

D&D Master

Enginyeria de Requisits

Quadrimestre Tardor 2022/2023

Grup 22.1

Lolita Kudryavtseva

Marc Castro Chavez

Lluís Pujalte Feliu de Cabrera

Sergio Guerrero Rivero

Paula Barrachina Cáceres

Índice

1. Visión del proyecto	3
2. Partes interesadas	5
3. Objetivos	7
4. Visión general de la solución propuesta	8
4.1. Estructura	8
4.2. Usuarios Principales	8
4.3. Usos Principales	9
4.4. Relación con otros sistemas	9
4.5. Mockups	10
5. Características del sistema	13
5.1. Requisitos Funcionales	13
5.1.1. Área funcional de gestión de perfiles	15
5.1.2. Área funcional de gestión de partidas	16
5.1.3. Área funcional de gestión de personajes	16
5.1.4. Área funcional de gestión de acciones en la partida	17
5.2. Requisitos no Funcionales (de calidad)	18
6. Esquema conceptual	22
6.1. Esquema conceptual de los datos	22
6.1.1. Esquema	22
6.1.2. Restricciones textuales:	23
6.1.3. Glosario	23
6.2. Esquema del comportamiento	25
7. Argumentación del logro de los objetivos	40
8. Glossario	41

1. Visión del proyecto

A continuación se reunirán las capacidades y necesidades de nuestro sistema D&D Master.

La idea de D&D Master nace con el objetivo de solucionar diversos problemas que los jugadores del famoso juego de rol *Dungeons and Dragons* se han encontrado durante sus partidas. El sistema pretende proponer soluciones a estos inconvenientes, además de proporcionar a los jugadores una herramienta útil durante sus partidas que les hará disfrutar del juego de una forma más inmersiva y dinámica.

El principal problema que soluciona D&D Master es la dificultad para encontrar jugadores para interpretar el papel de *Dungeon Master* en las partidas. La solución propuesta por la plataforma es la parte central de nuestro sistema, que consiste en un *Digital Human* que interprete este rol dentro de la partida. El *Digital Human* se adaptará a las necesidades de la mesa y podrá ir leyendo las reacciones de los jugadores y actuando en respuesta a estas. Además, será fácilmente personalizable, ya que ofrecerá diversos aspectos, voces y modos para que los jugadores puedan encontrar el mejor compañero para sus partidas.

También será capaz de mostrar apoyo visual a los jugadores como por ejemplo mapas, imágenes, entre otras cosas, lo que ayudará a la inmersión de los participantes en la partida. Este apoyo visual vendrá dado mediante la asociación con la plataforma de generación de imágenes creadas por una *IA* denominada *DAL.LE*. Integrará también distintas herramientas para hacer las partidas más fáciles, como por ejemplo poder ver las fichas de los jugadores en todo momento, lanzar dados,...

Otra característica muy importante de D&D Master es que facilitará la iniciación de nuevos jugadores en este juego, ya que mediante un modo tutorial será capaz de guiar a los aventureros novatos en sus primeros pasos dentro del mundo de *Dungeons and Dragons*. Esto pretende generar un aumento de la base de jugadores del juego de rol y llegar a más tipos de gente que de otra forma hubiera sido difícil de hacerlo.

Un añadido de este sistema que ayudará a aumentar esta comunidad de jugadores y facilitará una comodidad a la hora de jugar a Dungeons and Dragons es la implementación de un modo online dado lugar debido a la integración de la plataforma *Roll20* en el sistema. De esta forma se podrán llevar a cabo partidas con jugadores de cualquier parte del mundo y en cualquier momento de una forma fácil y sencilla.

Todo el sistema vendrá encapsulado en una aplicación para ordenador que requerirá del uso de una webcam con una calidad mínima para poder funcionar, un micrófono y los datos de las diferentes partidas irán almacenados en una base de datos para así mejorar el sistema aprendiendo de partidas anteriores.

2. Partes interesadas

★ Jugadores de rol

- Rol: Hacer uso de la plataforma para jugar a *D&D*.
- Objetivos:
 - Poder jugar a *Dungeons and Dragons* con una gran inmersión en la partida. Tener un *DM* que sea un *Digital Human*, permitirá a los jugadores experimentar el juego de una forma más visual y auditiva.
 - Ser capaces de jugar sin la necesidad de un DM.
 - Jugar online con otros jugadores y ser capaces de conocer fanáticos nuevos de este juego
 - Disponer de una variedad más amplia de ambientaciones para sus partidas.

★ *Wizards of the coast*

- Rol: Aportar las licencias del producto e invertir en el proceso de desarrollo o aportar soporte técnico.
- Objetivos:
 - Promocionar su marca y productos para llegar a nuevas personas y así crecer la base de jugadores de D&D
 - Introducir una nueva manera de jugar más inmersiva
 - Aumentar las ventas
 - Ampliar su rango de mercado gracias a nuestra plataforma

★ Gente que quiera empezar a jugar

- Rol: Hacer uso de la plataforma para aprender a jugar a D&D
- Objetivos:
 - Aprender de manera más sencilla y autónoma a jugar a D&D.
 - Jugar online.

★ Jugadores puristas

- Rol: -
- Objetivo:
 - Que no se use la plataforma y el juego mantenga su purismo.

★ Expertos en interfaces

- Rol: Asegurarse de que la interfaz de la plataforma cumple los estándares requeridos.
- Objetivo:
 - Desarrollar una interfaz para la plataforma.
 - Hacer una interfaz útil, fiable y estética.

★ Vendedores de tiendas especializadas

- Rol: Dar a conocer la plataforma.
- Objetivo:
 - Aumentar la venta de productos de rol.
 - Dar más visibilidad al juego.

★ Ilustradores

- Rol: Realizar las ilustraciones de la plataforma.
- Objetivo:
 - Que sus ilustraciones se den a conocer gracias a la plataforma.

★ Expertos en *Digital Humans*

- Rol: Gestionar la parte del desarrollo de los *Digital Humans*.
- Objetivo:
 - Dar visibilidad a la tecnología.
 - Ver como se desarrolla la tecnología y cómo responde al proyecto.

3. Objetivos

3.1 Clientes

- Gracias al lanzamiento de la aplicación reducir el tiempo medio gastado¹ en la preparación de la partida en 50%.
- Desde la salida de la aplicación poder empezar una partida con otros jugadores siempre que se quiera, sin necesidad de buscar un Dungeon Master.

3.2 Administración

- Nuestro objetivo principal como administradores del sistema es que pasado el primer año tras el lanzamiento el número de clientes aumente un 10%.
- Otro punto importante a conseguir es que durante los 3 primeros meses después del lanzamiento de la plataforma, al menos un 70% de los usuarios del sistema hayan finalizado al menos una partida.
- Se consigue una mejora en el sistema de reacciones del *Digital Human* de un 20% a partir del primer año después del lanzamiento de la plataforma. (Sergio)

3.3 Inversores

- Pasados los 3 años desde el lanzamiento de Digital Dungeon Master se han rentabilizado los costes de inversión iniciales.

¹ El tiempo promedio: la mayoría de los jugadores tardan una hora o mas
https://slyflourish.com/how_long_to_prep.html

4. Visión general de la solución propuesta

En este apartado describimos brevemente el sistema propuesto indicando su estructura, usos principales y relaciones que tendrá con otros sistemas. Incluyendo Mockups para representar gráficamente nuestro sistema.

4.1. Estructura

La estructura principal de Digital Dungeon Master consta de una aplicación para ordenador que requerirá del uso de una webcam para poder funcionar, además de un micrófono.

Los usuarios interactuarán directamente con un *Digital Human* que hará el papel de Dungeon Master lo cual hará posible que la partida sea más inmersiva y dinámica. Para realizar esta interacción los usuarios mantendrán una conversación con el *Digital Human* que mediante la webcam y el micrófono podrá leer las reacciones de estos y responder consecuentemente. Además toda interacción con el sistema se podrá efectuar mediante una conversación con el *Digital Human*, de esta forma los usuarios podrán configurar la aplicación a su gusto, cambiar los ajustes, crear las fichas de los personajes entre otras cosas sin necesidad del uso de otros periféricos como teclado o ratón.

El *Digital Human* por su parte, tratará de adaptarse al grupo de jugadores para así hacer posible una aventura personalizada en todo momento. Para hacer esto el sistema capturará las reacciones de los usuarios a las distintas situaciones e irá aprendiendo las preferencias de los mismos.

4.2. Usuarios Principales

Los usuarios principales de nuestro sistema serán los jugadores de Dungeons and Dragons. Que podrán registrarse y jugar a nuestra aplicación en la página web del sistema.

Entre ellos jugadores expertos con conocimientos previos del juego y jugadores nuevos atraídos por la aplicación o con interés en D&D. Estos nuevos jugadores podrán aprender de manera autónoma e intuitiva como jugar de una manera más sencilla. Todo gracias a los tutoriales incluidos en el juego y la ayuda del *Digital Human*.

4.3. Usos Principales

El uso principal de Digital Dungeon Master será realizar partidas de *Dungeons and Dragons* sin necesidad de un DM.

El sistema engloba todo tipo de funcionalidades útiles para una partida de *D&D* para satisfacer las necesidades de los jugadores. Proporcionará un fácil acceso a las *fichas de personajes* y las modificará a medida que la partida vaya avanzando. Además mostrará soporte visual a los jugadores que consta de mapas, imágenes de objetos, localizaciones,... En adición a esto el sistema moverá las distintas piezas que representarán a los personajes en los mapas previamente mencionados. Otra funcionalidad importante es el uso de dados que permitirá elegir entre tirarlos digitalmente o físicamente. Para las tiradas físicas reconocerá los resultados obtenidos mediante la webcam.

Además de ser capaz de reaccionar a las situaciones de la partida consecuentemente, el *Digital Human* podrá actuar como distintos *NPCs* y cambiar su forma (vestimenta, cara, voz,...) para hacer la situación más inmersiva.

Otra de las ventajas de nuestra plataforma es que permitirá jugar online a los jugadores.

4.4. Relación con otros sistemas

Nuestra aplicación interactuará con diversos sistemas y guardará toda la información de los jugadores y partidas en nuestra base de datos.

El sistema principal del cual hará uso nuestra aplicación es Roll20, el cual permitirá mostrar el tablero por pantalla, modificar las fichas de los personajes y jugar online con otros jugadores además de llevar integrado al *Digital Human*.

Para dar soporte visual a la partida, el *Digital Human* interactuara con DAL.LE, sistema que permite la generación de imágenes a partir de una IA. Estas imágenes serán mostradas por pantalla para que los usuarios puedan verlas.

Para la comunicación entre los jugadores en un juego online, se podrá usar *discord* el cual estará integrado en la aplicación, permitiendo al *Digital Human* y a los jugadores comunicarse durante la partida mediante videollamada.

4.5. Mockups

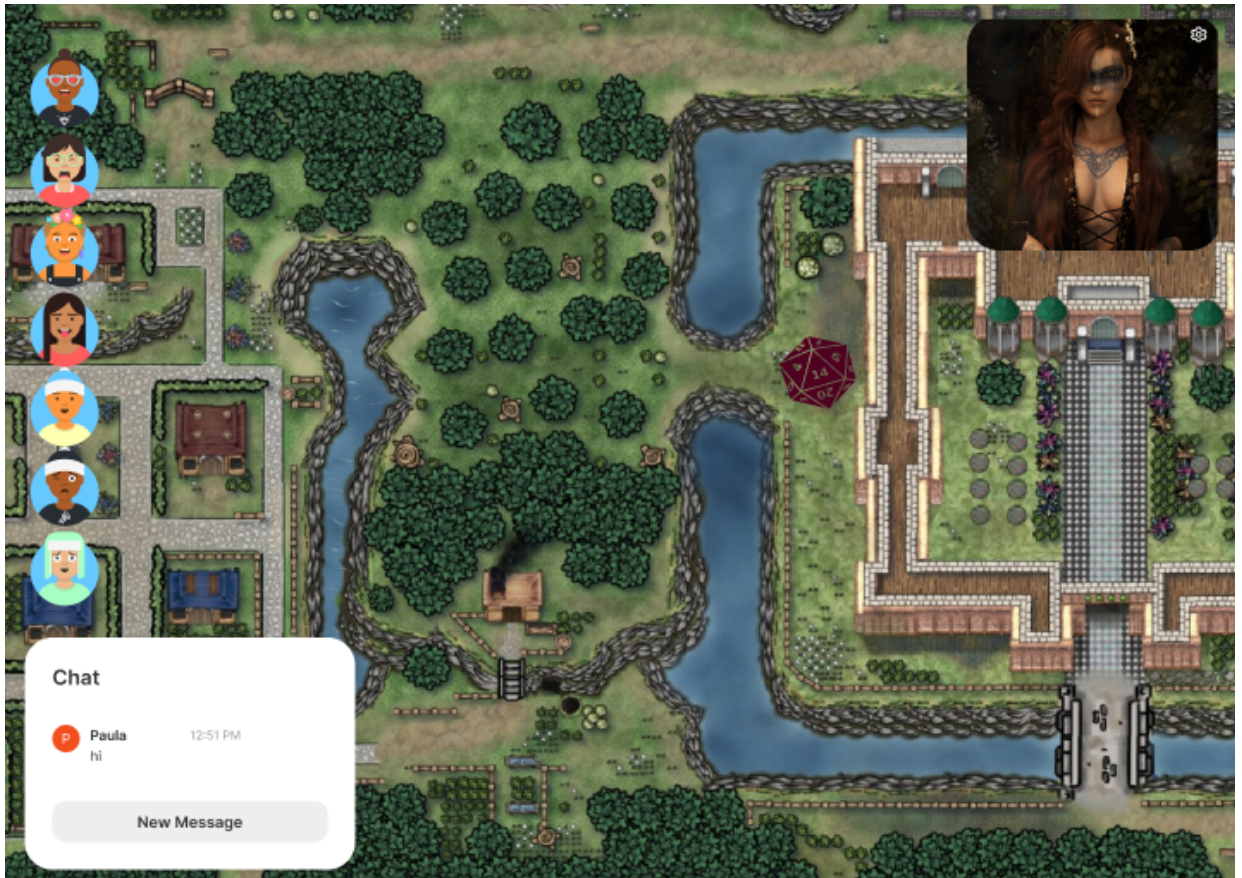


Figura 1. Mockup de una partida en curso.

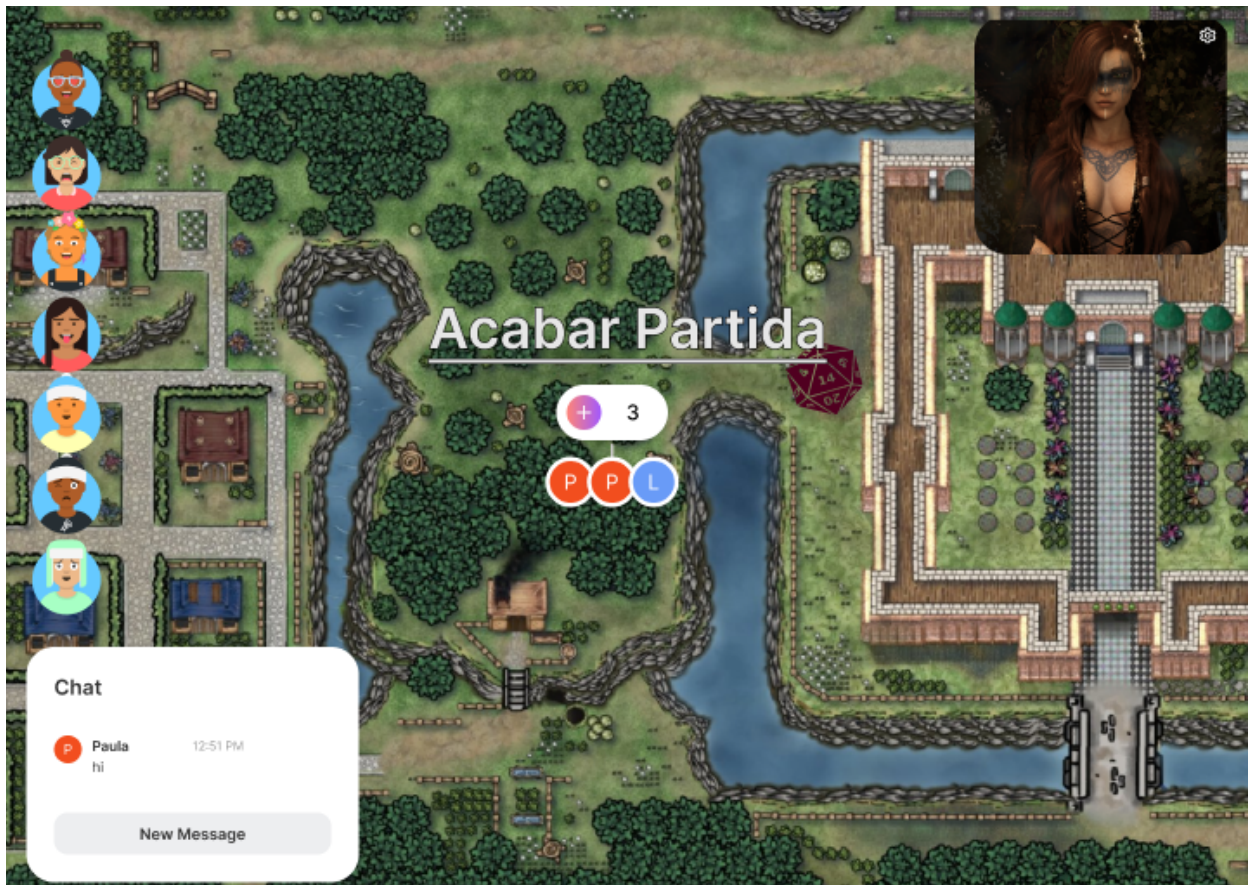


Figura 2. Mockup de la votación de final de partida.

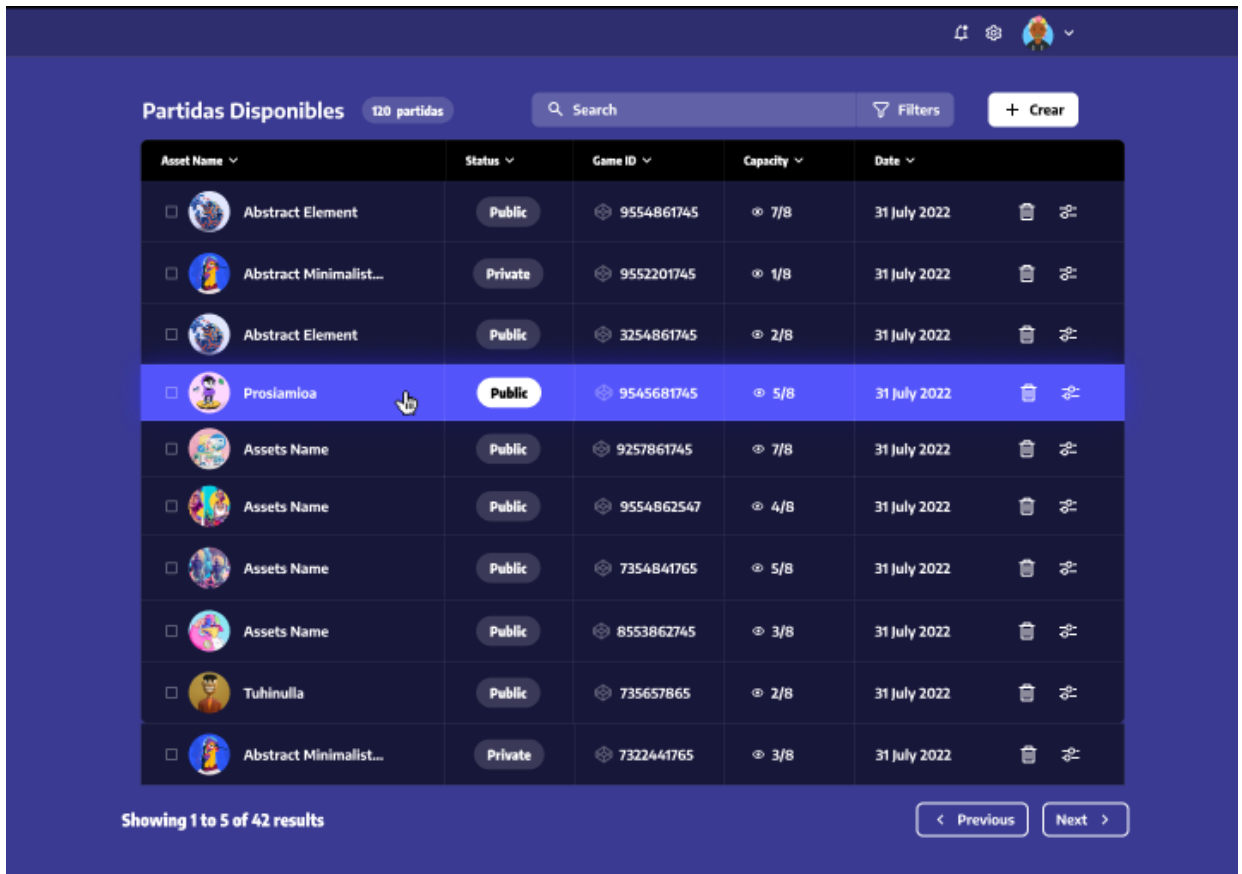


Figura 3. Mockup de la interfaz para buscar partidas.

5. Características del sistema

5.1. Requisitos Funcionales

En este apartado indicamos todos los casos de uso de nuestro sistema e incluimos el diagrama de estos. Además de una breve descripción de cada uno.

Diagrama de casos d'ús amb totes les funcionalitats del sistema:

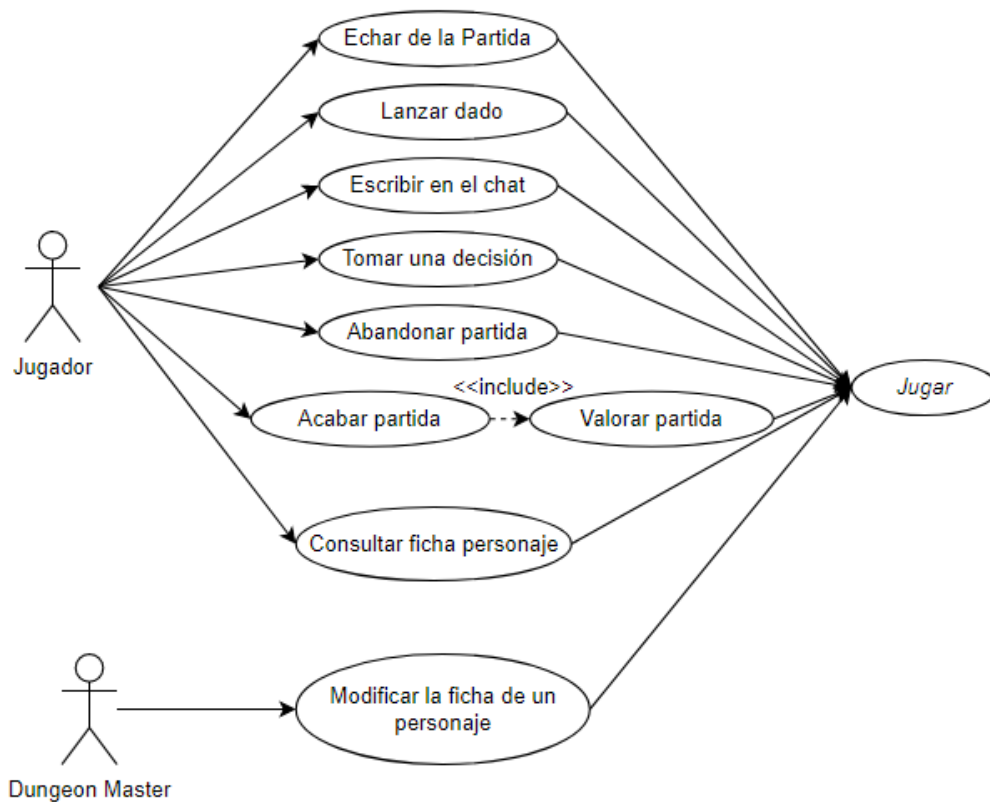


Figura 4. Diagrama casos de uso acciones en la partida.

Como se puede observar en la *Figura 4*, el caso de uso Valorar Partida está incluido en el de terminarla. Esto se debe a que en nuestro sistema los usuarios están obligados a valorar la partida una vez la terminan.

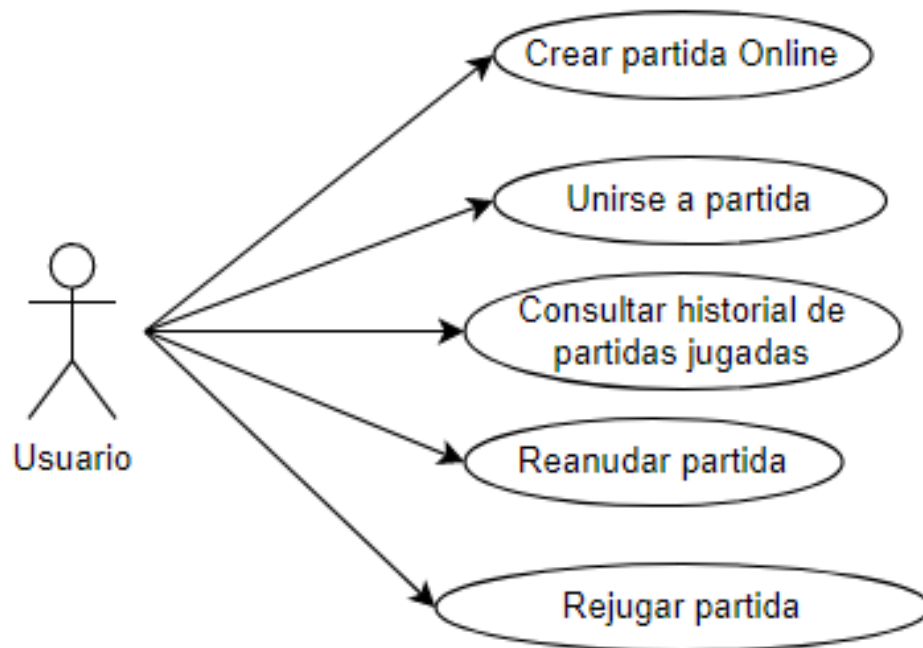


Figura 5. Diagrama casos de uso gestión de partidas.

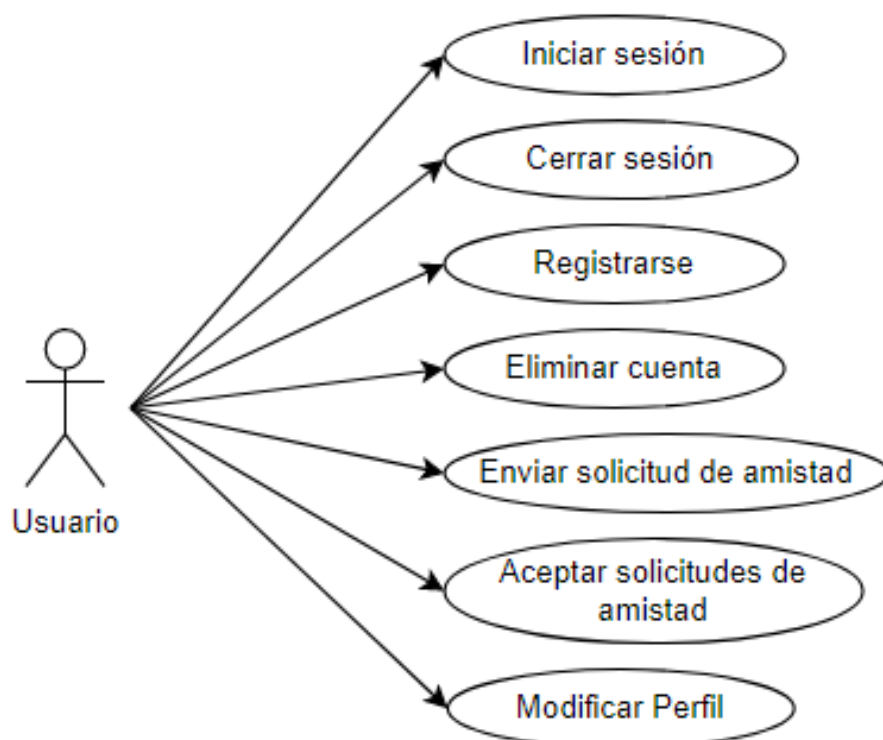


Figura 6. Diagrama casos de uso gestión de perfil.

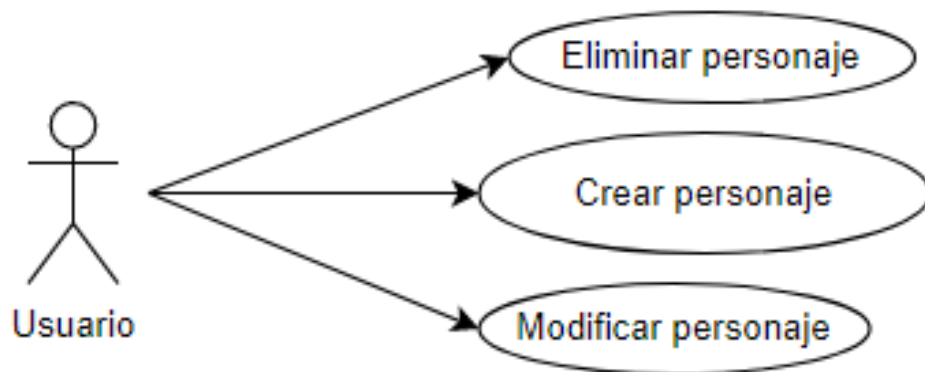


Figura 7. Diagrama casos de uso gestión de personajes.

5.1.1. Área funcional de gestión de perfiles

Iniciar sesión: Cuando un Usuario quiere iniciar sesión debe indicar su nombre de usuario y su contraseña.

Cerrar sesión: Cuando un Usuario quiere cerrar sesión de la plataforma debe indicarlo.

Registrarse: Para que un usuario no registrado pueda crearse una cuenta deberá indicar los siguientes datos: Nombre de usuario, contraseña, número de teléfono y edad.

Eliminar cuenta: Cuando un usuario registrado quiera eliminar su cuenta deberá indicarlo e introducir su contraseña. Seguidamente le llegará un mail en el que deberá confirmar la eliminación de la cuenta.

Enviar solicitud de amistad: Para añadir un amigo el usuario deberá indicar el nombre del usuario al que quiere añadir como amigo y finalmente enviarle una solicitud.

Aceptar solicitudes de amistad: Para aceptar solicitudes de amistad un usuario deberá previamente haberla recibido y finalmente presionar el botón para aceptarla.

Modificar perfil: Para modificar el perfil del usuario, este deberá indicar qué datos del perfil desea modificar, finalmente antes de guardar los cambios tendrá que indicar su contraseña y confirmarlos.

5.1.2. Área funcional de gestión de partidas

Crear partida online: Para crear una partida online se deberá indicar el tipo de partida a crear en este caso online, e indicar si se quiere hacer una partida privada o pública. Además se deberá indicar la *skin* que se desea usar para el dungeon master, la restricción de edad y el idioma en el que se desea jugar. Finalmente seleccionara el personaje que querrá usar para la partida.

Crear partida local: Para crear una partida local el usuario deberá crear una partida e indicar el tipo de partida, local. Seguidamente indicará el número de jugadores y los personajes que usará cada uno. Además de escoger el idioma, la skin del *Digital Human* y la restricción de edad.

Unirse a partida: Para unirse a una partida se deberá indicar la partida a la que el usuario se quiere unir. En la pantalla de unirse a partida aparecerán dos secciones, una para las partidas creadas por amigos, privadas, y otra para las partidas públicas.

Eliminar partida: Para eliminar una de las partidas de las que el usuario es creador deberá seleccionar la partida y confirmar su eliminación.

Consultar historial de partidas jugadas: El usuario podrá consultar el historial de partidas en la pantalla adecuada. En ella aparecerá un listado de todas las partidas que ha jugado con anterioridad.

Reanudar partida: Para continuar con una partida que ya esté empezada el usuario deberá seleccionar la partida que quiere reanudar.

Rejugar partida: Para rejugar una partida el usuario deberá consultar el historial de partidas jugadas e indicar que partida quiere rejugar, a continuación se creará una partida con los ajustes de la partida seleccionada.

5.1.3. Área funcional de gestión de personajes

Crear personaje: Para crear un personaje el usuario deberá indicar el nombre del personaje, las *estadísticas*, la *clase* y el trasfondo(*background*).

Modificar personaje: Para modificar un personaje desde fuera de una partida, el usuario deberá seleccionar el personaje que quiere modificar. Solo se podrán modificar los personajes que no están asociados a ninguna partida o que estén asociados a una partida terminada.

Eliminar personaje: Para eliminar uno de los personajes que tiene un usuario, este deberá seleccionar el personaje y borrarlo.

5.1.4. Área funcional de gestión de acciones en la partida

Echar de la partida: Para echar a un jugador de la partida uno de los usuarios deberá iniciar una votación para echar a uno de los jugadores y el resto votará para tomar la decisión. Si sale mayoría el jugador seleccionado será expulsado de la misma.

Lanzar dado: Por pantalla se notificará al jugador en cuestión su turno para tirar los dados y que dado deberá usar. El jugador deberá arrastrar el dado por pantalla para tirarlo.

Escribir en el chat: El jugador podrá escribir un mensaje y enviarlo por el chat grupal de la partida que esté en curso.

Tomar una decisión: Cuando la historia requiera que uno de los jugadores tome una decisión o uno de los jugadores quiera intervenir deberá expresar verbalmente la acción a realizar.

Abandonar partida: Cuando un jugador quiera abandonar una partida deberá indicarlo. Cuando uno de los jugadores abandone la partida, esta seguirá su curso y el *Digital Human* adaptará la historia al abandono de este. El jugador que ha abandonado la partida podrá unirse en otro momento.

Acabar partida: Cuando se quiera terminar la partida se realizará una votación y si sale mayoría el *Digital Human* pondrá un final a la historia.

Modificar la ficha de un personaje: Cuando la partida lo requiera, el *Digital Human* podrá modificar la ficha de un personaje.

Consultar la ficha de un personaje: El usuario deberá indicar el nombre del personaje del cual quiera consultar la ficha y el sistema le devolverá los *stats*, el *background* y la *clase*.

Valorar partida: Una vez la partida sea finalizada por todos los jugadores, una pantalla emergente aparecerá para realizar una encuesta particular de la cual se hará una media para obtener la valoración total de la partida.

5.2. Requisitos no Funcionales (de calidad)

Requisitos de percepción (Look and Feel):

Número	1
Tipo de Requisito	10a. Apariencia
Descripción	El producto ha de ser atractivo para los clientes que lo utilicen, además su estética debe coincidir con la de la empresa.
Justificación del requisito	Un buen diseño hace más atractiva la aplicación haciendo también que haya más personas interesadas en usarla.
Condición de satisfacción	Se cogerá un grupo de entre 50 y 100 personas y se les hará una encuesta sobre la estética de la aplicación y una nota numérica entre 1 y 10. La media de los resultados ha de estar por encima del 7,5.

Requisitos de usabilidad y humanidad:

Número	2
Tipo de Requisito	11a. Facilidad de uso
Descripción	El producto tiene que ser fácil de utilizar para que cualquier tipo de usuario pueda utilizarlo sin problemas.
Justificación del requisito	Un sistema con fácil interacción hace que los usuarios tengan una experiencia agradable y quieran seguir utilizando el sistema.
Condición de satisfacción	Los usuarios han de poder utilizar el sistema sin necesidad de conocimientos previos sobre sistemas similares. La tasa de errores ha de ser igual o menor al 5%. Se realizará una encuesta con nota numérica entre el 1 y el 10 sobre lo fácil de usar que es el sistema. Cogeremos un grupo entre 50 y 100 personas y se hará la media de las valoraciones, dicha media tiene que estar por encima del 7,5.

Número	3
Tipo de Requisito	11c. Aprendizaje
Descripción	El producto tiene que ser fácil de aprender para cualquier usuario.
Justificación del requisito	Pueden haber usuarios que nunca hayan jugado al Dungeons and Dragons o usado un sistema similar al nuestro. Esos usuarios deben ser capaces de aprender a usar nuestro producto de forma rápida para evitar frustraciones y que no vuelvan a utilizar el sistema.
Condición de satisfacción	Los usuarios no necesitarán unos conocimientos previos antes de usar nuestro sistema.

Requisitos de rendimiento:

Número	4
Tipo de Requisito	12a. Velocidad y latencia
Descripción	El sistema responderá rápidamente a las acciones de los usuarios sin retardo.
Justificación del requisito	Una partida tiene que ser fluida y sin retardos para evitar que los jugadores se cansen y frustren esperando a que el sistema acabe de ejecutar sus acciones.
Condición de satisfacción	Cualquier acción que realice un usuario tarda menos de 1 segundo en ejecutarse en el 90% de los casos.

Número	5
Tipo de Requisito	11e. Robusteza
Descripción	El producto ha de continuar funcionando en circunstancias anormales o extremas.
Justificación del requisito	El sistema tiene que proporcionar sus servicios durante y después de un evento anormal. Además, tiene que garantizar la recuperación de los datos ante cualquier problema con el sistema.
Condición de satisfacción	El sistema ha de realizar una copia de seguridad diaria de los datos. El sistema tiene que tener backups para ser utilizados en caso de que el sistema principal falle.

Número	6
Tipo de Requisito	12h. Longevidad
Descripción	El producto tiene que estar disponible en el tiempo.
Justificación del requisito	El sistema ha de estar en funcionamiento y disponible un tiempo mínimo para estimular el crecimiento de este, dado que si el sistema está caído puede frustrar a los jugadores que quieran acceder. Además, hay que realizar actualizaciones para adaptarlo a nuevos usuarios, nuevos requisitos de los sistemas operativos o nuevos requisitos.
Condición de satisfacción	El producto se mantendrá disponible y actualizado continuamente sin interrupciones durante 5 años.

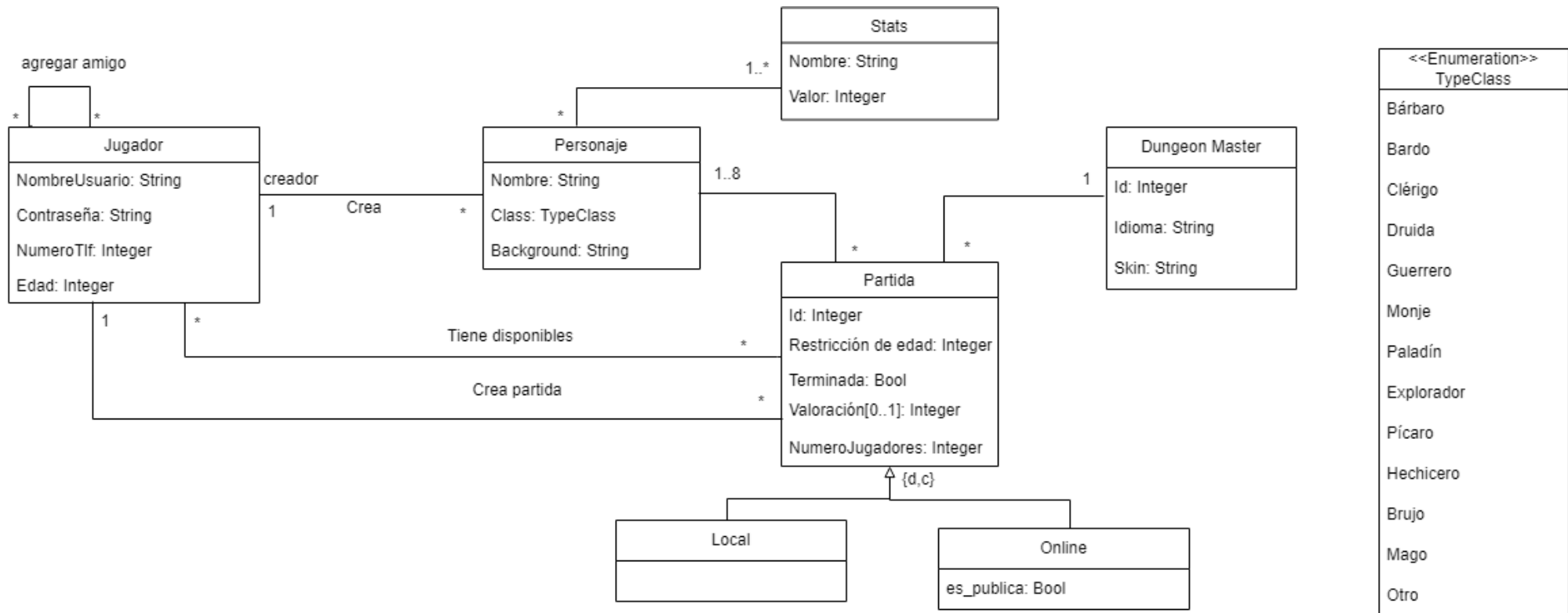
Requisitos operativos y de entorno:

Número	7
Tipo de Requisito	13c. Interacción con sistemas adyacentes
Descripción	Las interacciones con las aplicaciones con las que estamos asociados han de funcionar correctamente.
Justificación del requisito	El sistema funciona principalmente con Roll20, si esa interacción fallará, sería imposible jugar una partida.
Condición de satisfacción	Las interacciones entre el producto y los sistemas asociados funcionaran correctamente.

6. Esquema conceptual

6.1. Esquema conceptual de los datos

6.1.1. Esquema



6.1.2. Restricciones textuales:

1. Claves externas: (Usuario, NombreUsuario), (Personaje, Nombre), (Partida, Id), (Stats, Nombre + Personaje,Nombre) (Dungeon Master, Id) .
2. Una partida solo puede tener valoración si el booleano Terminada tiene valor true.
3. Un usuario solo puede jugar a partidas que tenga disponibles.
4. Una partida online que sea privada, solo aparecerá disponible a aquellos usuarios que sean amigos del usuario creador.
5. Una partida online que sea pública aparecerá disponible a todos los usuarios
6. Un usuario no puede añadirse como amigo a si mismo
7. Un usuario no puede unirse a una partida si la restricción de edad de la partida es superior a su edad.
8. El usuario creador de la partida debe ser uno de los jugadores

6.1.3 Glosario

En este apartado explicamos brevemente los conceptos del diagrama que són específicos de nuestro sistema o puede dar pie a confusiones.

Descripción de clases:

Personaje: Cada jugador de la partida tiene un personaje con una clase, estadísticas,... con el juega la partida.

PartidaLocal: Una partida local está formada por jugadores que se encuentran físicamente en el mismo sitio y utilizan un solo dispositivo para acceder a la plataforma. Además, los personajes de cada jugador deberán estar almacenados en la cuenta del usuario que crea la partida.

PartidaOnline: Una partida online está formada por jugadores que usan un dispositivo por usuario para acceder a la plataforma. En este caso, cada usuario deberá tener su personaje correspondiente en la partida almacenado en su cuenta.

Descripción de asociaciones:

Privacidad: Una partida online puede ser privada o pública.

Tiene disponibles: Un usuario tendrá disponibles todas aquellas partidas a las que pueda unirse o esté jugando. Si una partida es de privacidad pública, está disponible para todos los usuarios de la plataforma. Por otra parte las partidas privadas, solo aparecerán disponibles a aquellos usuarios que sean amigos del usuario creador de la partida.

El jugador podrá ver toda la información de las partidas que tenga disponibles, para decidir si entrar o no.

Añadir amigo: Un usuario tendrá una lista de amigos con los que podrá crear partidas online privadas

Descripción de atributos:

Class: Cada personaje en una partida tiene una clase.

Stats: Las estadísticas de un personaje.

Background: Trasfondo del personaje.

6.2. Esquema del comportamiento

1. Enviar solicitud de amistad

Actor principal	Jugador
Precondición	1. El Jugador está registrado en el sistema.
Disparador	El jugador quiere enviar una solicitud de amistad a otro jugador.
Escenario principal del éxito	<ol style="list-style-type: none">1. El usuario indica que quiere añadir una persona a su lista de amigos.2. Introduce un nombre para buscarlo en el sistema.3. Confirma que el resultado de búsqueda es el perfil correcto.4. El usuario envía friend request.
Extensiones	<p>2a. El nombre introducido no corresponde a ningún usuario. 2a1. El sistema informa al usuario que la búsqueda no ha producido resultados.</p> <p>3a. El nombre introducido es de un usuario del cual ya es amigo.</p>

Feature: Enviar solicitud de amistad

Como usuario de la aplicación que está logueado, **quiero poder** enviar solicitudes de amistad a otros usuarios **para poder** agregarlos y crear partidas con ellos.

Criterios de aceptación

- Se ha enviado correctamente la solicitud de amistad.
- Aparece en la lista de solicitudes de amistad del usuario al que se le ha enviado

Escenarios

Escenario: Enviar solicitud de amistad a un usuario existente

Dado un usuario registrado en el sistema **cuando** indica el nombre de un usuario existente al que quiere enviar la solicitud de amistad **entonces** se le envía la solicitud de amistad.

Escenario: Enviar solicitud de amistad a un usuario que no existe

Dado un usuario registrado en el sistema **cuando** indica el nombre de un usuario que no existe en la plataforma al que quiere enviar la solicitud de amistad **entonces** salta una excepción indicando al usuario que no existe el usuario indicado.

2. Unirse a partida

Actor principal	Jugador
Precondición	<ol style="list-style-type: none">1. El Jugador está registrado en el sistema.2. La partida a la que quiere unirse el usuario existe.
Disparador	El Jugador quiere unirse a una partida.
Escenario principal del éxito	<ol style="list-style-type: none">1. El usuario indica que quiere unirse a una partida.2. El sistema muestra dos listas de partidas en curso: partidas públicas y partidas privadas.3. El usuario elige a qué partida quiere unirse.4. El usuario elige con qué personaje va a unirse a una partida.5. El usuario se une a una partida elegida.
Extensiones	<ol style="list-style-type: none">2a. Todas las partidas están llenas.2b. No hay partidas en curso.

Feature: Unirse a la partida

Como usuario de la aplicación que está logueado, **quiero poder** unirme a una partida **para poder** jugar esa partida.

Criterios de aceptación

- El usuario se ha unido a la partida indicada

Escenarios

Escenario: Unirse a una partida existente

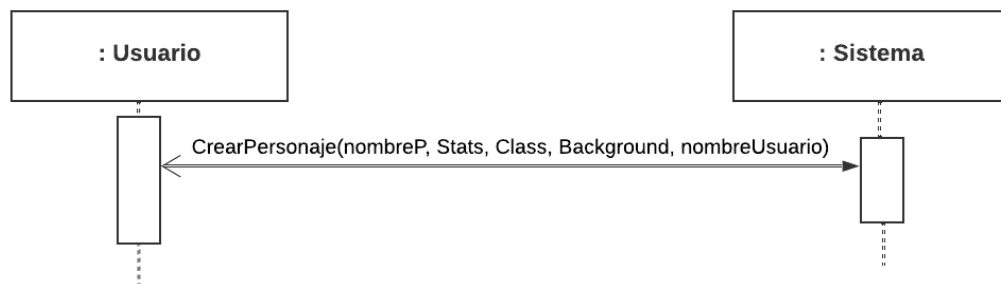
Dado un usuario registrado en el sistema **cuando** quiere unirse a una partida que no está llena y está disponible **entonces** se le agrega a la lista de jugadores de la partida

Escenario: Intentar unirse a una partida llena

Dado un usuario registrado en el sistema **cuando** quiere unirse a una partida que está llena **entonces** salta una excepción indicando al usuario que la partida que ha seleccionado está llena.

3. Crear personaje

Actor principal	Jugador
Precondición	1. El Jugador está registrado en el sistema.
Disparador	El jugador quiere crear un personaje.
Escenario principal del éxito	<ol style="list-style-type: none"> 1. El Jugador indica al Sistema que quiere crear un personaje. 2. El Sistema muestra una lista a la que el Jugador puede añadir un nuevo personaje. 3. El Jugador rellena la información necesaria para la creación de un nuevo personaje y la guarda. 4. El Sistema confirma que el proceso de creación de un nuevo personaje ha acabado con éxito.
Extensiones	<p>3a. El Jugador no ha rellenado todos los campos necesarios antes de guardar.</p> <p>3b. El Jugador ya tiene un personaje con el mismo nombre y las mismas características.</p>



Especificación de la operación

Contexto: Sistema::CrearPersonaje(nombreP: String, Stats: String, Class: TypeClass, Background: String, nombreUsuario: String)

Pre: Usuario.allInstances() -> exists (u | u.UserName = nombreUsuario) and Personaje.allInstances() -> not exists (p | p.Name = nombreP)

Post: Personaje.allInstances() -> exists (p | p.ocllsNew() and p.Name = nombreP and p.Stats = Stats and p.Class = Class and p.Background = Background and p.creador.UserName = nombreUsuario)

4. Acabar partida

Actor principal	Jugador
Precondición	<ol style="list-style-type: none">1. El Jugador está registrado en el sistema.2. La partida está en proceso.3. El Jugador participa en la partida.4. El jugador está jugando a la partida en ese momento
Disparador	El Jugador quiere acabar la partida a la que está jugando
Escenario principal del éxito	<ol style="list-style-type: none">1. La decisión de acabar la partida se pone en la votación entre todos los jugadores de una partida.2. En caso de la mayoría de votos a favor el Sistema marca la partida como acabada.
Extensiones	2a. La mayoría de votos está en contra de la solución de acabar una partida.

Feature: Acabar Partida

Como usuario de la aplicación que está logueado y en una partida en curso, **quiero poder** acabar una partida **para poder** dar fin a una partida en transcurso.

Criterios de aceptación

- Se ha finalizado correctamente la partida.
- Aparece en la lista de partidas jugadas del usuario.

Escenarios

Escenario: Acabar una partida con votación positiva

Dado un usuario registrado en el sistema **cuando** indica que quiere finalizar una partida existente y que está en transcurso **entonces** se inicia una votación entre todos los jugadores de la partida y si esta es positiva la partida finaliza.

Escenario: Acabar una partida con votación negativa

Dado un usuario registrado en el sistema **cuando** indica que quiere finalizar una partida existente y que está en transcurso **entonces** se inicia una votación entre todos los jugadores de la partida y si esta es negativa la partida no finaliza.

5. Escribir en el chat

Actor principal	Jugador
Precondición	1. El Jugador está registrado en el sistema.
Disparador	El Jugador quiere escribir en el chat.
Escenario principal del éxito	1. El Jugador abre el chat. 2. El Jugador escribe su mensaje en el chat y lo envía.
Extensiones	2a. El Jugador olvida de enviar un mensaje antes de cerrar un chat.

Feature: Escribir en el chat

Como usuario de la aplicación que está logueado y en una partida en curso , **quiero poder** escribir en el chat **para poder** comunicarme con el resto de jugadores.

Criterios de aceptación

- El mensaje aparecerá en el chat de la partida

Escenarios

Escenario: Escribir en el chat

Dado un usuario de la aplicación logueado que está en una partida **cuando** indica que mensaje quiere enviar y lo envía **entonces** se muestra el mensaje por el chat.

6. Tomar una decisión

Actor principal	Jugador
Precondición	1. El Jugador está registrado en el sistema.
Disparador	El Jugador quiere tomar una decisión.
Escenario principal del éxito	1. El Jugador comunica una acción de su personaje por escrito en el chat o en voz por el micrófono.
Extensiones	1a. El Jugador tiene un micrófono apagado.

Feature: Tomar una decisión

Como usuario de la aplicación que está logueado y en una partida en curso **quiero poder** tomar una decisión en la partida **para poder** avanzar la partida de la forma deseada

Criterios de aceptación

- La partida sigue transcurriendo según lo deseado

Escenarios

Escenario: Tomar una decisión

Dado usuario de la aplicación que está logueado y en una partida en curso **cuando** el *Digital Human* le plantea una situación en la que debe decidir **entonces** tomará la decisión y el *Digital Human* seguirá la partida a partir de la decisión tomada.

Escenario: El jugador no responde a la decisión indicada

Dado usuario de la aplicación que está logueado y en una partida en curso **cuando** el *Digital Human* le plantea una situación en la que debe decidir y el usuario no responde **entonces** el *Digital Human* tratara de volver a hacer la pregunta y si sigue sin responder continuará la partida.

7. Eliminar personaje

Actor principal	Jugador
Precondición	El Jugador está registrado en el sistema
Disparador	El Jugador quiere eliminar un personaje.
Escenario principal del éxito	<ol style="list-style-type: none">1. El Jugador indica al Sistema que quiere eliminar un personaje.2. El Sistema muestra una lista donde el Jugador puede seleccionar el personaje que quiere eliminar.3. El Jugador selecciona el personaje.4. El Sistema elimina el personaje.
Extensiones	<ol style="list-style-type: none">1. El Sistema informa al Jugador que la búsqueda no ha producido resultados.

Feature: Eliminar personaje

Como usuario de la aplicación que está logueado, **quiero poder** eliminar un personaje **para poder** eliminarlo de mi perfil.

Criterios de aceptación

- El personaje ha sido eliminado correctamente.

Escenarios

Escenario: Eliminar personaje

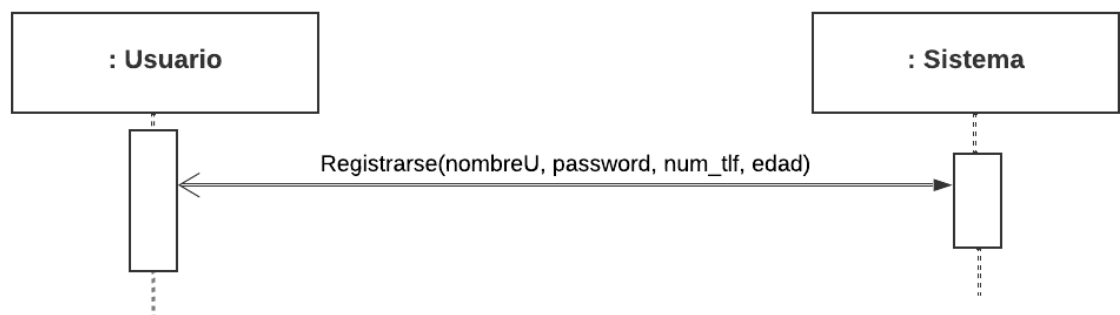
Dado un usuario registrado en el sistema **cuando** indica que quiere eliminar un personaje propio y existente **entonces** se elimina el personaje.

Escenario: Eliminar personaje no existente

Dado un usuario registrado en el sistema **cuando** indica que quiere eliminar un personaje propio y no existente **entonces** se muestra un mensaje de error indicando que el personaje no existe.

8. Registrarse

Actor principal	Jugador
Precondición	El Jugador no está inscrito en el Sistema.
Disparador	El Jugador quiere registrarse en el Sistema.
Escenario principal del éxito	<ol style="list-style-type: none">1. El Jugador indica al Sistema que quiere registrarse.2. El Sistema muestra una ventana donde el Jugador puede introducir su correo electrónico, una contraseña y un nombre de usuario.3. El Jugador introduce el correo electrónico, una contraseña y un nombre de usuario.4. El Sistema crea un nuevo usuario con los atributos especificados.
Extensiones	<ol style="list-style-type: none">1. Ya existe un usuario con el mismo correo electrónico<ol style="list-style-type: none">1a. El Sistema informa del error y da la opción de introducir otro correo2. Ya existe un usuario con el mismo nombre de usuario.<ol style="list-style-type: none">2a. El Sistema informa del error y da la opción de introducir otro correo



Especificación de la operación

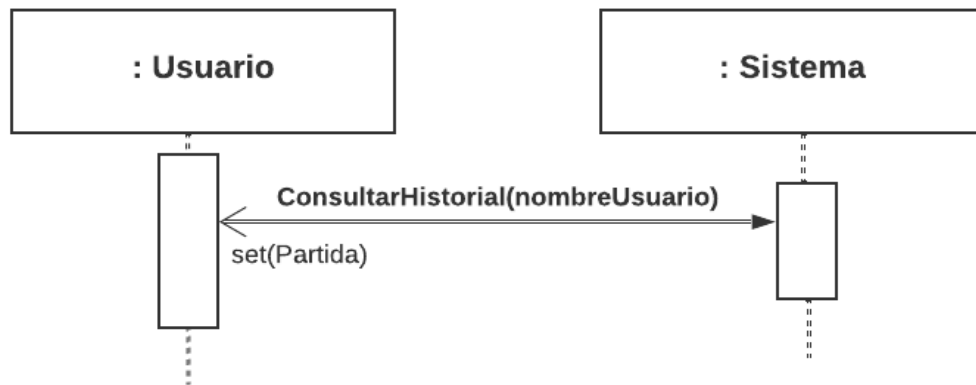
Contexto: Sistema::Registrarse(nombreU: String, password: String, num_tlf: Integer, edad: Integer)

Pre: Usuario.allInstances () -> not exists (u | u.UserName = nombreU)

Post: Usuario.allInstances () -> exists (u | u.ocllsNew() and u.UserName = nombreU and u.Password = password and u.TlfNumber = num_tlf and u.Edad = edad)

9. Consultar historial de partidas jugadas

Actor principal	Jugador
Precondición	El Jugador está registrado en el sistema
Disparador	El Jugador quiere consultar su historial de partidas jugadas.
Escenario principal del éxito	<ol style="list-style-type: none">1. El Jugador indica al Sistema que quiere ver su historial de partidas jugadas.2. El Sistema muestra una lista con todas las partidas jugadas por el Jugador.
Extensiones	<ol style="list-style-type: none">1. El Jugador no tiene ninguna partida jugada. 1a. El Sistema informa del error



Especificación de la operación

Contexto: Sistema::ConsultarHistorial(nombreUsuario: String) : Set(Partida)

Pre: Usuario.allInstances () -> exists (u | u.UserName = nombreUsuario)

Body: let partidas: Set(Partidas) = Partidas.allInstances () -> select (p | p.personaje -> select (pj | pj.usuario -> select (u | u.UserName = nombreUsuario))) in result = partidas

10. Lanzar dado

Actor principal	Jugador
Precondición	El Jugador está registrado en el sistema
Disparador	El Jugador quiere lanzar un dado.
Escenario principal del éxito	<ol style="list-style-type: none">1. El Jugador indica al Sistema que quiere lanzar un dado.2. El Sistema muestra una lista con los diferentes dados que se pueden utilizar en ese momento.3. El Jugador selecciona el dado que quiere utilizar.4. El Sistema lanza el dado y muestra por pantalla el número obtenido.

Feature: Lanzar dado

Como jugador, **quiero poder** lanzar un dado cuando la partida lo requiera **para poder** resolver una situación que requiere el uso de dados

Criterios de aceptación

- Se ha podido lanzar el dado.
- El resultado del dado se ha registrado en la partida y esta sigue su curso a partir del resultado obtenido.

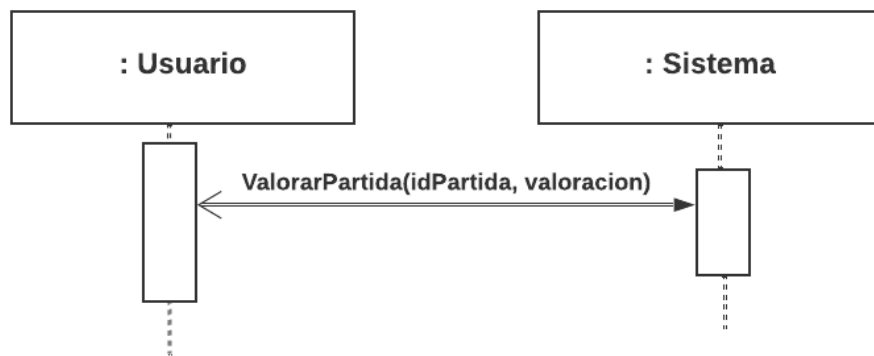
Escenarios

Escenario: Lanzar dado

Dado jugador **cuando** se requiera una tirada de dados **entonces** se iniciara una tirada de dados, el jugador lo lanzara y la partida continuará a partir del resultado obtenido.

11. Valorar partida

Actor principal	Jugador
Precondición	<p>El Jugador está registrado en el sistema</p> <p>La Partida ha finalizado</p> <p>El Jugador ha participado en la partida a valorar</p>
Disparador	El Jugador quiere valorar la partida.
Escenario principal del éxito ni	<ol style="list-style-type: none">1. El Jugador indica al Sistema que quiere valorar una partida con un entero entre 0 y 5.2. El Sistema registra la valoración.



Especificación de la operación

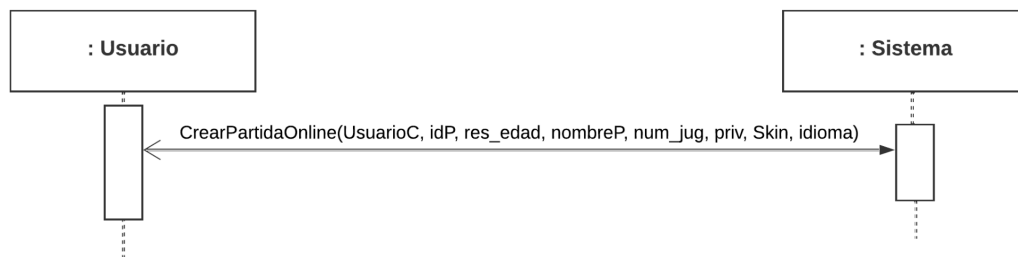
Contexto: Sistema::ValorarPartida(NombreP:String, idPartida: Integer, valoracion: Integer)

Pre: Personaje.allInstances() -> exists (p | p.Nombre = NombreP and p.partida.Id = idPartida and p.partida.Terminada = true)

Post: Partida.allInstances () -> exists (pt | pt.id = idPartida and pt.Valoración = valoración)

12. Crear partida online

Actor principal	Jugador
Precondición	El Jugador está registrado en el sistema
Disparador	El Jugador quiere crear una partida online
Escenario principal del éxito	<ol style="list-style-type: none">1. El Jugador indica al Sistema que quiere crear una partida online.2. El Sistema crea una partida online con el Jugador como host.
Extensiones	<ol style="list-style-type: none">1. El Jugador ya tiene una partida activa.<ol style="list-style-type: none">1a. El sistema informa del error.



Especificación de la operación

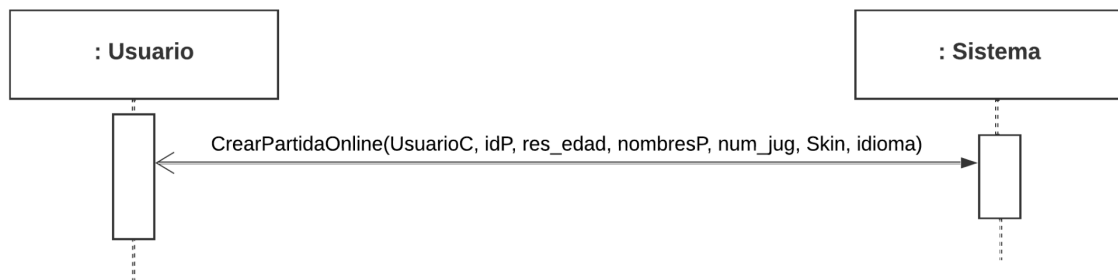
Contexto: Sistema::CrearPartidaOnline(UsuarioC: String, idP: Integer, res_edad: Integer, nombreP: String, num_jug: Integer, priv: TypePrivacidad, Skin: String, idioma: String)

Pre: Usuario.allInstances () -> exists (u | u.UserName = UsuarioC) and Personaje.allInstances () -> exists (p | p.Name = nombreP and p.creador.UserName = UsuarioC) and Partida.allInstances () -> not exists (ptd | ptd.id = idP)

Post: Partida.allInstances () -> exists (ptd | pt.ocllsNew() and ptd.id = idP and ptd.Restricción_de_edad = res_edad and ptd.Terminada = false and ptd.Valoración = null and ptd.ocllsType(Online) and ptd.oclAsType(Online).NumeroJugadores = num_jug and ptd.oclAsType(Online).Privacidad = priv and ptd.oclAsType(Online).usuario.UserName = UsuarioC and ptd.personaje.Name = nombreP and ptd.Dungeon_Master.Skin = Skin and ptd.Dungeon_Master.Idioma = idioma)

13. Crear partida local

Actor principal	Jugador
Precondición	El Jugador está registrado en el sistema
Disparador	El jugador quiere crear una partida local.
Escenario principal del éxito	<ol style="list-style-type: none"> 1. El Jugador indica al Sistema que quiere crear una partida local. 2. El Sistema crea una partida local.
Extensiones	<ol style="list-style-type: none"> 1. El Jugador ya tiene una partida activa. 1a. El sistema informa del error.



Especificación de la operación

Contexto: Sistema::CrearPartidaLocal(UsuarioC: String, idP: Integer, res_edad: Integer, nombresP: set(String), num_jug: Integer, Skin: String, idioma: String)

Pre: Usuario.allInstances () -> exists (u | u.UserName = UsuarioC and u.personaje.Name -> includesAll(nombresP)) and Partida.allInstances () -> not exists (ptd | ptd.id = idP)

Post: Partida.allInstances () -> exists (ptd | pt.ocllsNew() and ptd.id = idP and ptd.Restricción_de_edad = res_edad and ptd.Terminada = false and ptd.Valoración = null and ptd.ocllsType(Local) and ptd.oclAsType(Local).NumeroJugadores = num_jug and ptd.oclAsType(Local).usuario.UserName = UsuarioC and ptd.personaje.Name -> includesAll(nombresP) and ptd.Dungeon_Master.Skin = Skin and ptd.Dungeon_Master.Idioma = idioma

14. Rejugar partida

Actor principal	Jugador
Precondición	El Jugador está inscrito en el Sistema. La partida ha finalizado y está en el historial de partidas jugadas.
Disparador	El Jugador quiere volver a jugar una partida.
Escenario principal del éxito	<ol style="list-style-type: none">1. El Jugador indica al Sistema que quiere volver a jugar una partida ya finalizada.2. El Sistema crea una partida online y envía la invitación a los jugadores participantes de la partida que se quiere volver a jugar.
	<ol style="list-style-type: none">1. El Sistema informa del error.

Feature: Rejugar partida

Como usuario de la aplicación que está logueado, **quiero poder** rejugar una partida **para poder** volver a jugar la misma partida de manera diferente.

Criterios de aceptación

- El jugador puede jugar la partida.
- La partida aparece en partidas disponibles.

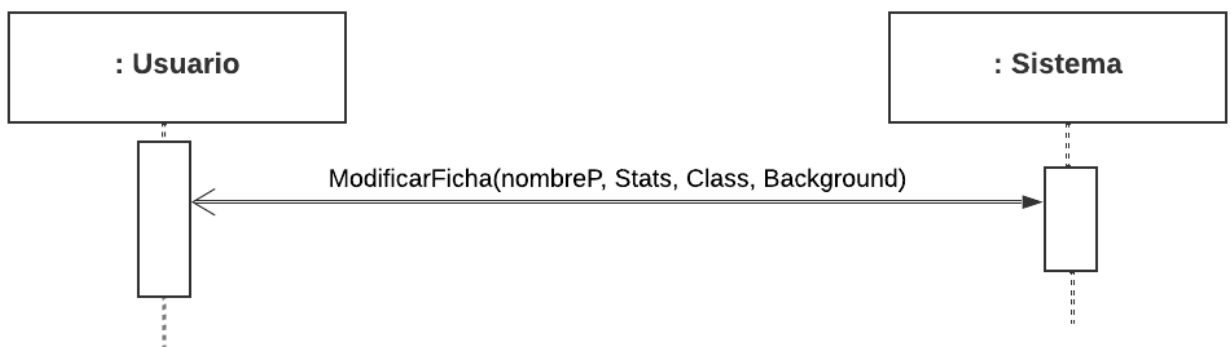
Escenarios

Escenario: Rejugar partida

Dado un usuario registrado en el sistema **cuando** indica que quiere rejugar una partida finalizada de su historial **entonces** se añade a partidas disponibles, se envían las invitaciones a los jugadores participantes y se puede jugar.

15. Modificar ficha de un personaje

Actor principal	Jugador
Precondición	<ol style="list-style-type: none">1. El Jugador está registrado en el sistema.2. El Jugador tiene al menos un personaje creado.
Disparador	El jugador quiere modificar la ficha de un personaje.
Escenario principal del éxito	<ol style="list-style-type: none">1. El Jugador accede a una lista de personajes disponibles.2. Elige un personaje.3. Introduce las modificaciones.
	1a. El Jugador no tiene ningún personaje creado.



Especificación de la operación:

Contexto: Sistema::ModificarFicha(nombreP: String, Stats: String, Class: TypeClass, Background: String)

Pre: Personaje.allInstances () -> exists (p | p.Name = nombreP)

Post: Personaje.allInstances () -> exists (p.Name = nombreP and p.Stats = Stats and p.Class = Class and p.Background = Background)

7. Argumentación del logro de los objetivos

Debido a que nuestro sistema permitirá jugar a *Dungeons and Dragons* utilizando un *Digital Human* como Dungeon Master, los jugadores de rol y la gente que quiere empezar a jugar verán cumplidos sus objetivos. Por una parte podrán jugar sin que nadie interprete el papel de DM, ya que de esto se encargará el *Digital Human*. Por otra parte mediante la generación de imágenes y mapas o la skin del *Digital Human* los jugadores tendrán una gran inmersión en la partida. Gracias a la extensa base de datos, se podrán ofrecer diversas ambientaciones a los jugadores. También con el modo *online* se ofrecerá la posibilidad de realizar partidas a distancia además de una forma de conocer a otros jugadores. Finalmente mediante el modo tutorial los jugadores principiantes podrán aprender las mecánicas del juego mientras disfrutan de una partida adaptada a su nivel .

Gracias al gran público que tiene la comunidad de *Dungeons and Dragons*, aproximadamente 40 millones de jugadores en todo el mundo, y nuestra versión novedosa del juego con una tecnología innovadora e inmersiva, se espera atraer a distintos tipos de público de distintos rangos de edades, tanto nuevos jugadores como jugadores veteranos. Este incremento del número de jugadores repercutirá directamente en las ventas tanto del juego físico como de nuestra aplicación, promocionando la marca de Wizards of the coast y dando a conocer el juego y otros juegos de la marca a nuevos públicos viendo cumplidos sus objetivos.

Los vendedores de tiendas especializadas por su parte, verán aumentadas las ventas gracias a que nuestra plataforma dará más visibilidad y reconocimiento al juego ya que como se ha mencionado anteriormente.

Finalmente los expertos en *Digital Humans* podrán ver un uso novedoso de la tecnología y un desarrollo de la misma. Como la parte central de nuestro sistema es el *Digital Human*, la tecnología se dará a conocer en el caso de que la plataforma tenga éxito por lo que los expertos en esta tecnología verán cumplidos sus objetivos.

En conclusión, los stakeholders de nuestro proyecto verán cumplidos sus objetivos con el desarrollo de la plataforma.

8. Glossario

NPCs (Non Playable Characters): Término utilizado en videojuegos que hace referencia a personajes que no son controlados por una persona real.

Dungeons and Dragons: Es un juego de rol en el que el objetivo principal de los jugadores es contar una historia.

Dungeon Master (DM): Es el encargado de narrar la historia en Dungeons and Dragons así mismo de supervisar y dirigir el curso narrativo de una partida. Entre sus funciones también se encuentra definir las consecuencias de los personajes e interpretar a los NPC.

Inteligencia Artificial (IA): Se refiere a sistemas o máquinas que imitan la inteligencia humana para realizar tareas y que pueden mejorar iterativamente a partir de la información que recopilan.

DALL·E: Programa de inteligencia artificial que crea imágenes a partir de descripciones textuales.

Roll20: Aplicación web para jugar juegos de mesa en persona o de forma online.

D&D: Abreviatura de las siglas de *Dungeons and Dragons*.

Wizards of the coast: Creadores y distribuidores de Dungeons and Dragons.

Estadísticas: En *Dungeons and Dragons*, en el momento de la creación de personajes, a cada jugador se le asignan unas estadísticas a partir del resultado de una serie de tiradas con dados. Cada personaje tiene las siguientes: Fuerza, Destreza, Constitución, Inteligencia, Sabiduría,...

Clase: Igual que la raza, la clase de un personaje determinará de lo que este es capaz y le dará ciertos bonos en algunas estadísticas, algunas clases son: mago, bardo, arquero, guerrero, etc.

Stats: Significa estadísticas en inglés.

Skin: Diferentes apariencias que puede adoptar un personaje.

Background: Significa trasfondo en inglés. En *Dungeons and Dragons*, en el momento

Fichas de personajes: Cada personaje de la partida tiene una ficha propia donde se indican sus estadísticas, background, clase y otros atributos importantes para la partida como objetos que lleva el personaje etc. Sirve para saber qué acciones puede realizar tu personaje y para guardar información de cómo evoluciona en la partida, cambios en la vida, las estadísticas o nuevos objetos.