

Proyecto Fin de Ciclo Grado Superior  
Desarrollo de Aplicaciones Multiplataforma  
Miteris - Curso 2022-2023

# PROYECTO DE FIN DE CICLO

## JUEGO ONLINE: CAVES & ORCS

**Autor**

Ricardo Sánchez Cortés

**Tutor**

David Ruiz Castrillo

 ***miteris***



# JUEGO ONLINE: CAVES & ORCS



AUTOR  
RICARDO SÁNCHEZ CORTÉS



### Resumen

El presente proyecto consiste en un juego web narrado tipo RPG (role playing game). La idea está basada en el sistema de juego de muchos juegos de rol como puede ser la famosa saga de juegos y novelas “Dungeons & Dragons”. Las reglas del juego son inventadas, así como el sistema de combate y la resolución de eventos.

La aplicación en sí permite el registro como usuario para la creación de personajes propios, los cuales son utilizados para el juego en sí.

Esta versión es una demo del juego, pero más adelante puede ser extendida al juego completo. Actualmente se puede elegir entre 80 combinaciones distintas de personaje. Además, cada personaje creado será único puesto que el jugador podrá elegir donde reparte sus puntos de atributos. Estos puntos de atributos se conseguirán al crear el personaje y cada vez que se complete con éxito un evento, pudiendo así ir mejorando los personajes a medida que se juega. Además, cada tipo de personaje tiene una manera diferente de interactuar en los eventos por lo que los jugadores pueden probar distintos personajes para superar los retos de maneras diferentes.

Los combates y distintos eventos se resolverán con “tiradas de dados” que consistirán en la obtención de un número aleatorio dentro de determinados parámetros, el cual definirá si el personaje tiene éxito o se fracasa. Los atributos de los personajes tendrán gran peso a la hora de determinar el resultado.

La versión completa contará con un elevado número de eventos que tendrán una dificultad incremental de manera que se mantenga el reto a medida que se vaya mejorando a los personajes. Habrá un evento final más difícil que los anteriores que necesite haber mejorado lo suficiente al personaje para poder ser superado.

## Abstract

This project consists of a narrated RPG (role playing game) type web game. The idea is based on the game system of many role-playing games such as the famous saga of games and novels "Dungeons & Dragons". The rules of the game are invented, as well as the combat system and the resolution of events.

The application itself allows registration as a user to create your own characters, which are used for the game itself.

This version is a demo of the game but can later be extended to the full game. Currently you can choose between 80 different character combinations. In addition, each character created will be unique since the player will be able to choose where he distributes his attribute points. These attribute points will be obtained when creating the character and each time an event is successfully completed, thus being able to improve the characters as the game is played. Also, each type of character has a different way of interacting in events so players can try different characters to overcome challenges in different ways.

The combats and different events will be resolved with "rolls of the dice" that will consist of obtaining a random number within certain parameters, which will define whether the character succeeds or fails. The attributes of the characters will have great weight when determining the result.

The full version will have a large number of events that will have increasing difficulty so that the challenge is maintained as the characters are improved. There will be a final event more difficult than the previous ones that needs to have improved the character enough to be able to be overcome.

# TABLA DE CONTENIDO

<b>1. Introducción .....</b>	<b>1</b>
1.1. Motivación por el tema elegido.....	1
1.2. Objetivos .....	2
1.3. Metodología utilizada .....	2
1.4. Resumen del contenido de la memoria.....	2
<b>2. Justificación de la relevancia de la temática .....</b>	<b>3</b>
<b>3. Fundamentación teórica, antecedentes y estado del arte .....</b>	<b>4</b>
<b>4. Análisis del estudio realizado .....</b>	<b>6</b>
4.1. Análisis y estudio del tema elegido .....	6
4.2. Especificación de requisitos.....	10
4.2.1. Introducción.....	10
4.2.2. Descripción General.....	11
4.2.3. Requisitos Específicos .....	12
4.2.4. Apéndices.....	13
<b>5. Diseño de la aplicación .....</b>	<b>14</b>
<b>6. Conclusiones e implicaciones.....</b>	<b>17</b>
<b>7. Propuestas de mejora y líneas de futuro .....</b>	<b>18</b>
7.1. Jugabilidad .....	18
7.2. Plataforma .....	18
7.3. Diseño .....	19
<b>8. Bibliografía y/o referencias.....</b>	<b>20</b>
<b>9. Anexos .....</b>	<b>21</b>
Anexo 1: Tablas de objetivos, actores y casos de uso .....	21
Anexo 2: Diagramas de casos de uso.....	30
Anexo 3: Manual de usuario .....	32

# ÍNDICE DE ILUSTRACIONES

Ilustración 1. Escenario Irithyll (Dark Souls 3) .....	1
Ilustración 2. Consola de hogar Atari Pong .....	4
Ilustración 3. Consolas de distintas épocas .....	5
Ilustración 4. Logo Dungeons & Dragons.....	6
Ilustración 5. Muestra imagen personaje .....	8
Ilustración 6. Muestra imagen enemigo .....	9
Ilustración 7. Logo MySQL.....	14
Ilustración 8. Logo Hibernate.....	14
Ilustración 9. Paquetes de clases .....	14
Ilustración 10. Carpetas del contenido .....	14
Ilustración 11. Diagrama E-R Base de datos .....	15
Ilustración 12. Clase Personaje .....	16
Ilustración 13. Manual - Elegir acciones .....	32
Ilustración 14. Manual - Ejemplo batalla .....	33
Ilustración 15. Manual - Registro .....	34
Ilustración 16. Manual - Login.....	34
Ilustración 17. Manual - Crear personaje .....	35
Ilustración 18. Manual - Repartir atributos .....	35
Ilustración 19. Manual - Vista personaje .....	35
Ilustración 20. Manual - Ranking .....	36



## 1. Introducción

### 1.1. Motivación por el tema elegido

Este tema ha sido elegido por el interés del autor en el mundo de los videojuegos. Es un área con muchas posibilidades que permite liberar la imaginación de los artistas dando lugar a obras maestras como muchas de las que se encuentran actualmente en el mercado.

Este trabajo pretende ser un primer paso para el camino de la creación de videojuegos en el cual pretende embarcarse el autor del mismo. Es un trabajo muy humilde y sencillo, pero permite sentar bases y utilizar distintos estilos de creación de juegos. Por ejemplo, en la versión completa del juego el sistema de sucesión de eventos se basará en el sistema “Rogue Like”, el cual presenta eventos autogenerados. Estos eventos se generan mezclando unos parámetros que ya existen, lo cual permite que la combinación de estos parámetros genere un gran número de eventos con mucho menos trabajo para el creador. Por ejemplo, se pueden diseñar 5 escenarios, 5 tipos de enemigos y 5 tipos de misiones que hay que lograr en cada evento. Eligiendo al azar un escenario, un tipo de enemigo y un tipo de misión nos encontramos ante la posibilidad de 125 ( $5 \times 5 \times 5 = 125$ ) eventos diferentes.



*Ilustración 1. Escenario Irithyll (Dark Souls 3)*

Si además introduces un sistema en el cual puedes elegir aumentar la dificultad de estos eventos otorgando cada vez mejores recompensas que mejoren a los personajes con los que se juega, se consigue que el jugador quiera repetir las partidas con tal de ir superando retos cada vez más difíciles e ir mejorando más y más al personaje.

### 1.2. Objetivos

Se pretende realizar un juego visual que, aunque sea sencillo, pueda entretener a sus jugadores. La posibilidad de personalizar al personaje pretende que el jugador se sienta más próximo al mismo y así se inmerse más en el juego. Se han añadido imágenes para crear una mayor inmersión.

### 1.3. Metodología utilizada

Se ha programado con los lenguajes Java y Javascript. Para la interfaz se ha utilizado HTML, CSS y Bootstrap. Para la gestión de las librerías se ha utilizado Maven. En cuanto al almacenamiento de datos se ha optado por un base de datos MySQL con Hibernate como intermediario para su gestión. Se ha utilizado un servidor de Tomcat para la elaboración y pruebas en modo local.

Todo se ha realizado con la metodología MVC (modelo-vista-controlador), tratando de escribir un código limpio y ordenado para facilitar las futuras mejoras o resolución de errores.

### 1.4. Resumen del contenido de la memoria

Esta memoria se compone de todo lo necesario para entender la aplicación tanto en su proceso de elaboración como en su funcionamiento en sí.

Se comenzará explicando el porqué de la aplicación, cuales han sido las causas de su creación y de donde ha salido la idea que le ha dado forma. Para ello, se retrocederá en el tiempo para explicar el origen de aplicaciones similares, su desarrollo a lo largo de las décadas y su presencia en la actualidad.

Después de eso, viene el cómo se fue dando forma a la idea y la evolución de la misma a lo largo del recorrido de creación hasta el resultado final actual. En este apartado se irán dando pinceladas a las distintas partes de las que se compone la aplicación y se irá definiendo en cuerpo y forma.

A continuación, se ha simulado una especificación de requisitos como la que se generaría si fuera un proyecto a petición de un cliente. De esta manera se pueden ver los casos de uso, actores y demás elementos desde el punto de vista técnico y así tener las trazas generales sobre las que se sostiene la aplicación.

Lo anterior sirve para profundizar en la explicación que se da en el siguiente apartado del propio diseño de la aplicación y las herramientas utilizadas para su elaboración.

En los últimos apartados se mencionan las conclusiones a las que se ha llegado con la terminación del proyecto, lo que se ha aprendido de él y las posibles mejoras y expansiones que puede tener un futuro.

Al final del presente documento se encontrarán todas las tablas y diagramas y el manual de usuario para la comprensión y uso del programa por parte de los usuarios. Este manual de usuario está escrito en un tono más coloquial y cercano puesto que quién lo va a leer son los jugadores de la aplicación, que serán de todas las edades por lo que se ha intentado hacer lo más amigable posible.

## 2. Justificación de la relevancia de la temática

El ocio tiene mucha demanda hoy en día puesto que vivimos una vida atropellada y llena de estrés por llegar a tiempo a la siguiente reunión o conseguir hacer a tiempo el trabajo que nos han pedido. La gente busca una vía de escape que les permita olvidarse, al menos durante un rato, de todos los problemas e inquietudes que les acechan. Hay muchas maneras de conseguir ese escape y una de ellas es el ocio digital. Actualmente utilizamos muchos dispositivos electrónicos para distraernos y entretenernos. Los ordenadores sin ir más lejos son utilizados para infinidad de tareas y la mayoría de la gente cuenta ya con un ordenador personal, por ello son muchas las personas que lo utilizan para el ocio. Si a ello le añadimos la infinidad de posibilidades que nos brinda internet obtenemos a gran parte de la población que pasa muchas horas frente a las pantallas.

Por todo ello los videojuegos son muy populares ya que permiten al usuario ver inmerso en un mundo ficticio en el que abstraerse. El mercado de los videojuegos es muy amplio y está siempre en expansión. Continuamente están saliendo nuevos modelos de videojuegos con los que sorprender a los usuarios. Esto hace que, aunque los videojuegos sean para ocio, sean algo con notable impacto en la sociedad actual ya que son muy demandados por la gente.

Un videojuego aúna la gran complejidad de lógica que lleva en su código que tienen que implementar los desarrolladores junto con la búsqueda de una estética lo más atractiva posible, lo que exige mucho trabajo por parte de los diseñadores. Precisan de un front end y un back end muy trabajados. Por otro lado, en las ocasiones en las que el juego te cuenta una historia, se necesita un argumento que tenga gancho y haga al usuario querer jugar toda la historia hasta el final.

Todos estos factores son algo muy atractivo para el autor de este proyecto ya que todo ello representa un reto y sobre todo mucha imaginación. Por ello se ha elegido este tema que precisa de originalidad y brinda la satisfacción de haber superado el reto una vez terminado. De hecho, estos motivos son suficientes para que el autor de este proyecto esté interesado en trabajar como desarrollador de videojuegos en un futuro.

### 3. Fundamentación teórica, antecedentes y estado del arte

Los videojuegos existen desde la década de los 50 cuando en los primeros ordenadores se empezaron a implementar programas. Estos videojuegos eran del estilo de tres en raya y juegos parecidos. Al principio se necesitaban grandes ordenadores para poder utilizar estos juegos.

En la década de los 60 ya se empezaron a plantear utilizar las pantallas de los televisores para conectarlas a las máquinas que procesaban los videojuegos y de este modo poder visualizarlos en estas pantallas. Se crearon un par de juegos que funcionaban de esta manera.

Pero los videojuegos aún no estaban dispuestos para el público, fue en la década de los 70 cuando las máquinas recreativas se empezaron a instalar en salones de ocio. Algunos de los primeros juegos que se crearon con estas máquinas fueron Computer Space y Galaxy War, pero el juego que marcó la diferencia fue el Pong un videojuego creado por la empresa Atari. Posteriormente en esa década aparecieron juegos como el Space Invaders o el Asteroids. Estas máquinas recreativas tuvieron gran éxito entre el público lo que supuso que se instalaran en muchos salones recreativos expandiéndose rápidamente este sistema de negocio por las ciudades de muchos países del mundo.

Simultáneamente a las máquinas recreativas, en esta misma década también se empezaron a comercializar las videoconsolas con un sistema doméstico que permitía poder jugar en los hogares. Atari se destacó por ir sacando muchos nuevos juegos durante esta época.



*Ilustración 2. Consola de hogar Atari Pong*

En los 80 se expandieron los videojuegos de 8 bits, los gráficos ya iban mejorando y los juegos empezaban a ser más visuales y entretenidos. Un gran ejemplo de esta década es el Pacman. En esta época salieron a la venta varias videoconsolas como la NES de Nintendo o la Master System de Sega. A finales de los 80 aparecieron consolas que ya funcionaban con 16 bits permitiendo mejorar los gráficos de los juegos. Super Mario Bros fue uno de los grandes éxitos que salieron durante estos años. Este juego ya tenía una historia en la que avanzabas según ibas pasándote sus pantallas.

A partir de los 90 comenzaron a salir muchas videoconsolas y juegos, a estas alturas ya estaba claro que era un negocio muy rentable con un gran éxito entre el público. Con la aparición de la SNES (Super Nintendo) los fans de los videojuegos descubrieron una nueva consola con unos gráficos mucho mejores a todo lo que se había visto anteriormente y unos juegos que les hacían estar pegados a la pantalla durante horas. Algunos de estos juegos como el Killer Instinct o el Donkey Kong Country ya utilizaban una tecnología de gráficos 3D renderizados previamente.

Según avanzaba esta década fueron saliendo nuevos juegos y nuevas plataformas para jugarlos. Los ordenadores empezaban a desarrollarse lo suficiente como para poder utilizarlos para los gráficos y renderizados de los nuevos juegos que estaban saliendo. Con la llegada de internet se empezaron a desarrollar juegos multijugador que permitían jugar a la misma partida en varios ordenadores a la vez de modo que podías jugar con muchos jugadores al mismo tiempo. Esto hizo que surgieran las LAN

parties, eventos en los que la gente se juntaba con ordenadores para jugar juntos a juegos. Con ellas también surgió un nuevo sistema de negocio, que consistía en montar un establecimiento con muchos ordenadores donde los clientes pagaban por horas para jugar entre ellos. Estos establecimientos eran llamados los Cybers por los jóvenes del momento.

A partir de los años 2000 las consolas y los juegos han seguido mejorando de manera sorprendente. Con la mejora de la velocidad de internet se expandieron enormemente los juegos multijugador, permitiendo a los jugadores jugar desde sus casas con gente de todo el mundo. Esto también hizo que los Cybers perdieran casi toda la clientela y fuesen cerrando poco a poco puesto que ya no era necesario moverse de casa ni pagar para jugar con entre amigos.



*Ilustración 3. Consolas de distintas épocas*

En las siguientes décadas ha habido siempre una guerra de mercado entre las distintas plataformas en las que se podía jugar a los videojuegos. Entre los usuarios estaban los que preferían una consola u otra y los que preferían directamente el ordenador. Han surgido muchas empresas dedicadas a hacer videojuegos y algunos de los juegos sólo salen a la venta para una plataforma concreta buscando que el público se incline a comprar esa consola concreta, puesto que si los jugadores ven que empiezan a salir a la venta muchos juegos interesantes exclusivos de esa consola pueden sentirse inclinados a adquirirla. Los fans se han quejado mucho en este sentido puesto que no quieren tener que comprarse varias consolas para jugar a los últimos éxitos que salen al mercado, y por ello en los últimos años muchos de los estrenos salen ya a la venta para varias plataformas. Entre las distintas opiniones de que plataforma es mejor, actualmente, hay dos bandos claramente diferenciados: los que prefieren las videoconsolas y los que prefieren los ordenadores.

En estas últimas décadas han surgido los juegos de venta online y sin soporte físico. Esto significa que ya no se requiere un disco para insertar en el aparato y poder jugar, sino que el usuario puede comprar a través de una plataforma virtual el acceso a determinado juego y jugar cuando quiera accediendo con su usuario a la plataforma. Esto facilita mucho la tarea de venta para los creadores de videojuegos, sobre todo a aquellas pequeñas empresas que están empezando a desarrollar juegos puesto que publicando sus juegos en estas plataformas consiguen que puedan verlos y adquirirlos personas de todo el mundo.

En la actualidad es relativamente sencillo publicar un juego en una plataforma que llegue a mucha gente, pero como contrapartida a esto la competencia es inmensa, por lo que es posible que ese juego quede enterrado en el torbellino de juegos que salen al público constantemente. Eso sí, cuando un estreno se hace viral tiene el éxito asegurado gracias a la facilidad de comunicación que brinda internet.

Como se aprecia por todo lo mencionado, la producción de juegos es algo muy demandado y se ha decidido enfocar este proyecto en ello, ya que es un campo muy interesante, con muchos retos y con infinidad de posibilidades.

## 4. Análisis del estudio realizado

### 4.1. Análisis y estudio del tema elegido

Lo primero que se decidió es que sería una aplicación web, donde los usuarios la pudieran utilizar con sólo acceder a la URL de la misma.

En un inicio la idea era hacer una aplicación que permitiese crear personajes para luego utilizarlos en algún juego. En muchos juegos, sobre todo los RPG o los MMORPG (Massive Multiplayer Online Role Playing Game) la creación del personaje es un paso previo al juego en sí. En la actualidad, existen creadores de personaje que permiten un alto grado de personalización del personaje, pudiendo incluso definir el aspecto físico modificando las facciones, las formas del cuerpo, marcas físicas como lunares y cicatrices y mucho más. En el presente proyecto se pretendía hacer algo sencillo, pero que permitiera crear un personaje con características elegidas por el usuario y que existiesen suficientes combinaciones como para que no fuese fácil tener dos personajes iguales. Debido al gran conocimiento previo del autor en juegos RPG ambientados en mundos fantásticos medievales donde existe la magia y los monstruos se decidió que la creación de los personajes iría orientada en esa dirección. Hay



*Ilustración 4. Logo Dungeons & Dragons*

multitud de juegos y novelas en este ámbito, pero sobre todo el presente proyecto se ha inspirado en "Dungeons & Dragons". Lo primero que se buscaba era no hacer una copia del juego en el que estaba inspirado y por ello se cambiaron las clases de personaje creando unas similares a las que aparecen en el citado juego.

Se empezó por hacer la base de datos para poder ir teniendo una idea de cómo dar forma a la aplicación. Se optó por una base de datos MySQL para este proyecto.

Se planteó programar la aplicación en Phyton, pero debido a que ello exigía aprender desde cero ese lenguaje se decidió que sería más prudente trabajar con un lenguaje ya conocido. En un futuro se programarán aplicaciones en Phyton para probar ese lenguaje. Por ello finalmente se decidió utilizar Java como lenguaje para la aplicación, esto permitiría crear una aplicación sólida y más completa que si se hubiera tratado de programar en un lenguaje no utilizado hasta el momento.

Una vez se tenía planteada la base de datos y se había decidido el lenguaje empezó la parte del código en sí. Se escogió como gestor de la base de datos, el sistema de Hibernate por sus ventajas al utilizar los objetos de la programación orientada a objetos con una base de datos relacional.

En este surgió una duda, ¿cómo gestionar la tabla de los personajes? El motivo de esta duda fue porque si pensamos en un juego que se juega en todo el mundo, la tabla de los personajes puede tener miles de filas y al hacer la búsqueda de personajes por el id de usuario, el programa tendría que recorrer toda esta tabla tan grande. Se plantearon tres opciones:

- Que hubiera una sola tabla con todos los personajes de todo el mundo y buscar en esa enorme tabla de personajes por el id del usuario a todos los personajes de ese usuario.



- Que se crease una nueva tabla de personajes cada vez que se crease un nuevo usuario y así no habría que buscar a los personajes sino simplemente acudir a esa tabla recoger todos los personajes en ella.
- Almacenar a los personajes en un JSON que se leyera según el id del usuario.

Se descartaron las dos últimas opciones puesto que al final iban a crear mayor carga de trabajo a la aplicación y no parecían más eficientes que la primera, así que se decidió que realmente no era tan problemático hacer la búsqueda en una tabla única de personajes.

Se comenzó programando la parte del login de los usuarios, se buscaba hacer un login sencillo para el usuario pero que fuese sólido en cuanto a posibles errores. Tiene validaciones para comprobar que el usuario introduce los datos que se deben introducir en cada campo y que no se dejan campos vacíos.

También tiene un sistema para codificar las contraseñas de los usuarios de modo que estas nunca se guardan en la base de datos sin codificar, ni son manejadas sin codificar.

Por otro lado, estaba la incógnita de qué poderes darles a los administradores, se optó por tener a un administrador principal que puede editar y borrar a cualquier otro usuario y el resto de los administradores que pueden editar y borrar a todos aquellos usuarios que no sean administradores. Por lo tanto, existen tres tipos de usuario, el administrador principal, el resto de los administradores y los usuarios normales.

Una vez terminada la parte del login, se planteaba como presentar las pantallas de creación de los personajes. También se debía definir qué características iban a tener esos personajes. En este punto fue cuando surgió la idea de no hacer solamente un sistema de creación de personajes sino acompañarlo de un juego como tal. Se empezó a pensar cómo debía ser ese juego y las reglas que debía tener. De este modo, se decidió que los personajes tendrían unas características físicas que serían: el género, la raza y la clase (profesión). Por otro lado, tendrían unos atributos que definirían sus habilidades a la hora de interactuar con su entorno, estas habilidades serían: fuerza, agilidad, vigor, magia y labia. El juego consistiría en una serie de eventos o pruebas que deberían superar los personajes utilizando sus atributos. La raza y la clase afectarían a estos atributos, pero también lo harían el superar o no cada uno de los eventos. Por cada evento superado se podrían aumentar los atributos y por cada evento fallido se disminuirían. De este modo, el jugador deberá tener cuidado para no perder todos sus puntos de atributos y perder al personaje. Estos eventos se transmitirían al jugador de manera narrada, puesto que un sistema con animaciones requiere de tecnologías que se salen del alcance del presente proyecto. Por último, parecía necesario poner un objetivo que motivase a los jugadores a mejorar a sus personajes y seguir jugando los eventos y se ideó poner un evento final al cual los jugadores pudieran acceder cuando quisieran, pero para poder superarlo necesitarían mejorar sus atributos por ser este más difícil que los eventos normales.

Una vez ideado el sistema de juego se decidió que se dividiría el proyecto en dos fases: la fase 1 sería el sistema de creación de personajes y la fase 2 el juego. Debido a la escasez de tiempo, la fase 2 no se podría completar como se desearía por lo que se desarrollaría sólo a modo de demo

(demostración). Esto es algo que hacen con muchos juegos para mostrar al público como va a ser el juego antes de sacar el juego completo en sí.

Tras esto se comenzó con la programación de la fase 1. En determinado momento se vio necesario el uso de Javascript para programar los formularios que permiten crear al personaje. Concretamente para el sistema de reparto de atributos. Para la parte estética de la aplicación se han utilizado muchas imágenes extraídas de internet, concretamente de la página Pinterest.

Surgió la necesidad de decidir si se permitía al usuario a crear un personaje sin estar logueado. Puesto que muchas veces el hecho de tener que registrarte para usar una página es algo que produce rechazo entre los usuarios y a menudo acaban no usando la página por no tener que registrarse, se decidió que se permitiría crear al personaje sin necesidad de estar logueado, pero si se quería guardar en la base de datos de la aplicación sí que era necesario login. Esto implicaba una vista de visualización del personaje tras haberlo creado pero previa al almacenamiento de este, con un botón para guardarlo en la base de datos, el cual te enviaría a la página de login si el usuario no estuviera logueado aún. Tras hacer login de esta manera, el programa debería enviarte de nuevo a la página de la vista del personaje para poder proceder al guardado del mismo sin perder la creación realizada.



*Ilustración 5. Muestra imagen personaje*

Para personalizar un poco más a los personajes se buscaron imágenes de cada una de las 80 combinaciones posibles de género, raza y clase. De esta forma, se le puede poner aspecto al personaje, lo cual ayuda a la inmersión en el juego.

En cuanto al sistema del juego en sí, tras pensar las reglas que tendrían los combates y las distintas pruebas que habilidad que tuvieran que hacer los personajes para superar los distintos obstáculos con los que se encontrasen, sólo quedaba pensar como hacer el sistema de los eventos. Cada evento comienza con una descripción para la puesta en situación. El evento transporta al jugador por un pequeño hilo argumental hasta que en un momento dado el jugador tiene que tomar decisiones para ver que rumbo tomar. Estas decisiones van acompañadas de pruebas de habilidad que dependerán de los atributos del personaje y de una tirada de dados aleatoria. Dependiendo de si se tiene éxito o no, el personaje será guiado en una dirección u otra. En ocasiones se evitarán combates y en otras no se podrán eludir. En el caso de la demo, siempre hay que combatir, pero las decisiones previas pueden hacer el combate posterior más sencillo o complicado.

El evento de la demo puede finalizar por tres causas: los enemigos son abatidos, el personaje es abatido o el personaje huye. Si los enemigos son abatidos el personaje será recompensado, sin embargo, si el personaje es abatido, este será penalizado pudiendo hasta morir, de manera que ese personaje ya no podría volver a utilizarse. En el caso de que el personaje huya con éxito, no será ni recompensado ni penalizado en manera alguna.



El juego aún no contiene un sistema de monetización, pero es algo que se puede incluir más adelante. En vez de hacer pagar a los usuarios por usar la aplicación como si se estuvieran comprando un juego, es mejor dejar el acceso gratis para que el juego llegue a más personas, pero limitarlo si la persona está en modo gratuito, creando dos tipos de usuarios, como podrían ser los usuarios normales y los usuarios premium. Estas limitaciones pueden ser de diversas naturalezas como limitar el número de personajes que un usuario normal puede crear o limitar las razas y clases a las que pueden acceder o limitar el número de eventos distintos que le pueden aparecer. Por ejemplo, se podría hacer que los usuarios normales sólo pudieran crear tres personajes con su cuenta, sólo pudieran acceder a 3 razas y 5 clases y sólo les aparecieran combinaciones de 3 escenarios, 3 tipos de enemigos y 3 tipos de misiones pudiendo solamente jugar 27 ( $3 \times 3 \times 3 = 27$ ) eventos diferentes, mientras que los usuarios premium podrían crear todos los personajes que quisieran, podrían acceder a todas las razas y clases diferentes que haya y les aparecerían todas las combinaciones existentes pudiendo jugar en cientos de eventos diferentes. Esto no se ha introducido aún en la aplicación, pero si se quisiera sacar un beneficio de la misma sería una buena manera de hacerlo.

Lo que sí que se ha implementado es un sistema de ranking. En este ranking se pueden ver a todos los personajes del mundo en las posiciones que ocupan teniendo en cuentas los puntos ganados con cada uno. Estos puntos se ganan consiguiendo victorias en las partidas y se pierden con las derrotas. Cada victoria vale un punto y cada derrota resta un punto.

La idea es que los enemigos nivelen su dificultad dependiendo de los puntos que tenga el personaje de este modo se adaptan al poder de cada personaje evitando así que el juego se vuelva demasiado fácil o difícil. Esta funcionalidad de la variación de la dificultad se introduciría en la versión completa del juego.

Como se ha comentado, la idea a futuro es completar el juego creando un sistema con varios eventos o misiones a realizar, con distintos encuentros o problemas que surjan en esas misiones y varios enemigos o PNJ (personajes no jugadores) que aparezcan en esos encuentros. De esta manera, poder utilizar un sistema “Rogue Like” tal y como se comentó anteriormente en el presente documento. Además, se podría incorporar un sistema de niveles de dificultad en el cual los eventos son más difíciles cuanto mayor es la dificultad por lo que será necesario tener atributos mayores, y por otro lado, también las recompensas por superarlo y las penalizaciones por fallar serán mayores con la dificultad.



*Ilustración 6. Muestra imagen enemigo*

## 4.2. Especificación de requisitos

### 4.2.1. Introducción

En este caso no hay ningún cliente que nos haya especificado ningún requisito para el programa así que se elaborará la especificación con los requisitos propios autoimpuestos.

Esta especificación se ha elaborado para el desarrollo del programa web del presente documento, el cual se trata de un juego de entretenimiento al que se puede acceder a través de internet.

#### Propósito

El propósito de esta especificación es marcar, de manera clara y entendible por todos, los objetivos, requisitos y restricciones del proyecto.

Va dirigida a aquellas personas que se encarguen del desarrollo y mantenimiento de la aplicación, a los usuarios finales de la misma y así como a las demás partes que puedan intervenir directa o indirectamente en el proyecto.

#### Ámbito del sistema

La aplicación será nombrada “Caves & Orcs” siempre que haya consenso con la aceptación de dicho nombre por todas las partes implicadas.

La aplicación debe permitir la gestión de usuarios y la gestión de personajes.

No se podrán realizar cambios en el modo de juego desde el nivel de los usuarios ni administradores.

Actualmente hay aplicaciones similares, pero esta destaca por su sencillez y rápida adaptación del usuario que no requiere de un tutorial para poder manejar la aplicación. A pesar de ello si que existirá un manual donde se indique el funcionamiento y las reglas del juego.

#### Definiciones, Acrónimos y Abreviaturas

Definiciones:

- Administrador principal: El administrador con número de identificación igual a 1, no puede ser borrado ni modificado.
- Administrador: Cualquier persona logueada en la aplicación con derechos de administrador.
- Usuario normal: Cualquier persona logueada en la aplicación, pero sin derechos de administrador.
- Usuario: Cualquier persona que accede a la aplicación, esté logueado o no.

#### Visión General de la especificación

Consta de cuatro partes diferenciadas:

- Introducción: Es la sección actual, cual sirve de resumen breve de la actuación.
- Descripción general: Aquí se explicarán el contexto, el entorno y las funciones de la aplicación.

- Requisitos específicos: En esta parte se darán todos los detalles como los requisitos y las restricciones.
- Apéndices: Por último, se incluirá toda aquella la información no englobada en los apartados anteriores.

### 4.2.2. Descripción General

A continuación, se detallarán los condicionantes que afectarán al programa como el enfoque que buscado, las necesidades de los usuarios y los datos que se quieren manejar.

#### Perspectiva del Producto

La interacción del producto será a través de la web. La aplicación se alojará en un servidor al cual se accederá a través de la URL.

No interactuará con ninguna otra aplicación.

#### Funciones del Producto

Las funciones que deberá desempeñar son las siguientes:

- Generación, modificación y eliminación de usuarios: Los usuarios podrán registrarse en la aplicación y modificar sus datos cuando lo deseen. Los administradores podrán modificar los datos del resto de los usuarios normales y eliminar sus cuentas. El administrador principal podrá modificar los datos del resto de los usuarios (tanto normales como administradores) y eliminar sus cuentas. Todos los administradores también pueden nombrar a usuarios normales como administradores.
- Generación, modificación y eliminación de personajes: Se deben poder generar personajes por parte del usuario. Este introducirá los datos de su usuario en el login y tras ello podrá crear los personajes que se adheridan a su cuenta. Los atributos del personaje se podrán ir modificando como funcionalidad del propio juego. El usuario podrá también eliminar a sus personajes cuando lo desee.
- Gestión del juego: Los usuarios podrán avanzar a través del juego con la toma de decisiones eligiendo distintas opciones que se les ofrecerán, esto determinará el resultado de cada evento en el juego.
- Registro de logros de jugador: Habrá un sistema de registro de los logros que vaya consiguiendo el jugador con cada uno de sus personajes, creando así un ranking que los usuarios podrán revisar cuando deseen.

#### Características de los Usuarios

Los usuarios van a ser cualquier persona que acceda desde el ordenador de su casa a la página web, por lo que debemos asumir que no tienen ningún conocimiento informático. Por ello, la aplicación debe ser muy visual, fácil de usar y lo más intuitiva posible.

### Restricciones

Existen las siguientes restricciones a tener en cuenta para el programa:

- Debe de poder ejecutarse desde cualquier ordenador con internet.

### Suposiciones y Dependencias

Se supone que los usuarios saben manejar de manera básica un ordenador y el navegador de internet. Con ello no deberían tener problemas para utilizar la aplicación.

La única dependencia, es con el navegador que se utilice para ejecutarla. Deberá ser compatible con los navegadores más comunes.

### Requisitos Futuros

Se deberá ampliar la parte del juego para que haya más eventos e incluso la posibilidad de seguir un hilo argumental.

### 4.2.3. Requisitos Específicos

En esta sección se detallarán todos los requisitos que debe cumplir el programa. Estos requisitos condicionarán y definirán el funcionamiento del programa según las especificaciones recibidas por el cliente.

### Interfaces Externas

El programa se visualizará a través de la pantalla de un ordenador y su navegación será mediante distintas ventanas del navegador. Su manejo será mediante ratón y teclado.

### Funciones

A continuación, se describirán todas las funciones del sistema. Más adelante en el apartado de los apéndices se mostrarán los requisitos mediante la metodología de recolección de requisitos MBD.

- OBJ-1 -> Gestión de información de los usuarios: se deberá poder introducir una ficha para cada usuario que contenga su nombre de usuario, contraseña, nombre, apellidos, email y rol. También deberán poder consultarse estos datos, modificarse y eliminarse.
- OBJ-2 -> Gestión de información de los personajes: se deberá poder introducir una ficha para cada personaje que contenga su nombre, salud, genero, raza, clase, atributos y el usuario al que está asociado. También deberán poder consultarse estos datos, modificarse y eliminarse. También se podrá consultar un listado con los personajes del usuario.
- OBJ-3 -> Jugabilidad: se deberá poder jugar con cada personaje y al finalizar cada partida habrá una variación en los atributos del personaje. Durante el juego se deberán poder ir eligiendo distintas decisiones para obtener distintos resultados.
- OBJ-4 -> Registro de logros: Cada vez que se termine una partida, se otorgarán puntos de logro al personaje de modo que estos queden registrados en un ranking de logros.

### Requisitos de Rendimiento

La aplicación será diseñada para soportar todos los usuarios que quieran conectarse sin que colapse el servidor. Por ello, se deberá elegir un servidor adecuado para la aplicación.

### Restricciones de Diseño

- El tiempo de respuesta de la aplicación ha de ser lo menor posible.
- La aplicación deberá procesar todas las peticiones que se realicen desde los distintos equipos.

### Atributos del Sistema

Los atributos de calidad del programa serán los siguientes:

- Fiabilidad:

La aplicación será estable y sin errores.

- Mantenibilidad:

Este software puede estar sujeto a modificaciones por parte del desarrollador, no se dará servicio técnico del mismo más allá del manual de usuario y las notas de las actualizaciones.

- Portabilidad:

El programa podrá utilizarse desde cualquier ordenador que cumpla con las características requeridas para soportarlo.

- Seguridad:

Se accederá al contenido del mismo sin necesidad de un usuario y contraseña, pero las funcionalidades estarán limitadas. Una vez logueado con credenciales personales y únicas de cada usuario, se tendrá acceso a más funcionalidades.

Habrán distintos perfiles de usuario con distintos niveles de acceso.

- Administrador principal: Tendrá acceso a todas las funcionalidades excepto modificar o borrar su propio usuario.
- Administrador: Tendrá acceso a todas las funcionalidades, excepto modificar o borrar a otros administradores.
- Usuario normal: Tendrá acceso a todas las funcionalidades, excepto modificar o borrar a otros usuarios.
- Usuario visitante: Tendrá acceso limitado a las funcionalidades.

### 4.2.4. Apéndices

En este apartado se mostrarán los requisitos mediante la metodología de recolección de requisitos MBD, para ello se deben consultar las tablas de objetivos, actores y casos de uso que se encuentran en el Anexo 1 de la presente memoria.

## 5. Diseño de la aplicación

Para el diseño de la aplicación como ya se ha mencionado anteriormente se ha utilizado el lenguaje Java junto con el IDE Eclipse para el desarrollo del todo el código. Se ha utilizado para alguna de las vistas algo de Javascript, para lo cual se ha usado el programa Visual Studio Code como base para la escritura de esa parte del código, ya que está más preparado para Javascript de lo que lo está Eclipse.



Ilustración 7.  
Logo MySQL

El sistema de gestión de la base de datos es MySQL, por lo que se ha usado el programa Heidi SQL para la visualización y las pruebas de la misma. Añadir en este punto que la aplicación está programada para que si no encuentra la base de datos por la causa que sea genere una nueva base de datos con los datos necesarios para que la aplicación funcione correctamente. Claro está, esto sería para casos en los que se está probando la aplicación ya que la pérdida de la base de datos supone también la perdida de todos los datos introducidos por los usuarios, es decir todos los usuarios y los personajes.

Como capa previa a la base de datos se ha utilizado Hibernate el cual nos permite utilizar los objetos de Java con una base de datos relacional como la que utilizamos en este proyecto. Hibernate trabaja sobre el sistema JDBC (Java Database Connectivity) pero mejorando su uso para poder mediante menciones en las clases POJO (Plain Old Java Object) usar estas directamente para la base de datos.



HIBERNATE

Ilustración 8.  
Logo Hibernate

En cuanto a las clases de Java, se ha seguido el patrón Modelo-Vista-Controlador separando las vistas de los objetos POJO y de los servlets, utilizando también una capa de servicios para las operaciones lógicas y una capa DAO para la gestión de la base de datos.

Se han agrupado las distintas capas separando en tres partes todo: la parte de la gestión de usuarios, la parte de la gestión de personajes y la parte de gestión del juego.

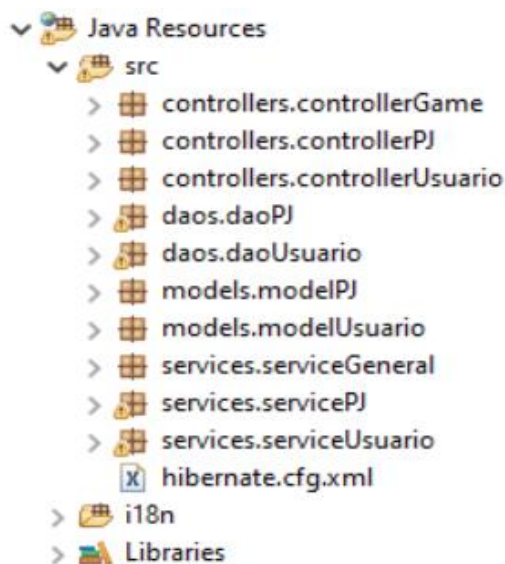


Ilustración 9. Paquetes de clases

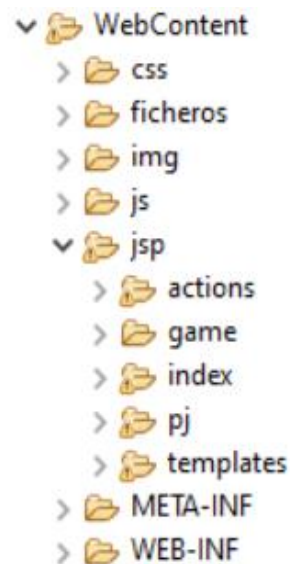


Ilustración 10. Carpetas del contenido

Las vistas a su vez tienen una subdivisión más ya que en la parte de la gestión de los usuarios se ha dividido en las carpetas “index”, “actions” y “templates”. Como sus nombres indican en “index” tenemos lo referente a la vista inicial, en “templates” están todas aquellas vistas que son utilizadas dentro de otras vistas como pueden ser las cabeceras o el pie de página y en “actions” están el resto de las vistas necesarias para la gestión de los usuarios.

Para las vistas se ha utilizado HTML con CSS y Bootstrap como apoyo a la parte de CSS. Todas las vistas tienen en común tres elementos: la cabecera, el pie y los mensajes emergentes. Estas tres vistas incrustadas (con el elemento @include) se encuentran en la carpeta “templates” tal y como se ha mencionado.

A continuación, se muestra el diagrama de E-R para la base de datos, este va a ser el mismo que para las clases de Java puesto que con Hibernate, mediante las anotaciones, se utilizan las clases POJO como tablas de la base de datos. Se utiliza la anotación @Entity para indicar a Hibernate que esa clase debe usarla como una entidad de la base de datos.

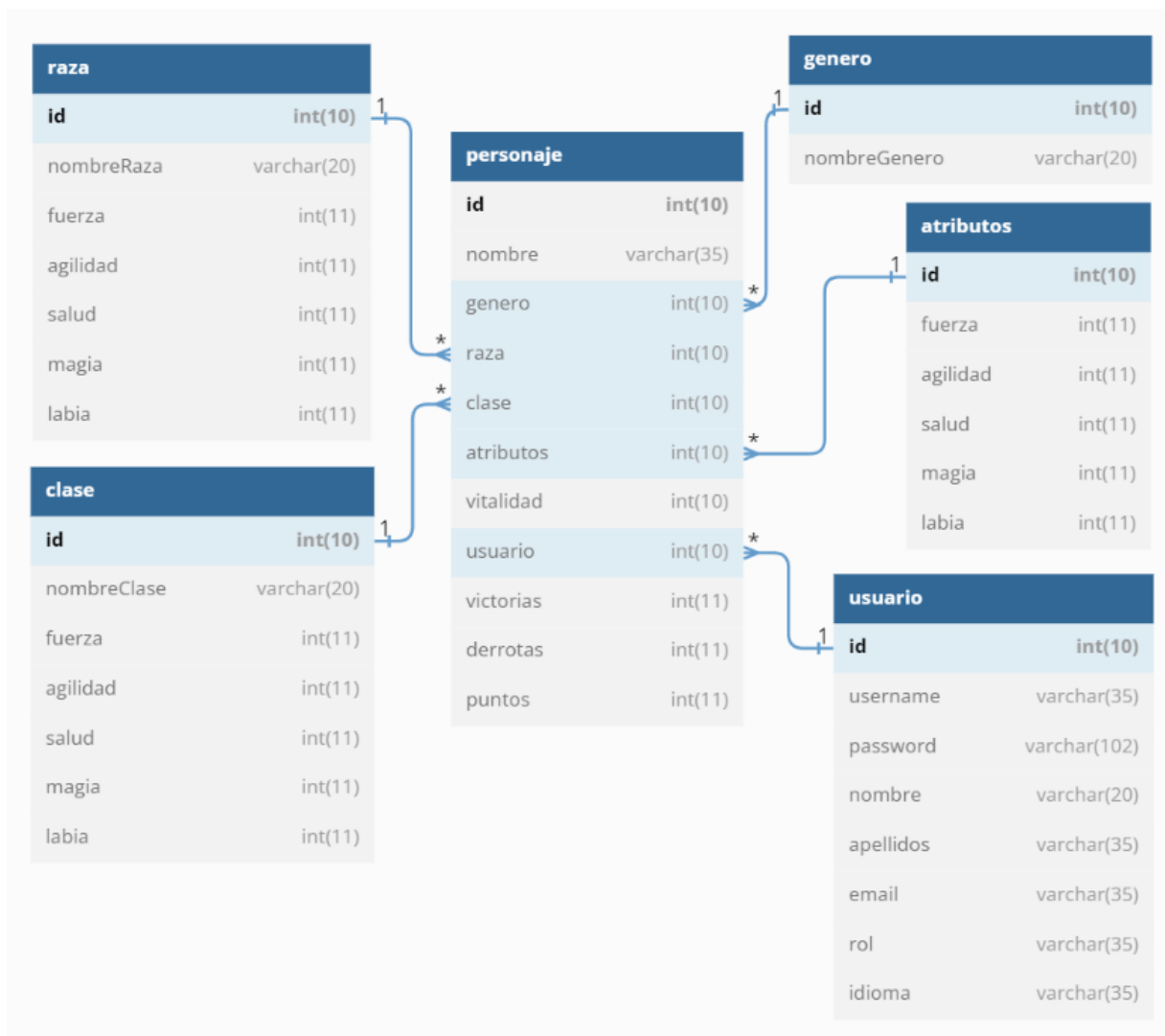


Ilustración 11. Diagrama E-R Base de datos

Como se puede apreciar todas las tablas giran en torno a la tabla personaje. Este sistema nos permite acceder los datos de cualquiera de las otras tablas solamente con utilizar la tabla personaje. Al estar utilizando Hibernate, mediante las anotaciones `@ManyToOne` y `@JoinColumn` indicamos las relaciones entre tablas y de este modo accediendo al objeto personaje podemos fácilmente acceder también a los valores de las tablas a las que está ligado.

```
@Entity
public class Personaje {

    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    private String nombre;
    private int salud;
    private int victorias = 0;
    private int derrotas = 0;
    private int puntos = 0;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "genero_id", nullable = false)
    private Genero genero;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "raza_id", nullable = false)
    private Raza raza;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "clase_id", nullable = false)
    private Clase clase;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "atributos_id", nullable = false)
    private Atributos atributos;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "usuario_id")
    private Usuario usuario;
```

*Ilustración 12. Clase Personaje*

En cuanto a los casos de uso se han separado de la misma manera que las clases, se han creado tres subsistemas:

- Gestión de usuarios, con los siguientes casos de uso:
  - o UC-1: Alta usuario
  - o UC-2: Modificación usuario
  - o UC-3: Eliminación usuario
- Gestión de personajes, con los siguientes casos de uso:
  - o UC-4: Alta personaje
  - o UC-5: Modificación personaje
  - o UC-6: Eliminación personaje
- Gestión de juego, con los siguientes casos de uso:
  - o UC-7: Jugar partida

Los diagramas de los casos de uso se encuentran en el Anexo 2 de la presente memoria.



## 6. Conclusiones e implicaciones

Desarrollando esta aplicación se ha descubierto la dificultad que entraña la creación de un videojuego puesto que, aunque sea uno sencillo y con pocos recursos como el del presente proyecto, se necesitan muchas horas de dedicación y surgen muchos problemas hasta en situaciones que a priori se pensaba que eran tarea fácil. Ha menudo los problemas que más han retrasado el trabajo han sido aquellos que no se esperaba tener.

Por otro lado, se necesita a varios especialistas si se quiere hacer un trabajo de calidad. Es necesario a alguien que escriba el argumento y los diálogos, otra persona que se encargue de las ilustraciones o partes gráficas, alguien dedicado a la música, otro para el desarrollo del back end, otro en el front end y alguien que coordine todo ello. Si una sola persona desarrolla un juego de manera individual tiene que ser multidisciplinar y asumir todos los roles mencionados y esto lleva mucho trabajo y tiempo. También es cierto que día a día van surgiendo tecnologías que facilitan mucho estos trabajos, no me imagino lo difícil que debió ser para los primeros desarrolladores la creación de esos videojuegos de 8 bits de los que se hablaba en el apartado de fundamentación teórica y los antecedentes. Es sorprendente pensar en todo el trabajo que han realizado todos los creadores de videojuegos para llegar desde aquellos sencillos juegos hasta los complejos y espectaculares grandes estrenos que hay actualmente.

Por ello, una sola persona puede desarrollar videojuegos, pero nunca podrá desarrollar proyectos de alta calidad, esto antes o después implica que los creadores de videojuegos, si quieren realizar grandes éxitos deben trabajar en equipo con otros profesionales especializados. Aunque crear un videojuego por uno mismo es muy satisfactorio, la parte de crear un videojuego en equipo también puede ser emocionante dando lugar a buenos vínculos con los demás implicados.

Concretamente en este proyecto, el autor se ha dado cuenta de que no estaba desarrollando el juego para el fin para el que se espera, que es la entrega del proyecto de fin de ciclo, sino que lo estaba desarrollando para su satisfacción personal y para dar su primer paso en lo que podría ser un futuro de desarrollador de videojuegos. Se ha aprendido mucho, de como plantearse los problemas que surgen y como ir superándolos y eso sienta las bases para tener perseverancia y no desistir en futuros proyectos, sobre todo al observar el resultado final una vez superados todos esos obstáculos.

## 7. Propuestas de mejora y líneas de futuro

### 7.1. Jugabilidad

Las siguientes mejoras van unidas a posibles líneas de trabajo futuro que se podrían dar para que la aplicación fuese más completa y entretenida para los jugadores.

La mejora principal para el proyecto es ampliar el juego a su versión definitiva. Esto implica aumentar el número de eventos que el jugador puede encontrarse y darle un objetivo final creando un evento más difícil, el cual para ser superado hay que mejorar al personaje lo suficiente con los eventos normales.

También se podría introducir un inventario en el cual el personaje pueda llevar consigo armas, armadura u objetos. Esto permitiría muchas más opciones a la hora de interactuar con el entorno. Las armas por ejemplo pueden mejorar las tiradas de atributo en los combates, las armaduras harían más difíciles las tiradas de atributo en combate a los enemigos y los objetos pueden ayudar para otras tiradas de atributo como por ejemplo puede ser un intento de huida con agilidad o un intento de convencer a alguien con una tirada de labia. También pueden mejorarte en ese evento un atributo concreto, como por ejemplo tomar una poción que aumente la vitalidad para tener más salud en un combate.

Otra mejora sería la existencia de un mapa sobre el cual el personaje puede decidir donde viajar para hacer las misiones específicas de cada zona, de esta manera se podrían ir completando zonas del mapa al hacer todas las misiones de la misma y cuando el personaje hubiera completado todas, se le desbloquease la zona final que debe superar para superar el juego. De este modo se podría incluso crear un hilo argumental creando una historia con sentido a todo el juego.

También existe la posibilidad de la introducción de un sistema de monetización como el comentado anteriormente, en el cual tengas que pagar para poder crear y almacenar más personajes de los inicialmente permitidos. Esta mejora si la unimos a las anteriores, también se podría ampliar a la compra de ítems o equipo para tener personajes más poderosos.

Lo anterior podría estar motivado por querer estar más alto en el sistema de ranking, de modo que a los puestos más altos se les puede dar premios para mejorar aún más a sus personajes.

Por otro lado, se podrían hacer torneos temporales en los cuales durante por ejemplo diez días hay que intentar acumular todos los logros posibles y quienes consigan más puntos de logro recibirán personajes inéditos y únicos con los que poder jugar.

### 7.2. Plataforma

Para poder ampliar el nicho de mercado del juego, este se podría migrar a otras plataformas como puede ser para móvil o aplicación de escritorio del ordenador. De este modo, se podría conseguir una mayor monetización al atraer a más jugadores al juego. Aún así por las características del mismo seguiría siendo gratuita su obtención, pero con los sistemas de monetización mencionados.

### **7.3. Diseño**

Actualmente los textos de la aventura están metidos en los archivos de idiomas, escritos directamente. Como mejora a futuro estos textos deberían almacenarse de una manera más manejable y permitir a los administradores que editaran o incluso crearan nuevos textos para crear nuevas historias.

El fondo de la aplicación siempre es el mismo, la taberna. Esto más adelante debería cambiarse para que el fondo cambiase dependiendo de la parte de la aventura en la que se encontrase el personaje.

También se podría incluir música de fondo e incluso sonidos con los ataques y demás acciones.

El envío de correos informativos es una funcionalidad que está preparada en el código, aunque deshabilitada para evitar el envío de correos durante las pruebas. Está preparado para enviar correos al realizar un registro, al cambiar la contraseña y cuando un administrador borra a un usuario. Debería mejorarse el diseño de los correos y además esta funcionalidad se podría usar por ejemplo para enviar correos informativos de novedades introducidas en el juego o de próximos torneos.

## 8. Bibliografía y/o referencias

- Cursos:
  - <https://www.youtube.com/@bluuweb>
  - <https://www.youtube.com/programacionats>
  - <https://www.youtube.com/@pildorasinformaticas>
- CSS:
  - <https://getbootstrap.com/>
- Dudas:
  - <https://stackoverflow.com/>
  - <https://www.w3schools.com/>
  - <https://developer.mozilla.org/es/>
- Imágenes y diseño:
  - <https://www.pinterest.es/>

## 9. Anexos

### Anexo 1: Tablas de objetivos, actores y casos de uso

#### Objetivos

OBJETIVO	OBJ-1: Gestión de información de los usuarios
DESCRIPCIÓN	El sistema deberá poder introducir una ficha para cada usuario que contenga su información.
SUBOBJETIVOS	-> El sistema deberá permitir consultar los datos. -> El sistema deberá permitir modificar los datos. -> El sistema deberá permitir eliminar los datos.

OBJETIVO	OBJ-2: Gestión de información de los personajes
DESCRIPCIÓN	El sistema deberá poder introducir una ficha para cada personaje que contenga su información.
SUBOBJETIVOS	-> El sistema deberá permitir consultar los datos. -> El sistema deberá permitir modificar los datos. -> El sistema deberá permitir eliminar los datos.

OBJETIVO	OBJ-3: Jugabilidad
DESCRIPCIÓN	El sistema deberá permitir jugar los eventos con un personaje.
SUBOBJETIVOS	-> El sistema deberá permitir entrar al juego. -> El sistema deberá permitir tomar elecciones en el juego.

OBJETIVO	OBJ-4: Registro de logros
DESCRIPCIÓN	El sistema deberá permitir ver el ranking de los personajes.
SUBOBJETIVOS	-

#### Actores

ACTOR	ACT-1: Administrador principal
DESCRIPCIÓN	Este actor representa a la persona con permisos de administrador principal del sistema.
COMENTARIOS	-

ACTOR	ACT-2: Administrador
DESCRIPCIÓN	Este actor representa a la persona con permisos de administrador del sistema.
COMENTARIOS	-

ACTOR	ACT-3: Usuario normal
DESCRIPCIÓN	Este actor representa a los usuarios logueados que no tienen permisos de administrador del sistema.
COMENTARIOS	-

ACTOR	ACT-4: Usuario visitante
DESCRIPCIÓN	Este actor representa a los usuarios no logueados que tienen funcionalidades limitadas.
COMENTARIOS	-

### Requisitos de información

REQ. DE INFORMACIÓN	IRQ-1: Información de los usuarios
ACTORES	-> ACT-1: Administrador principal -> ACT-2: Administrador -> ACT-3: Usuario normal
OBJ Y REQ ASOCIADOS	-> OBJ-1: Gestión de información de los usuarios -> UC-1: Alta usuario -> UC-2: Modificación usuario -> UC-3 Eliminación usuario
DESCRIPCIÓN	El sistema deberá almacenar información correspondiente a los actores <ACT-1: Administrador principal>, <ACT-2: Administrador> y <ACT-3: Usuario normal>, en concreto su nombre de usuario, contraseña, nombre, apellidos, email y rol.
COMENTARIOS	-

REQ. DE INFORMACIÓN	IRQ-2: Información de los personajes
ACTORES	-> ACT-1: Administrador principal -> ACT-2: Administrador -> ACT-3: Usuario normal
OBJ Y REQ ASOCIADOS	-> OBJ-2: Gestión de información de los personajes -> UC-4: Alta personaje -> UC-5: Modificación personaje -> UC-6: Eliminación personaje
DESCRIPCIÓN	El sistema deberá almacenar información correspondiente a los personajes, en concreto su nombre, salud, genero, raza, clase, atributos y el usuario al que están asociados.
COMENTARIOS	-

## Casos de uso

REQ. FUNCIONAL	UC-1: Alta usuario
ACTORES	-> ACT-3: Usuario normal -> ACT-4: Usuario visitante
OBJ Y REQ ASOCIADOS	-> OBJ-1: Gestión de información de los usuarios -> IRQ-1: Información de los usuarios
DESCRIPCIÓN	El sistema deberá comportarse tal como se describe en el siguiente caso de uso <UC-1: Alta usuario> cuando el <ACT-4: Usuario visitante> vaya a darse de alta como un nuevo usuario.
PRECONDICIÓN	Se está dando de alta a un usuario.
SECUENCIA NORMAL	PASO -> ACCIONES
	P1 -> El actor <ACT-4: Usuario visitante> introduce sus datos como futuro usuario. P2 -> El sistema comprueba los datos. P3 -> El sistema guarda los datos.
POSCONDICIÓN	-
EXCEPCIONES	PASO -> ACCIONES
	P1 -> Si el actor <ACT-4: Usuario visitante> cancela la operación, el sistema no guarda ningún dato. A continuación, este caso de uso queda sin efecto. P2 -> Si no es posible guardar la información, el sistema notifica la incidencia. A continuación, este caso de uso queda sin efecto. P3 -> Si en la comprobación de los datos, el sistema encuentra algún error, se lo hace saber al actor <ACT-4: Usuario visitante> y se vuelve a P1 de la secuencia normal de ejecución.
COMENTARIOS	-

REQ. FUNCIONAL	UC-2: Modificación usuario
ACTORES	-> ACT-1: Administrador principal -> ACT-2: Administrador -> ACT-3: Usuario normal
OBJ Y REQ ASOCIADOS	-> OBJ-1: Gestión de información de los usuarios -> IRQ-1: Información de los usuarios
DESCRIPCIÓN	El sistema deberá comportarse tal como se describe en el siguiente caso de uso <UC-2: Modificación usuario> cuando el <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> vaya a modificar los datos de un usuario.
PRECONDICIÓN	Se están modificando los datos de un usuario.
SECUENCIA NORMAL	PASO -> ACCIONES
	P1 -> El actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> introduce los datos modificados del usuario. P2 -> El sistema comprueba los datos. P3 -> El sistema guarda los datos.
POSCONDICIÓN	-
EXCEPCIONES	PASO -> ACCIONES
	P1 -> Si el actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> cancela la operación, el sistema no guarda ningún dato. A continuación, este caso de uso queda sin efecto. P2 -> Si no es posible guardar la información, el sistema notifica la incidencia. A continuación, este caso de uso queda sin efecto. P3 -> Si en la comprobación de los datos, el sistema encuentra algún error, se lo hace saber al actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> y se vuelve a P1 de la secuencia normal de ejecución.
COMENTARIOS	-



REQ. FUNCIONAL	UC-3: Eliminación usuario
ACTORES	-> ACT-1: Administrador principal -> ACT-2: Administrador
OBJ Y REQ ASOCIADOS	-> OBJ-1: Gestión de información de los usuarios -> IRQ-1: Información de los usuarios
DESCRIPCIÓN	El sistema deberá comportarse tal como se describe en el siguiente caso de uso <UC-3: Eliminación usuario> cuando el <ACT-1: Administrador principal> o <ACT-2: Administrador> vaya a eliminar los datos de un usuario.
PRECONDICIÓN	Se están eliminando los datos de un usuario.
SECUENCIA NORMAL	PASO -> ACCIONES
	P1 -> El actor <ACT-1: Administrador principal> o <ACT-2: Administrador> selecciona el usuario a eliminar. P2 -> El sistema comprueba los datos y solicita confirmación. P3 -> El actor <ACT-1: Administrador principal> o <ACT-2: Administrador> confirma la operación. P4 -> El sistema elimina los datos.
POSCONDICIÓN	-
EXCEPCIONES	PASO -> ACCIONES
	P1 -> Si el actor <ACT-1: Administrador principal> o <ACT-2: Administrador> cancela la operación, el sistema no guarda ningún dato. A continuación, este caso de uso queda sin efecto. P2 -> Si no es posible eliminar la información, el sistema notifica la incidencia. A continuación, este caso de uso queda sin efecto. P3 -> Si en la comprobación de los datos, el sistema encuentra algún error, se lo hace saber al actor <ACT-1: Administrador principal> o <ACT-2: Administrador> y se vuelve a P1 de la secuencia normal de ejecución.
COMENTARIOS	-

REQ. FUNCIONAL	UC-4: Alta personaje
ACTORES	-> ACT-1: Administrador principal -> ACT-2: Administrador -> ACT-3: Usuario normal
OBJ Y REQ ASOCIADOS	-> OBJ-2: Gestión de información de los personajes -> IRQ-2: Información de los personajes
DESCRIPCIÓN	El sistema deberá comportarse tal como se describe en el siguiente caso de uso <UC-4: Alta personaje> cuando el actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> vaya a dar de alta a un nuevo personaje.
PRECONDICIÓN	Se está dando de alta a un personaje.
SECUENCIA NORMAL	PASO -> ACCIONES
	P1 -> El actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> introduce los datos del personaje. P2 -> El sistema comprueba los datos. P3 -> El sistema guarda los datos
POSCONDICIÓN	-
EXCEPCIONES	PASO -> ACCIONES
	P1 -> Si el actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> cancela la operación, el sistema no guarda ningún dato. A continuación, este caso de uso queda sin efecto. P2 -> Si no es posible guardar la información, el sistema notifica la incidencia. A continuación, este caso de uso queda sin efecto. P3 -> Si en la comprobación de los datos, el sistema encuentra algún error, se lo hace saber al actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> y se vuelve a P1 de la secuencia normal de ejecución.
COMENTARIOS	-

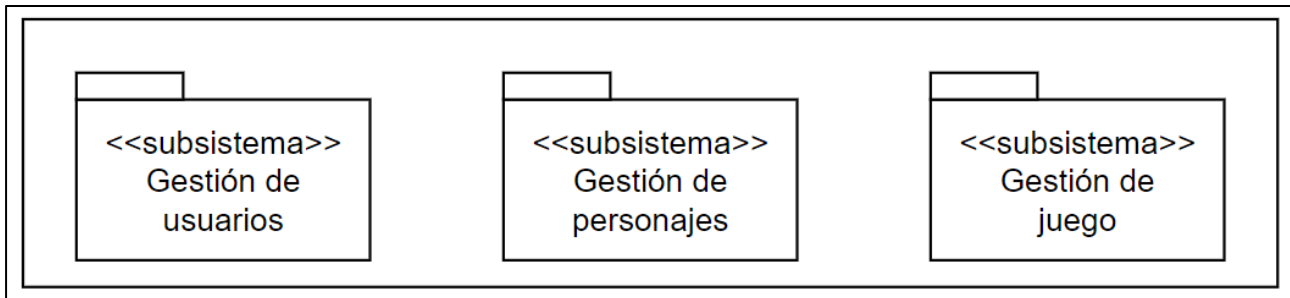
REQ. FUNCIONAL	UC-5: Modificación personaje
ACTORES	-> ACT-1: Administrador principal -> ACT-2: Administrador -> ACT-3: Usuario normal
OBJ Y REQ ASOCIADOS	-> OBJ-2: Gestión de información de los personajes -> IRQ-2: Información de los personajes
DESCRIPCIÓN	El sistema deberá comportarse tal como se describe en el siguiente caso de uso <UC-5: Modificación personaje> cuando el actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> vaya a modificar los datos de un personaje.
PRECONDICIÓN	Se están modificando los datos de un personaje.
SECUENCIA NORMAL	PASO -> ACCIONES
	P1 -> El actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> introduce los datos del personaje. P2 -> El sistema comprueba los datos. P3 -> El sistema guarda los datos
POSCONDICIÓN	-
EXCEPCIONES	PASO -> ACCIONES
	P1 -> Si el actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> cancela la operación, el sistema no guarda ningún dato. A continuación, este caso de uso queda sin efecto. P2 -> Si no es posible guardar la información, el sistema notifica la incidencia. A continuación, este caso de uso queda sin efecto. P3 -> Si en la comprobación de los datos, el sistema encuentra algún error, se lo hace saber al actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> y se vuelve a P1 de la secuencia normal de ejecución.
COMENTARIOS	-

REQ. FUNCIONAL	UC-6: Eliminación personaje
ACTORES	-> ACT-1: Administrador principal -> ACT-2: Administrador -> ACT-3: Usuario normal
OBJ Y REQ ASOCIADOS	-> OBJ-2: Gestión de información de los personajes -> IRQ-2: Información de los personajes
DESCRIPCIÓN	El sistema deberá comportarse tal como se describe en el siguiente caso de uso <UC-5: Modificación personaje> cuando el actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> vaya a eliminar los datos de un personaje.
PRECONDICIÓN	Se están eliminando los datos de un personaje.
SECUENCIA NORMAL	PASO -> ACCIONES
	P1 -> El actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> selecciona el personaje a eliminar. P2 -> El sistema comprueba los datos y solicita confirmación. P3 -> El actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> confirma la operación. P4 -> El sistema elimina los datos.
POSCONDICIÓN	-
EXCEPCIONES	PASO -> ACCIONES
	P1 -> Si el actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> cancela la operación, el sistema no elimina ningún dato. A continuación, este caso de uso queda sin efecto. P2 -> Si no es posible eliminar la información, el sistema notifica la incidencia. A continuación, este caso de uso queda sin efecto. P3 -> Si en la comprobación de los datos, el sistema encuentra algún error, se lo hace saber al actor <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> y se vuelve a P1 de la secuencia normal de ejecución.
COMENTARIOS	-

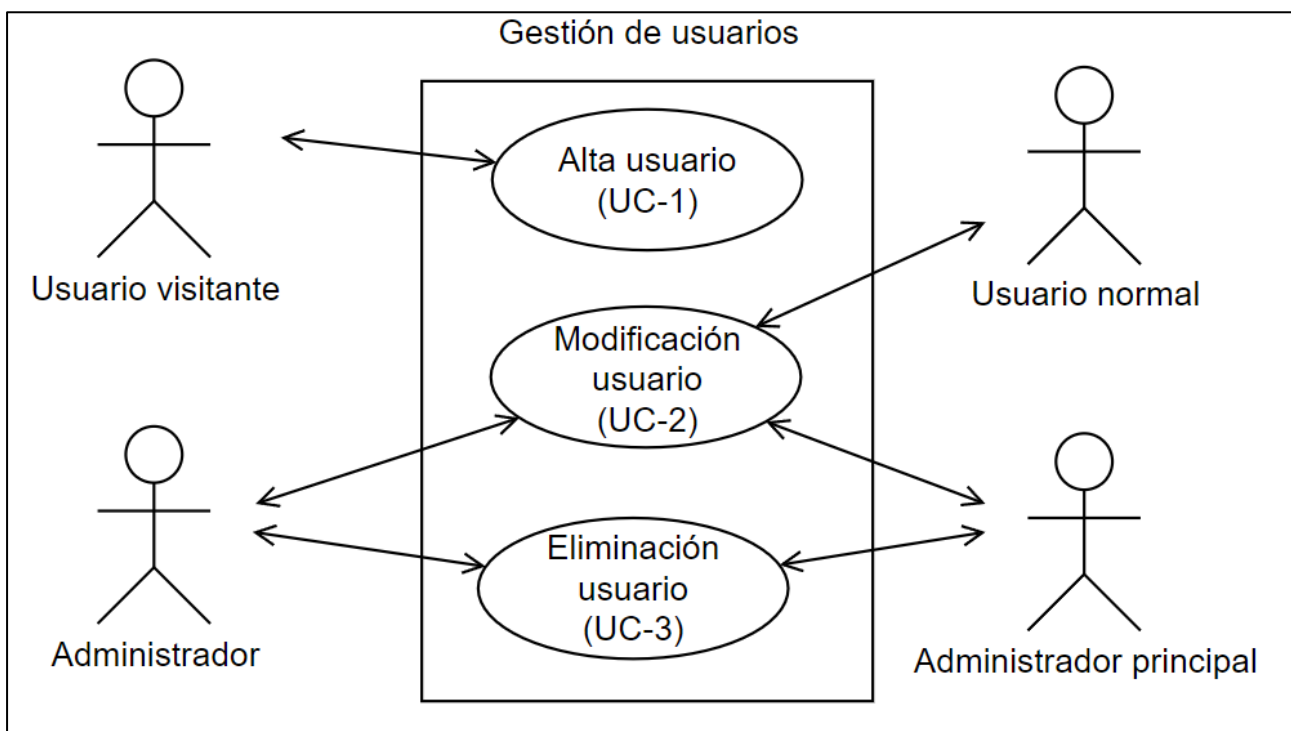
REQ. FUNCIONAL	UC-7: Jugar partida
ACTORES	-> ACT-1: Administrador principal -> ACT-2: Administrador -> ACT-3: Usuario normal
OBJ Y REQ ASOCIADOS	-> OBJ-2: Gestión de información de los personajes -> OBJ-3: Jugabilidad -> OBJ-4: Registro de logros -> IRQ-2: Información de los personajes
DESCRIPCIÓN	El sistema deberá comportarse tal como se describe en el siguiente caso de uso <UC-7: Jugar partida> cuando el <ACT-1: Administrador principal>, <ACT-2: Administrador> o <ACT-3: Usuario normal> vaya a jugar una partida con un personaje.
PRECONDICIÓN	Se está jugando una partida con un personaje existente.
SECUENCIA NORMAL	PASO -> ACCIONES
	<p>P1 -&gt; El actor &lt;ACT-1: Administrador principal&gt;, &lt;ACT-2: Administrador&gt; o &lt;ACT-3: Usuario normal&gt; comienza a jugar la partida con el personaje seleccionado.</p> <p>P2 -&gt; El sistema muestra el evento de la partida y pide seleccionar una opción.</p> <p>P3 -&gt; El actor &lt;ACT-1: Administrador principal&gt;, &lt;ACT-2: Administrador&gt; o &lt;ACT-3: Usuario normal&gt; selecciona una opción.</p> <p>P4 -&gt; El sistema comprueba los datos y devuelve un resultado.</p> <p>P5 -&gt; El sistema modifica el registro de logros y guarda los datos.</p>
POSCONDICIÓN	-
EXCEPCIONES	PASO -> ACCIONES
	<p>P1 -&gt; Si el actor &lt;ACT-1: Administrador principal&gt;, &lt;ACT-2: Administrador&gt; o &lt;ACT-3: Usuario normal&gt; cancela la operación, el sistema no guarda ningún dato. A continuación, este caso de uso queda sin efecto.</p> <p>P2 -&gt; Si no es posible guardar la información, el sistema notifica la incidencia. A continuación, este caso de uso queda sin efecto.</p> <p>P3 -&gt; Si en la comprobación de los datos, el sistema encuentra algún error, se lo hace saber al actor &lt;ACT-1: Administrador principal&gt;, &lt;ACT-2: Administrador&gt; o &lt;ACT-3: Usuario normal&gt; y este caso de uso queda sin efecto.</p>
COMENTARIOS	-

## Anexo 2: Diagramas de casos de uso

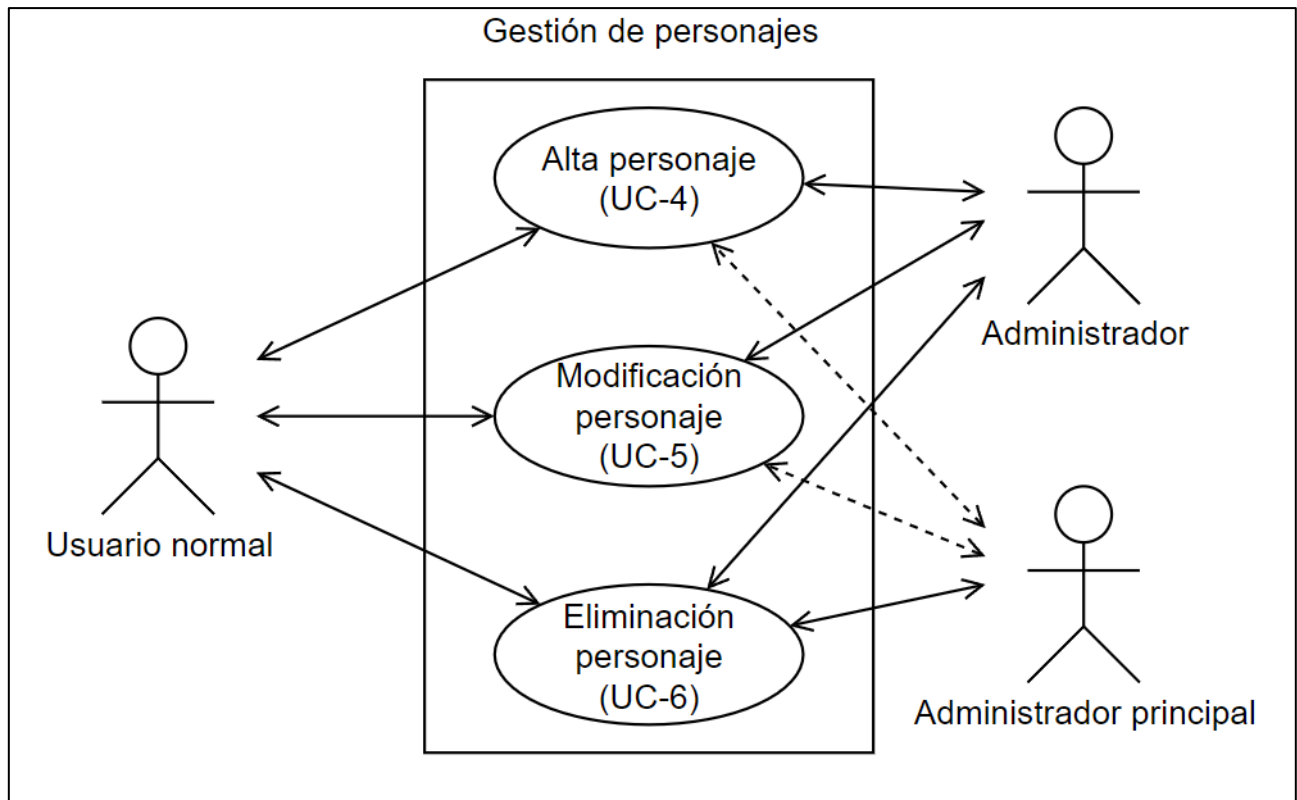
### Diagrama de subsistemas



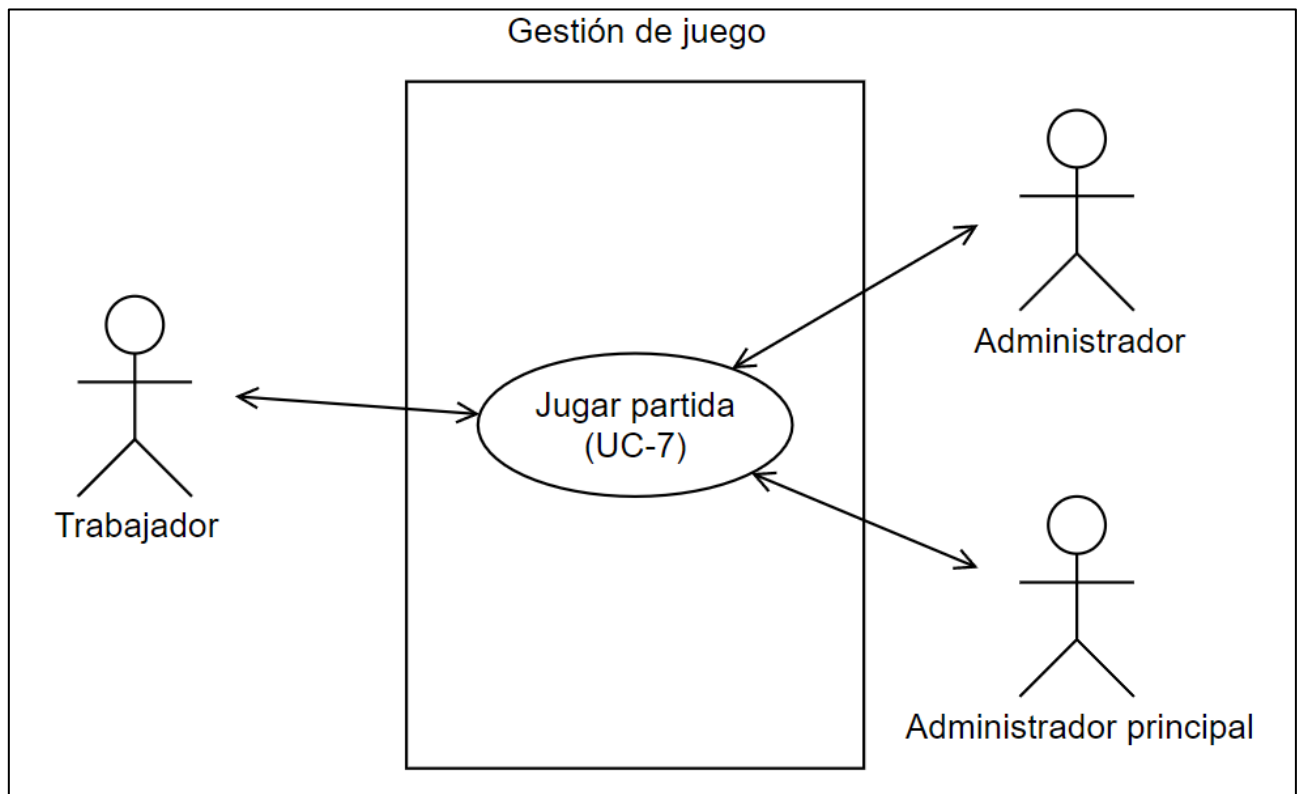
### Diagrama del subsistema Gestión de usuarios



## Diagrama del subsistema Gestión de personajes



## Diagrama del subsistema Gestión de juego



## Anexo 3: Manual de usuario

# Caves & Orcs

## Manual de usuario

### Introducción

En este manual se explicará el sistema de reglas del propio juego y su manejo por parte del jugador. Aunque es un juego sencillo y manejable por todos tiene sus propias reglas que conviene conocer para poder tomar mejores decisiones en el juego y mejorar el disfrute del jugador. Así que te animamos a que dediques unos minutos a echarle un vistazo a este manual para que puedas acabar con todos los orcos malvados del reino (sólo los malos, a los orcos buenos se les trata con cariño).

### Reglas del juego

Para la creación de personajes hay que tener en cuenta que cada raza y cada clase tiene distintas ventajas en los distintos atributos. Esos bonificadores marcarán los atributos base de tu personaje, el cual no podrá tener valores menores que los que se marcan durante el momento de la creación del mismo. Se te otorgarán 5 puntos a repartir como desees entre los atributos.

Una vez creado el PJ (personaje jugador: que pertenece a un jugador) este estará disponible para jugar y vivir aventuras, podrás utilizarlo siempre y cuando su vigor se mantenga por encima de cero. El vigor marca la salud y fortaleza del personaje y si llega a cero tu personaje habrá muerto y no se podrá usar más.

En las aventuras deberás tomar decisiones y elegir que acciones emprender en según qué situaciones, estas acciones llevan asociada una tirada de dados que indicará si consigues lo que pretendías o si por el contrario fallas en tu intento.

Arrancas el cartel de papel y te lo guardas con la intención de dirigirte sin demora hacia la cueva de los orcos. No tardas más de una hora en llegar a la montaña donde se encuentra la cueva y al poco rato oyes gritos y risas grotescas provenientes de un repecho de la ladera. Al asomarte con cuidado ves que se trata de un grupo de tres orcos los cuales están sentados comiendo alrededor de un fuego que han montado justo a la entrada de lo que parece una profunda gruta. Mientras comen bromean entre ellos golpeándose y gritándose en lo que podrá parecer un ambiente relajado para un orco. Distingues a uno de ellos como el que debe ser el jefe por su corpulencia y armadura. Aún no te han visto, puedes decidir que movimiento hacer a continuación. Puedes salir de tu escondite y retar a un duelo al jefe, o buscar un lugar desde el que poder atacar al jefe sin ser visto, o utilizar algún conjuro mágico, o huir.

Usar labia

Usar agilidad

Usar magia

Huida

*Ilustración 13. Manual - Elegir acciones*

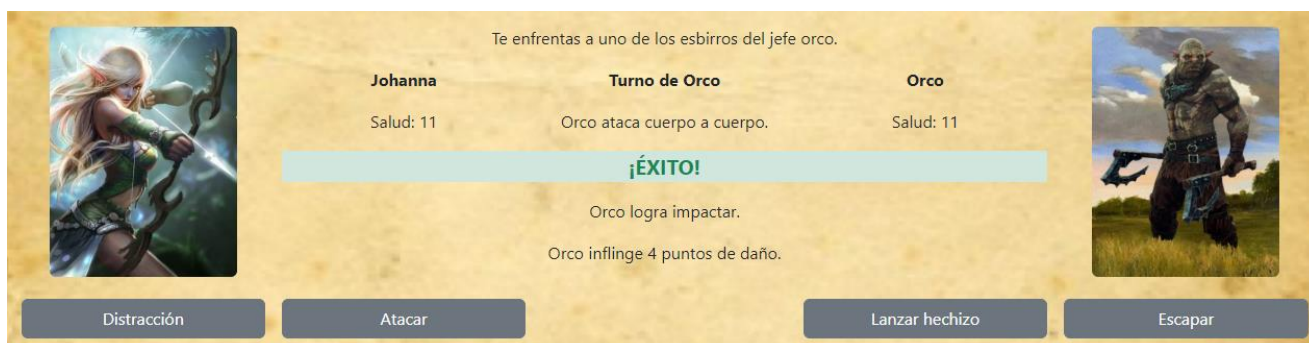


La tirada será con un dado de 20 caras y al resultado se le sumará el atributo implicado en la acción que desees tomar. Cada acción tendrá una dificultad que será un número fijado previamente. Para tener éxito, la suma anterior deberá igualar o superar la dificultad de la acción.

Estas elecciones y los resultados de las acciones determinarán que rumbo tomará la historia y los encuentros que tendrás. En ocasiones es posible que evites enfrentamientos directos y otras veces te verás inmerso en una intensa batalla.

Cuando en la acción esté implicado un PNJ (personaje no jugador: que pertenece al sistema), tendrás que realizar una prueba enfrentada contra él. En este caso, ambos tirareis dado de 20 caras y sumareis los resultados los atributos implicados respectivamente. Quien realiza la acción tiene que igualar o superar el resultado de quien sufre la acción para que esta tenga éxito.

En el caso del combate, funciona exactamente igual que lo mencionado en el párrafo anterior. Los combates funcionan por un sistema de turnos. Si realizas un ataque normal, utilizarás el atributo fuerza y tu enemigo utilizará su agilidad. Si tienes éxito causarás tanto daño como puntos de fuerza tengas. Si utilizas un ataque mágico, utilizarás el atributo magia y tu enemigo utilizará su agilidad. Si tienes éxito causarás tanto daño como puntos de magia tengas. Las otras dos opciones que tienes en el combate son distraer al enemigo o huir.



*Ilustración 14. Manual - Ejemplo batalla*

Distraer implica que realizas una finta o simplemente consigues que mire momentáneamente para otro lado y aprovechas para asestarle un golpe en algún punto vital. Para distraer ambos utilizareis la labia. Si tienes éxito, a continuación, realizarás automáticamente un ataque normal pero multiplicando por dos el atributo fuerza tanto para ver si aciertas como para el daño infligido. Si fallas en distraer no podrás realizar más acciones durante ese turno.

Para huir ambos utilizareis la agilidad. Con un éxito, consigues huir y finaliza la misión y con un fallo no te escapas y terminas el turno.

En las acciones realizadas por decisiones tomadas durante la trama (fuera del combate), si tienes éxito y este implica entrar en combate, tú comenzarás el primer turno, sin embargo, si fallas y ello te lleva a un combate, el enemigo tendrá ventaja y él comenzará el primer turno.

Si luchas contra varios enemigos tendrás que ir tumbándolos uno tras otro sin descanso hasta que no quede ninguno para vencer.

El combate puede terminar de tres maneras, que venzas a todos tus oponentes haciendo que la salud de cada uno llegue a cero y con ello obteniendo la victoria, que te venzan a ti haciendo que tu salud llegue a cero y con ello obteniendo la derrota o consiguiendo huir.

Si obtienes la victoria se te otorgarán 5 puntos para aumentar tus atributos como desees y obtendrás una victoria para tu puntuación en el ranking.

Si obtienes la derrota se te penalizará restándote un punto de cada atributo y obtendrás una derrota para tu puntuación en el ranking. Si a causa de la disminución de los atributos tu vigor llega a cero, tu personaje habrá muerto y no le podrás volver a utilizar, aunque le podrás visualizar en tu lista de personajes y en el ranking.

Si consigues huir no obtendrás premios ni penalizaciones quedándose tu personaje tal y como estaba antes de la aventura.

Entre aventuras se supone que los personajes descansan y recuperan todos sus puntos de salud por lo que siempre empezarán una misión con la salud al máximo.

En el ranking tendrán el puesto más alto quien más puntos tenga. Estos puntos son el resultado de restar el número de derrotas al número de victorias.

### Manejo y sistema de juego

Lo primero que debe hacer cualquier nuevo visitante es registrarse en la aplicación. Por supuesto, esto no es obligatorio, pero si no estás logueado no podrás guardar en tu cuenta los personajes que crees y por lo tanto no podrás jugar las aventuras. Sin loguearte puedes crear personajes y hacer pruebas de combinaciones para ver cuál te puede interesar, pero nada más.

Para registrarte pulsa en el enlace de registro y rellena el formulario. Una vez te hayas registrado deberás ir a la pantalla de login y loguearte en la aplicación. Para ello, en la pantalla de login, rellena tu usuario y contraseña. En este momento ya tendrás acceso a todo el sistema de juego.

El formulario de registro está sobre un fondo de papel antiguo. Incluye campos de texto para: 'Nombre de usuario:' (con el placeholder 'Nombre'), 'Contraseña:', 'Repetir Contraseña:', 'Nombre:', 'Apellidos:', y 'Email:'. En la parte inferior hay un botón gris con el texto 'Registrarse'.

*Ilustración 15. Manual - Registro*

El formulario de login está sobre un fondo de papel antiguo. Incluye campos de texto para: 'Nombre de usuario:' (con el placeholder 'Nombre de usuario') y 'Contraseña:'. Debajo de estos campos hay un botón gris con el texto 'Login'.

*Ilustración 16. Manual - Login*

Para crear un personaje vete al enlace de crear personaje y rellena los campos de nombre, género, raza y clase. Junto al formulario verás unos botones para ver la descripción de la raza o la clase que tengas seleccionadas en el formulario. Tras pulsar el botón para enviar los datos aparecerá una pantalla para que repartas puntos entre los atributos de tu personaje. Una vez repartidos pulsa el botón y se creará tu personaje.



Ilustración 17. Manual - Crear personaje



Ilustración 18. Manual - Repartir atributos

En este punto, te encontrarás ante una pantalla de visualización del personaje, pero aún no lo habrás guardado. Si pulsas el botón guardar y estás logueado el personaje se guardará en tu perfil, de lo contrario la aplicación te llevará a la pantalla de login para que hagas login y una vez logueado te mandará de nuevo a la pantalla de visualización del personaje donde podrás guardarlo.

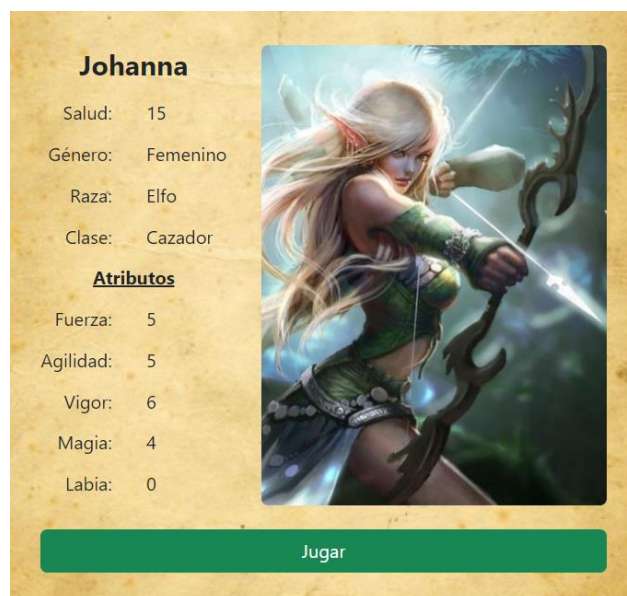


Ilustración 19. Manual - Vista personaje

Todos los personajes que crees los puedes ver en la lista de personajes. En esta lista puedes visualizar a cada personaje o borrarle. Ten en cuenta que si le borras también desaparecerá del ranking, más adelante se explica el funcionamiento del ranking.

En el modo visualización encontrarás un botón para comenzar una nueva partida. La partida comenzará narrándote cómo ha llegado tu personaje al lugar en el que se encuentra y poniéndote en situación. Se desarrollarán una serie de eventos hasta el momento en el que tendrás que ir tomando decisiones de que hacer pulsando a los distintos botones que te saldrán por pantallas. La historia puede llevarte a algún combate. En los combates cuando sea tu turno podrás realizar cuatro acciones: distraer al enemigo para asestarle un ataque furtivo, ataque normal, ataque mágico y huir. Dependiendo de los atributos de tu personaje deberás elegir sabiamente para salir airoso del combate.

Una vez finalizada la aventura, si has salido victorioso se te otorgarán puntos para repartir entre tus atributos, si has sido derrotado se te restarán puntos de los atributos y si has huido cual gallina no cambiarán tus atributos.

También puedes consultar el ranking de puntuaciones en el enlace de pertinente. Ahí podrás ver cómo van las puntuaciones de tus personajes respecto a las de los personajes de los demás jugadores.



Ranking					
Posición	Usuario	Personaje	Victorias	Derrotas	Puntos
1	Pepe	Chayane	11	2	9
2	Pepe	Goku	5	0	5
3	Pepe	Albatros	0	0	0
4	Pepe	Morgana	0	0	0
5	Pepe	Macareno	0	0	0
6	Pepe	Johanna	0	0	0
7	Pepe	Luisito	0	1	-1
8	Pepe	Leia	0	5	-5

Ilustración 20. Manual - Ranking