego se encuentra en ejecución y el jugador utiliza las teclas o los controles para interactuar con el juego.

**Postcondiciones:** Se verifica que las teclas y los controles responden correctamente a las acciones del jugador.

#### Flujo de Ejecución:

1. El jugador realiza una acción utilizando las teclas o los controles del juego.
2. El juego registra la acción y procesa la entrada del jugador.
3. El juego responde de manera apropiada y realiza la acción correspondiente en el juego.
4. El jugador percibe la respuesta del juego y evalúa si la responsividad es satisfactoria.
5. Si la responsividad no es satisfactoria, el jugador puede ajustar la configuración de las teclas y los controles o informar sobre el problema al desarrollador del juego.
6. El caso de uso finaliza.

#### Caso de Uso: Crear Usuario y Contraseña de un Nuevo Usuario

**Descripción:** Este caso de uso describe el proceso de creación de un nuevo usuario y contraseña en un sistema o aplicación.

**Actores:** Administrador del sistema, Nuevo Usuario

**Disparador:** Solicitud de registro por parte de un nuevo usuario.

**Precondiciones:** El nuevo usuario proporciona la información requerida para crear una cuenta.

**Postcondiciones:** Se crea un nuevo usuario con su correspondiente contraseña.

#### Flujo de Ejecución:

1. El nuevo usuario completa un formulario de registro con sus datos personales.
2. El administrador del sistema verifica la validez de los datos proporcionados por el nuevo usuario.
3. Si los datos son válidos, el administrador crea un nuevo usuario en el sistema.
4. El administrador genera una contraseña segura para el nuevo usuario.
5. El administrador comunica al nuevo usuario su nombre de usuario y contraseña.
6. El nuevo usuario recibe la información de inicio de sesión y puede acceder al sistema con su

nueva cuenta.

7. El caso de uso finaliza.

#### Caso de Uso: Verificación de la Cuenta

Descripción: Este caso de uso describe el proceso de verificación de una cuenta de usuario en un sistema o aplicación.

Actores: Usuario, Administrador del sistema

Disparador: Solicitud de verificación de cuenta por parte del usuario.

Precondiciones: El usuario ha proporcionado la información requerida para crear una cuenta.

Postcondiciones: La cuenta del usuario se verifica y se activa.

Flujo de Ejecución:

1. El usuario solicita la verificación de su cuenta a través de un enlace o proceso específico en la aplicación.
2. El administrador del sistema recibe la solicitud de verificación.
3. El administrador verifica la información proporcionada por el usuario y realiza los procedimientos necesarios para activar la cuenta.
4. Si la información es correcta, el administrador marca la cuenta como verificada y activa.
5. El usuario recibe una notificación de que su cuenta ha sido verificada y puede acceder a todas las funcionalidades del sistema o aplicación.
6. El caso de uso finaliza.

#### Caso de Uso: Recordar Inicio de Sesión

Descripción: Este caso de uso describe el proceso de recordar el inicio de sesión de un usuario en un sistema o aplicación.

Actores: Usuario, Sistema o Aplicación

Disparador: Solicitud de recordar inicio de sesión por parte del usuario.

Precondiciones: El usuario ha iniciado sesión previamente en el sistema o aplicación.

Postcondiciones: Se guarda la información necesaria para recordar el inicio de sesión del usuario.

Flujo de Ejecución:

1. El usuario marca la opción "Recordar inicio de sesión" al iniciar sesión en el sistema o aplicación.
2. El sistema o aplicación guarda la información de inicio de sesión del usuario, como el nombre de usuario y una clave de acceso segura.
3. En futuros accesos, el sistema o aplicación utiliza la información guardada para autenticar automáticamente al usuario.
4. El usuario accede al sistema o aplicación sin tener que ingresar sus credenciales manualmente.
5. El caso de uso finaliza

#### Caso de Uso: Autenticación y Accesibilidad de la Cuenta

Descripción: Este caso de uso describe el proceso de autenticación y acceso a la cuenta de usuario en un sistema o aplicación.

Actores: Usuario, Sistema o Aplicación

Disparador: Solicitud de acceso a la cuenta por parte del usuario.

Precondiciones: El usuario ha proporcionado las credenciales de acceso válidas.

Postcondiciones: El usuario accede a su cuenta y puede utilizar las funcionalidades del sistema o aplicación.

Flujo de Ejecución:

1. El usuario ingresa su nombre de usuario y contraseña en el formulario de inicio de sesión.
2. El sistema o aplicación verifica la autenticidad de las credenciales ingresadas.
3. Si las credenciales son válidas, el sistema o aplicación concede acceso a la cuenta del usuario.
4. El usuario es redirigido a la página principal o área de trabajo de su cuenta.
5. El usuario puede utilizar todas las funcionalidades disponibles para su cuenta en el sistema o aplicación.
6. El caso de uso finaliza.

Caso de Uso: Crear Reglas del Videojuego

Descripción: Este caso de uso describe el proceso de creación de las reglas del videojuego, que establecen las normas y condiciones de juego para los jugadores.

Actores: Diseñador de juegos, Equipo de desarrollo

Disparador: Inicio del proceso de diseño del videojuego.

Precondiciones: Se han definido los conceptos básicos del videojuego, como la temática, mecánicas principales y objetivos.

Postcondiciones: Se establecen las reglas del videojuego.

Flujo de Ejecución:

1. El diseñador de juegos se reúne con el equipo de desarrollo para discutir y definir las reglas del videojuego.
2. Se identifican los aspectos clave que deben regirse por reglas, como las acciones permitidas, puntuación, condiciones de victoria o derrota, entre otros.
3. El diseñador de juegos crea un documento o archivo que describe en detalle las reglas del videojuego.
4. Se revisa y se realiza una iteración de pruebas para ajustar y refinar las reglas del videojuego.
5. Una vez aprobadas, las reglas del videojuego se documentan oficialmente y se comparten con el equipo de desarrollo.
6. El equipo de desarrollo utiliza las reglas como base para implementar las mecánicas y sistemas necesarios en el videojuego.
7. El caso de uso finaliza.

Caso de Uso: Creación de un GDD (Game Design Document)

Descripción: Este caso de uso describe el proceso de creación de un GDD, que es un documento que recopila toda la información relevante sobre el diseño del videojuego.

Actores: Diseñador de juegos, Equipo de desarrollo

Disparador: Inicio del proceso de diseño y desarrollo del videojuego.

Precondiciones: Se han establecido los conceptos básicos del videojuego y se han definido las reglas.

Postcondiciones: Se crea un GDD completo que sirve como guía de referencia para el desarrollo del videojuego.

Flujo de Ejecución:

1. El diseñador de juegos recopila toda la información relevante sobre el videojuego, incluyendo la historia, personajes, mecánicas, arte, sonido, niveles, entre otros.
2. Se estructura la información de manera organizada y se redacta el GDD, incluyendo secciones como introducción, visión general, jugabilidad, arte y diseño de niveles, audio, entre otras.
3. El diseñador de juegos comparte el GDD con el equipo de desarrollo para que lo revisen y realicen aportes o sugerencias.
4. Se realizan iteraciones de revisión y ajustes en el GDD hasta que se alcance un consenso.
5. El GDD se finaliza y se utiliza como guía de referencia durante todo el proceso de desarrollo del videojuego.
6. El caso de uso finaliza.

Caso de Uso: Implementar un Sistema de Experiencia

Descripción: Este caso de uso describe el proceso de implementación de un sistema de experiencia en un videojuego, que permite a los jugadores ganar puntos de experiencia al completar acciones o desafíos en el juego.

Actores: Desarrollador de juegos, Jugador

Disparador: Inicio del desarrollo del sistema de experiencia.

Precondiciones: Se ha definido la mecánica del juego y las acciones que otorgan experiencia.

Postcondiciones: El sistema de experiencia está implementado y funcional en el juego.

Flujo de Ejecución:

1. El desarrollador de juegos diseña la mecánica del sistema de experiencia, determinando las acciones o eventos que otorgan puntos de experiencia.
2. Se implementa el sistema de seguimiento de experiencia para registrar y acumular los puntos obtenidos por el jugador.
3. El jugador realiza acciones en el juego que generan experiencia, como completar misiones, derrotar enemigos o resolver acertijos.
4. El sistema de experiencia registra los puntos ganados y actualiza el nivel o rango del jugador en función de la cantidad acumulada.
5. Se establecen recompensas y beneficios asociados a los niveles o rangos alcanzados por el jugador.
6. El jugador puede consultar su progreso y nivel de experiencia en el juego.
7. El caso de uso finaliza

Caso de Uso: Mejora de Habilidades y Atributos

Descripción: Este caso de uso describe el proceso de mejora de habilidades y atributos de un

personaje o avatar en un videojuego.

Actores: Jugador

Disparador: El jugador decide mejorar las habilidades y atributos de su personaje.

Precondiciones: El jugador tiene suficientes puntos de experiencia o recursos para realizar mejoras.

Postcondiciones: Las habilidades y atributos del personaje se mejoran según las decisiones del jugador.

Flujo de Ejecución:

1. El jugador accede a una interfaz o menú específico para gestionar las mejoras de habilidades y atributos.

2. Se muestran las opciones disponibles para mejorar, como fuerza, velocidad, resistencia, habilidades especiales, etc.

3. El jugador selecciona las habilidades o atributos que desea mejorar y asigna los puntos de experiencia o recursos correspondientes.

4. Se actualizan las estadísticas del personaje según las mejoras realizadas.

5. El jugador puede visualizar los cambios y las nuevas habilidades o capacidades desbloqueadas.

6. El caso de uso finaliza

Caso de Uso: Mejora de Equipamiento

Descripción: Este caso de uso describe el proceso de mejora del equipamiento de un personaje en un videojuego, que le proporciona ventajas y mejoras en el rendimiento.

Actores: Jugador

Disparador: El jugador decide mejorar el equipamiento de su personaje.

Precondiciones: El jugador tiene acceso a opciones de mejora de equipamiento y recursos necesarios.

Postcondiciones: El equipamiento del personaje se mejora según las decisiones del jugador.

Flujo de Ejecución:

1. El jugador accede a una interfaz o tienda para gestionar las mejoras de equipamiento.

2. Se muestra una lista de opciones de equipamiento disponibles para mejorar, como armas, armaduras, accesorios, etc.

3. El jugador selecciona el equipamiento que desea mejorar y asigna los recursos necesarios, como monedas virtuales, materiales, etc.

4. Se actualiza el equipamiento del personaje con las mejoras realizadas.

5. El jugador puede visualizar los cambios en las estadísticas y habilidades proporcionadas por el nuevo equipamiento.

6. El caso de uso finaliza

Caso de Uso: Actualización de la Progresión del Juego

Descripción: Este caso de uso describe el proceso de actualización y progresión del contenido del juego a medida que el jugador avanza en la historia o alcanza ciertos hitos.

Actores: Desarrollador de juegos

**Disparador:** El jugador alcanza un punto clave en la historia o cumple ciertos requisitos para desbloquear contenido adicional.

**Precondiciones:** Se ha definido la estructura y el contenido del juego en etapas o niveles.

**Postcondiciones:** Se desbloquea y actualiza el contenido del juego según el progreso del jugador.

**Flujo de Ejecución:**

1. El desarrollador de juegos establece hitos o requisitos para desbloquear nuevo contenido, como niveles, áreas, misiones, personajes, etc.

2. El jugador alcanza un hito o cumple los requisitos necesarios para desbloquear el contenido adicional.

3. Se muestra al jugador el nuevo contenido desbloqueado y se actualiza la progresión del juego en el sistema de seguimiento.

4. El jugador puede acceder al nuevo contenido y continuar su progreso en el juego.

5. El caso de uso finaliza.

#### Caso de Uso: Aumentar la Dificultad del Juego

**Descripción:** Este caso de uso describe el proceso de aumento de la dificultad en un videojuego para proporcionar un mayor desafío a los jugadores a medida que avanzan en el juego.

**Actores:** Desarrollador de juegos

**Disparador:** El jugador alcanza un nivel de habilidad o progreso que requiere un mayor desafío.

**Precondiciones:** Se ha establecido una estructura de niveles o dificultades escalonadas en el juego.

**Postcondiciones:** La dificultad del juego se incrementa para proporcionar un mayor desafío al jugador.

**Flujo de Ejecución:**

1. El desarrollador de juegos define un sistema de dificultad escalonada o niveles que se desbloquean a medida que el jugador prograza.

2. El jugador alcanza un nivel de habilidad o progreso que activa el aumento de dificultad.

3. Se aplican modificaciones al juego para aumentar la dificultad, como enemigos más fuertes, obstáculos adicionales, restricciones de tiempo, etc.

4. El jugador se enfrenta al nuevo nivel de dificultad y debe adaptarse a los desafíos adicionales.

5. El caso de uso finaliza

#### Caso de Uso: Crear Estados y Transiciones de Máquinas de Estado

**Descripción:** Este caso de uso describe el proceso de definición de los estados y transiciones en una máquina de estado para controlar el comportamiento de un sistema.

**Actores:** Desarrollador de software

**Disparador:** El desarrollador inicia el proceso de diseño y desarrollo de la máquina de estado.

**Precondiciones:** Se ha identificado el sistema o componente que requerirá una máquina de estado.

**Postcondiciones:** Se han creado los estados y transiciones necesarios para la máquina de estado.

**Flujo de Ejecución:**

1. El desarrollador identifica los posibles estados que el sistema puede tener en su ciclo de vida.

2. Se definen los estados y se asignan nombres descriptivos a cada uno.
3. El desarrollador define las transiciones entre los estados y los eventos que disparan dichas transiciones.
4. Se especifican las condiciones o acciones asociadas a cada transición, si las hay.
5. Se valida el diseño de la máquina de estado y se realiza cualquier ajuste necesario.
6. El caso de uso finaliza.

#### Caso de Uso: Crear Eventos de Máquinas de Estado

**Descripción:** Este caso de uso describe el proceso de identificación y creación de eventos que disparan las transiciones en una máquina de estado.

**Actores:** Desarrollador de software

**Disparador:** El desarrollador inicia el proceso de diseño y desarrollo de la máquina de estado.

**Precondiciones:** Se han definido los estados y transiciones de la máquina de estado.

**Postcondiciones:** Se han creado los eventos necesarios para disparar las transiciones.

**Flujo de Ejecución:**

1. El desarrollador identifica los posibles eventos que pueden ocurrir en el sistema y que son relevantes para la máquina de estado.
2. Se asignan nombres descriptivos a cada evento.
3. El desarrollador define las transiciones asociadas a cada evento y los estados de origen y destino correspondientes.
4. Se especifican las condiciones o acciones asociadas a cada transición, si las hay.
5. Se valida el diseño de los eventos y las transiciones.
6. El caso de uso finaliza.

#### Caso de Uso: Crear Acciones y Condiciones

**Descripción:** Este caso de uso describe el proceso de definición de acciones y condiciones asociadas a las transiciones de una máquina de estado.

**Actores:** Desarrollador de software

**Disparador:** El desarrollador inicia el proceso de diseño y desarrollo de la máquina de estado.

**Precondiciones:** Se han definido los estados y transiciones de la máquina de estado.

**Postcondiciones:** Se han creado las acciones y condiciones necesarias para las transiciones.

**Flujo de Ejecución:**

1. El desarrollador identifica las acciones que deben llevarse a cabo al ocurrir una transición entre estados.
2. Se asignan nombres descriptivos a cada acción.
3. El desarrollador define las condiciones que deben cumplirse para que una transición se active.
4. Se especifican las acciones asociadas a cada transición, si las hay.
5. Se valida el diseño de las acciones y condiciones.
6. El caso de uso finaliza.

### Caso de Uso: Implementar Tolerancia a Fallos

Descripción: Este caso de uso describe el proceso de implementación de mecanismos de tolerancia a fallos en una máquina de estado para garantizar la confiabilidad del sistema.

Actores: Desarrollador de software

Disparador: El desarrollador inicia el proceso de diseño y desarrollo de la máquina de estado.

Precondiciones: Se han definido los estados y transiciones de la máquina de estado.

Postcondiciones: Se han implementado los mecanismos de tolerancia a fallos necesarios en la máquina de estado.

Flujo de Ejecución:

1. El desarrollador identifica los posibles puntos de fallo en el sistema y las transiciones que podrían verse afectadas por ellos.
2. Se diseñan estrategias de recuperación de fallos para cada punto identificado.
3. Se implementan los mecanismos de recuperación de fallos en la máquina de estado.
4. Se validan los mecanismos de tolerancia a fallos.
5. El caso de uso finaliza.

### Caso de Uso: Encriptar los Datos de la Página Web

Descripción: Este caso de uso describe el proceso de encriptación de los datos transmitidos y almacenados en una página web para garantizar la seguridad de la información.

Actores: Desarrollador de software, Administrador del sistema

Disparador: El desarrollador o administrador inicia el proceso de implementación de la encriptación de datos en la página web.

Precondiciones: Se ha identificado la necesidad de proteger los datos de la página web.

Postcondiciones: Los datos transmitidos y almacenados en la página web están encriptados.

Flujo de Ejecución:

1. El desarrollador o administrador selecciona un algoritmo de encriptación adecuado para la página web.
2. Se implementa la encriptación de los datos en la capa de transporte (HTTPS) para asegurar la comunicación entre el servidor y el cliente.
3. Se utiliza un algoritmo de encriptación adecuado para almacenar los datos sensibles en la base de datos de la página web.
4. Se verifica que la encriptación se ha implementado correctamente y se realizan pruebas de funcionamiento.
5. El caso de uso finaliza

### Caso de Uso: Monitoreo y Prevención de Ataques Web

Descripción: Este caso de uso describe el proceso de monitoreo y prevención de ataques web para proteger la página web contra posibles vulnerabilidades y ataques maliciosos.

Actores: Administrador del sistema, Equipo de seguridad

**Disparador:** El administrador del sistema o el equipo de seguridad inicia el proceso de monitoreo y prevención de ataques web.

**Precondiciones:** Se ha implementado una página web y se ha identificado la necesidad de protegerla contra posibles ataques.

**Postcondiciones:** La página web está monitoreada y protegida contra ataques web.

**Flujo de Ejecución:**

1. El administrador del sistema o el equipo de seguridad implementa un sistema de monitoreo de seguridad en la página web.
2. Se realizan análisis de vulnerabilidades y pruebas de seguridad periódicas para identificar posibles riesgos y brechas de seguridad.
3. Se implementan medidas de prevención de ataques, como firewalls, sistemas de detección de intrusiones y filtrado de solicitudes maliciosas.
4. Se lleva a cabo una monitorización continua de la página web para detectar y responder a posibles ataques o anomalías en tiempo real.
5. Se realiza un seguimiento y registro de los intentos de ataque, así como de las acciones tomadas para mitigarlos.
6. El caso de uso finaliza.

**Caso de Uso: Crear Función de Autenticación y Autorización**

**Descripción:** Este caso de uso describe el proceso de creación de una función de autenticación y autorización en la página web para controlar el acceso de los usuarios y proteger los recursos del sistema.

**Actores:** Desarrollador de software, Administrador del sistema

**Disparador:** El desarrollador o administrador inicia el proceso de implementación de la función de autenticación y autorización en la página web.

**Precondiciones:** Se ha identificado la necesidad de controlar el acceso de los usuarios y proteger los recursos del sistema.

**Postcondiciones:** Se ha implementado la función de autenticación y autorización en la página web.

**Flujo de Ejecución:**

1. El desarrollador o administrador define los requisitos de autenticación y autorización para la página web.
2. Se implementa un sistema de registro y autenticación de usuarios, que incluye la verificación de credenciales y la generación de tokens de sesión.
3. Se definen los roles y permisos de los usuarios, así como las reglas de autorización para acceder a los recursos del sistema.
4. Se realiza una validación exhaustiva de la función de autenticación y autorización para garantizar su correcto funcionamiento.
5. El caso de uso finaliza

**Caso de Uso: Actualizar la Página con Parches de Seguridad**

**Descripción:** Este caso de uso describe el proceso de actualización de la página web con parches de seguridad para corregir posibles vulnerabilidades y garantizar un entorno seguro.

Actores: Administrador del sistema

Disparador: El administrador del sistema inicia el proceso de actualización de la página web con parches de seguridad.

Precondiciones: Se ha identificado la necesidad de corregir vulnerabilidades de seguridad en la página web.

Postcondiciones: La página web está actualizada con los últimos parches de seguridad.

Flujo de Ejecución:

1. El administrador del sistema identifica los parches de seguridad necesarios para corregir las vulnerabilidades detectadas en la página web.
2. Se descargan los parches de seguridad desde fuentes confiables y se verifican su autenticidad.
3. Se realiza una copia de seguridad de la página web y se detiene temporalmente su funcionamiento.
4. Se aplican los parches de seguridad a la página web, asegurando que se implementen correctamente.
5. Se realizan pruebas exhaustivas para verificar que los parches de seguridad no afecten negativamente el funcionamiento de la página web.
6. Se reinicia la página web y se realiza un monitoreo adicional para confirmar que los parches de seguridad se hayan aplicado correctamente.
7. El caso de uso finaliza

Caso de Uso: Crear un Diseño Amigable de Interfaz de Usuario

Descripción: Este caso de uso describe el proceso de diseño de una interfaz de usuario amigable y fácil de usar para mejorar la experiencia del usuario en la página web o la aplicación.

Actores: Diseñador de UX/UI, Desarrollador de software

Disparador: El diseñador o desarrollador inicia el proceso de creación de un diseño amigable de interfaz de usuario.

Precondiciones: Se ha identificado la necesidad de mejorar la usabilidad y la experiencia del usuario en la página web o la aplicación.

Postcondiciones: Se ha creado un diseño amigable de interfaz de usuario.

Flujo de Ejecución:

1. El diseñador realiza una investigación sobre las necesidades y preferencias de los usuarios objetivo.
2. Se definen los objetivos y requisitos del diseño de la interfaz de usuario, incluyendo la navegación, la disposición de elementos, los colores y la tipografía.
3. Se crea un prototipo de la interfaz de usuario utilizando herramientas de diseño gráfico y prototipado.
4. Se realiza una evaluación y prueba del prototipo con usuarios reales para obtener retroalimentación y realizar ajustes.
5. El diseñador trabaja en la iteración del diseño en base a la retroalimentación recibida.
6. El desarrollador implementa el diseño de la interfaz de usuario en la página web o la aplicación.
7. Se realiza una prueba de usabilidad final para verificar la efectividad del diseño amigable de interfaz de usuario.

8. El caso de uso finaliza.

#### Caso de Uso: Implementar Seguridad

Descripción: Este caso de uso describe el proceso de implementación de medidas de seguridad para proteger la página web o la aplicación contra posibles amenazas y vulnerabilidades.

Actores: Desarrollador de software, Administrador del sistema

Disparador: El desarrollador o administrador inicia el proceso de implementación de seguridad en la página web o la aplicación.

Precondiciones: Se ha identificado la necesidad de proteger la página web o la aplicación contra posibles amenazas y vulnerabilidades.

Postcondiciones: Se han implementado medidas de seguridad en la página web o la aplicación.

#### Flujo de Ejecución:

1. El desarrollador o administrador identifica las posibles amenazas y vulnerabilidades que podrían afectar a la página web o la aplicación.
2. Se implementan medidas de autenticación y autorización para controlar el acceso de los usuarios y proteger los datos sensibles.
3. Se utilizan técnicas de encriptación para proteger la comunicación y los datos almacenados.
4. Se implementan firewalls y sistemas de detección de intrusiones para proteger contra ataques externos.
5. Se realizan pruebas de seguridad periódicas para identificar posibles brechas y vulnerabilidades.
6. Se aplican parches de seguridad y actualizaciones para corregir posibles vulnerabilidades conocidas.

7. El caso de uso finaliza.

#### Caso de Uso: Crear una Navegación Clara

Descripción: Este caso de uso describe el proceso de diseño y desarrollo de una navegación clara y intuitiva para facilitar la interacción de los usuarios con la página web o la aplicación.

Actores: Diseñador de UX/UI, Desarrollador de software

Disparador: El diseñador o desarrollador inicia el proceso de creación de una navegación clara en la página web o la aplicación.

Precondiciones: Se ha identificado la necesidad de mejorar la navegación y la usabilidad de la página web o la aplicación.

Postcondiciones: Se ha creado una navegación clara y fácil de usar.

#### Flujo de Ejecución:

1. El diseñador realiza un análisis de las páginas y secciones de la página web o la aplicación para identificar los elementos de navegación necesarios.
2. Se definen las categorías y subcategorías de la navegación, así como las etiquetas y los íconos correspondientes.
3. Se crea un esquema de navegación visual que muestra la estructura y la relación entre las diferentes secciones y páginas.
4. Se desarrolla la navegación utilizando técnicas de diseño responsive para asegurar una

experiencia coherente en diferentes dispositivos.

5. Se realiza una prueba de usabilidad para evaluar la eficacia de la navegación y realizar ajustes si es necesario.

6. El desarrollador implementa la navegación en la página web o la aplicación, asegurándose de que sea funcional y fácil de usar.

7. Se realiza una prueba final de usabilidad para verificar la claridad y la eficiencia de la navegación.

8. El caso de uso finaliza.

#### Caso de Uso: Búsqueda de Efectos Especiales (Sonido y Partículas)

**Descripción:** Este caso de uso describe el proceso de búsqueda y selección de efectos especiales de sonido y partículas para su implementación en el juego.

**Actores:** Diseñador de sonido, Diseñador de efectos visuales, Desarrollador de software

**Disparador:** El diseñador de sonido o el diseñador de efectos visuales inicia el proceso de búsqueda de efectos especiales para el juego.

**Precondiciones:** Se ha identificado la necesidad de mejorar la experiencia del juego con efectos especiales de sonido y partículas.

**Postcondiciones:** Se han seleccionado los efectos especiales adecuados para su implementación en el juego.

#### Flujo de Ejecución:

1. El diseñador de sonido busca y selecciona efectos de sonido que se ajusten a las acciones y eventos del juego.

2. El diseñador de efectos visuales busca y selecciona efectos de partículas que complementen las acciones y eventos del juego.

3. Ambos diseñadores colaboran para asegurarse de que los efectos de sonido y partículas sean coherentes y se complementen entre sí.

4. Los diseñadores presentan las selecciones al desarrollador de software para su implementación.

5. El desarrollador de software integra los efectos de sonido y partículas en el juego según las especificaciones proporcionadas.

6. Se realiza una prueba de los efectos especiales para verificar su calidad y coherencia con el juego.

7. Los ajustes necesarios se realizan en caso de que los efectos no cumplan con las expectativas.

8. El caso de uso finaliza

#### Caso de Uso: Implementar Efectos Especiales en el Juego

**Descripción:** Este caso de uso describe el proceso de implementación de los efectos especiales seleccionados en el juego.

**Actores:** Desarrollador de software

**Disparador:** El desarrollador de software inicia el proceso de implementación de los efectos especiales en el juego.

**Precondiciones:** Se han seleccionado los efectos especiales adecuados para su implementación en el juego.

**Postcondiciones:** Los efectos especiales se han integrado en el juego.

**Flujo de Ejecución:**

1. El desarrollador de software recibe los efectos especiales seleccionados por los diseñadores.
2. El desarrollador de software identifica las secciones o eventos del juego donde se deben aplicar los efectos especiales.
3. Se implementa la lógica necesaria para activar los efectos especiales en los momentos adecuados del juego.
4. El desarrollador de software ajusta los parámetros de los efectos especiales según las necesidades del juego.
5. Se realiza una prueba del juego para verificar la correcta implementación de los efectos especiales.
6. Los ajustes necesarios se realizan en caso de que los efectos no cumplan con las expectativas.
7. El caso de uso finaliza.

**Caso de Uso: Revisar Coherencia de los Efectos Especiales**

**Descripción:** Este caso de uso describe el proceso de revisar que los efectos especiales implementados en el juego sean coherentes y se ajusten a la experiencia deseada.

**Actores:** Diseñador de sonido, Diseñador de efectos visuales

**Disparador:** Los diseñadores de sonido y efectos visuales revisan los efectos especiales implementados en el juego.

**Precondiciones:** Los efectos especiales han sido implementados en el juego.

**Postcondiciones:** Los efectos especiales son coherentes y se ajustan a la experiencia deseada.

**Flujo de Ejecución:**

1. Los diseñadores de sonido y efectos visuales analizan y evalúan los efectos especiales implementados en el juego.
2. Se verifican que los efectos de sonido se correspondan adecuadamente con las acciones y eventos del juego.
3. Se asegura que los efectos de partículas complementan y resaltan las acciones y eventos del juego.
4. Se realizan ajustes o modificaciones en los efectos especiales, si es necesario, para garantizar su coherencia y adecuación al juego.
5. Se realiza una prueba del juego para verificar la calidad y coherencia de los efectos especiales revisados.
6. Los ajustes necesarios se realizan en caso de que los efectos no cumplan con las expectativas.
7. El caso de uso finaliza.

**Caso de Uso: Diseñar Dificultad de Cada Nivel del Juego**

**Descripción:** Este caso de uso describe el proceso de diseño de la dificultad de cada nivel del juego.

**Actores:** Diseñador de niveles, Diseñador de juego

**Disparador:** El diseñador de niveles o el diseñador de juego inicia el proceso de diseño de la dificultad del juego.

**Precondiciones:** Se ha creado la estructura básica de los niveles del juego.

**Postcondiciones:** Se ha diseñado la dificultad adecuada para cada nivel del juego.

**Flujo de Ejecución:**

1. El diseñador de niveles analiza y establece los objetivos y desafíos de cada nivel.
2. Se determina la progresión de dificultad deseada a lo largo del juego.
3. Se ajustan los parámetros del juego, como velocidad, daño de enemigos, cantidad de obstáculos, entre otros, para crear la dificultad esperada en cada nivel.
4. Se realizan pruebas y ajustes iterativos para garantizar que la dificultad sea equilibrada y proporcionada.
5. El diseñador de niveles finaliza el diseño de la dificultad de cada nivel.
6. El caso de uso finaliza.

**Caso de Uso: Implementar Puntos de Control**

**Descripción:** Este caso de uso describe el proceso de implementación de puntos de control en el juego para permitir a los jugadores guardar su progreso y reiniciar desde un punto específico.

**Actores:** Diseñador de niveles, Desarrollador de software

**Disparador:** El diseñador de niveles o el desarrollador de software inicia el proceso de implementación de puntos de control en el juego.

**Precondiciones:** Se ha diseñado la estructura de niveles y la progresión del juego.

**Postcondiciones:** Los puntos de control se han implementado en los lugares adecuados del juego.

**Flujo de Ejecución:**

1. El diseñador de niveles identifica los puntos estratégicos en cada nivel donde se deben colocar los puntos de control.
2. El desarrollador de software implementa la lógica necesaria para guardar el progreso del jugador en los puntos de control.
3. Se realizan pruebas para verificar que los puntos de control funcionen correctamente y permitan reiniciar el juego desde el punto guardado.
4. Los ajustes necesarios se realizan en caso de que los puntos de control no cumplan con las expectativas.
5. El caso de uso finaliza

**Caso de Uso: Crear una IA Desafiante**

**Descripción:** Este caso de uso describe el proceso de creación de una inteligencia artificial (IA) desafiante para los enemigos o adversarios del juego.

**Actores:** Diseñador de IA, Diseñador de niveles, Desarrollador de software

**Disparador:** El diseñador de IA o el diseñador de niveles inicia el proceso de creación de la IA desafiante.

**Precondiciones:** Se ha diseñado la estructura de niveles y se han establecido las reglas de juego.

**Postcondiciones:** Se ha creado una IA desafiante que brinda un nivel adecuado de dificultad para los jugadores.

**Flujo de Ejecución:**

1. El diseñador de IA define los comportamientos y estrategias de la IA enemiga para que sean desafiantes para los jugadores.
2. El diseñador de niveles incorpora enemigos con la IA desafiante en los niveles correspondientes.
3. El desarrollador de software implementa la lógica necesaria para que la IA enemiga funcione correctamente.
4. Se realizan pruebas para verificar que la IA enemiga presente un desafío adecuado y se comporte de manera coherente.
5. Los ajustes necesarios se realizan en caso de que la IA no cumpla con el nivel de desafío deseado.
6. El caso de uso finaliza.

#### Caso de Uso: Crear una Nueva Cuenta de Usuario

Descripción: Este caso de uso describe el proceso de creación de una nueva cuenta de usuario en el sistema.

Actores: Usuario, Sistema

Disparador: El usuario desea crear una nueva cuenta en el sistema.

Precondiciones: El usuario ha ingresado al sistema y se encuentra en la pantalla de creación de cuenta.

Postcondiciones: Se ha creado exitosamente una nueva cuenta de usuario en el sistema.

Flujo de Ejecución:

1. El usuario completa el formulario de registro con la información requerida, como nombre, dirección de correo electrónico y contraseña.
2. El sistema valida los datos ingresados por el usuario, verificando que la dirección de correo electrónico no esté registrada previamente y que la contraseña cumpla con los requisitos de seguridad.
3. Si los datos son válidos, el sistema crea una nueva cuenta de usuario y almacena la información proporcionada.
4. El sistema muestra un mensaje de confirmación de que la cuenta se ha creado exitosamente.
5. El caso de uso finaliza

#### Caso de Uso: Desplegar Información de la Cuenta

Descripción: Este caso de uso describe el proceso de mostrar la información de la cuenta de usuario en el sistema.

Actores: Usuario, Sistema

Disparador: El usuario desea ver la información de su cuenta en el sistema.

Precondiciones: El usuario ha iniciado sesión correctamente en el sistema.

Postcondiciones: El sistema muestra la información de la cuenta del usuario en pantalla.

Flujo de Ejecución:

1. El usuario selecciona la opción "Ver información de la cuenta" en la interfaz del sistema.
2. El sistema recupera la información de la cuenta del usuario, como nombre, dirección de correo electrónico y fecha de creación.

- |   |
|---|
| 3. El sistema muestra la información de la cuenta del usuario en pantalla de manera legible y organizada. |
| 4. El usuario puede visualizar y verificar la información de su cuenta.                                   |
| 5. El caso de uso finaliza  |

**Caso de Uso: Crear Estadísticas de la Cuenta**

**Descripción:** Este caso de uso describe el proceso de creación y actualización de las estadísticas de la cuenta del usuario.

**Actores:** Sistema, Juego

**Disparador:** El sistema registra las acciones y eventos del juego para generar estadísticas de la cuenta del usuario.

**Precondiciones:** El usuario ha jugado al menos una partida en el juego.

**Postcondiciones:** Se crean y actualizan las estadísticas de la cuenta del usuario.

**Flujo de Ejecución:**

1. El sistema registra las acciones y eventos relevantes del juego, como puntuación, tiempo jugado, logros desbloqueados, entre otros.
2. El sistema procesa y calcula las estadísticas basadas en los datos registrados.
3. Las estadísticas se almacenan en la cuenta del usuario y se actualizan según corresponda.
4. El usuario puede acceder a las estadísticas desde su perfil o sección correspondiente.
5. El caso de uso finaliza

**Caso de Uso: Implementar Características Especiales**

**Descripción:** Este caso de uso describe el proceso de implementación de características especiales en la cuenta de usuario.

**Actores:** Usuario, Sistema

**Disparador:** El usuario solicita activar o utilizar características especiales en su cuenta.

**Precondiciones:** El usuario tiene acceso a características especiales disponibles en el sistema.

**Postcondiciones:** Las características especiales se activan y están disponibles para el usuario.

**Flujo de Ejecución:**

1. El usuario accede a la sección de configuración o características especiales de su cuenta.
2. El usuario selecciona las características especiales que desea activar o utilizar, como avatares personalizados, temas exclusivos, bonificaciones, etc.
3. El sistema verifica la disponibilidad y los requisitos necesarios para activar las características seleccionadas.
4. Si las características son válidas, el sistema las activa y las vincula a la cuenta del usuario.
5. El usuario puede utilizar y disfrutar de las características especiales activadas.
6. El caso de uso finaliza

## **Base de Datos:**

En este apartado mostramos la creación de nuestro esquema de entidad relación normalizado hasta la tercera forma normal, la cual nos permite reducir la ambigüedad de las tablas y registro repetidos para obtener un mejor análisis como entendimiento de los diferentes atributos de cada una de las entidades.

La tercera forma normal (3NF) en una base de datos se cumple cuando se cumplen dos condiciones:

La tabla está en segunda forma normal (2NF).

No hay dependencias transitivas.

Para justificar que las tablas están en tercera forma normal, debemos evaluar cada una de ellas en función de estas condiciones.

Tabla "usuarios":

Cumple con la segunda forma normal (2NF) porque no hay dependencias parciales en sus atributos. Todos los atributos dependen completamente de la clave primaria "id\_usuario".

No hay dependencias transitivas, ya que no existen dependencias indirectas entre los atributos. Por lo tanto, cumple con la tercera forma normal.

Tabla "personaje":

Cumple con la segunda forma normal (2NF) porque no hay dependencias parciales en sus atributos. Todos los atributos dependen completamente de la clave primaria "id\_personaje".

No hay dependencias transitivas, ya que no existen dependencias indirectas entre los atributos. Por lo tanto, cumple con la tercera forma normal.

Tabla "inventario":

Cumple con la segunda forma normal (2NF) porque no hay dependencias parciales en sus atributos. Todos los atributos dependen completamente de la clave primaria "id\_inventario".

No hay dependencias transitivas, ya que no existen dependencias indirectas entre los atributos. Por lo tanto, cumple con la tercera forma normal.

Tabla "nivel":

Cumple con la segunda forma normal (2NF) porque no hay dependencias parciales en sus atributos. Todos los atributos dependen completamente de la clave primaria "id\_nivel".

No hay dependencias transitivas, ya que no existen dependencias indirectas entre los atributos. Por lo tanto, cumple con la tercera forma normal.

Tabla "checkpoints":

Cumple con la segunda forma normal (2NF) porque no hay dependencias parciales en sus atributos. Todos los atributos dependen completamente de las claves primarias "id\_checkpoint", "id\_usuario", "id\_personaje", "id\_inventario" e "id\_nivel".

No hay dependencias transitivas, ya que no existen dependencias indirectas entre los atributos. Por lo tanto, cumple con la tercera forma normal.

Tabla "enemigos":

Cumple con la segunda forma normal (2NF) porque no hay dependencias parciales en sus atributos. Todos los atributos dependen completamente de la clave primaria "id\_enemigo". No hay dependencias transitivas, ya que no existen dependencias indirectas entre los atributos. Por lo tanto, cumple con la tercera forma normal.

Tabla "enemigos\_nombres":

Cumple con la segunda forma normal (2NF) porque no hay dependencias parciales en sus atributos. Todos los atributos dependen completamente de las claves primarias "id\_enemigo" y "nombre\_enemigo".

No hay dependencias transitivas, ya que no existen dependencias indirectas entre los atributos. Por lo tanto, cumple con la tercera forma normal.

