



Universidad
Zaragoza

Trabajo Fin de Grado

Aplicación web de asistencia para juegos de rol de
mesa

Assistant web application for tabletop role-playing
games

Autor

Rubén Subías Rodríguez

Director

Rubén Béjar Hernández

Titulación

Ingeniería Informática

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2022/2023

Aplicación web de asistencia para juegos de rol de mesa

Resumen

El avance de las tecnologías en la informática y electrónica ha generado un amplio abanico de posibilidades en el mundo del ocio. Al pensar en la relación de nuevas tecnologías y ocio, enseguida nos viene a la mente el mundo de los videojuegos, que han sufrido un auge en las últimas décadas. Sin embargo, otros tipos de entretenimiento más clásicos, como los juegos de mesa, se han visto potenciados por estas nuevas características.

Típicamente, los juegos de mesa se suelen relacionar con varios elementos fijos: un tablero, un listado concreto de reglas y algún elemento de azar, como cartas o dados. Sin embargo, hay muchos tipos de sistemas de juego en los que el principal elemento de ocio es la interacción verbal entre los jugadores, como sucede en los juegos de rol.

Sin lugar a dudas, el juego de rol que más ha trascendido a lo largo de los últimos años ha sido *Dragones y Mazmorras*, que incluso ha sido adaptado a otros formatos como el cine y los videojuegos. Sin embargo, las nuevas tecnologías también se han usado para reforzar el formato original y llevarlo a otra plataforma, como sucede con aplicaciones que adaptan estos juegos de mesa de manera on-line, permitiendo la existencia de partidas remotas.

Este Trabajo de Fin de Grado tiene como objetivo el desarrollo de una aplicación Web que sirva como una herramienta potente que los grupos de jugadores de *Dragones y Mazmorras* puedan utilizar en sus sesiones de juego y les permita una mejor experiencia. Esta aplicación incluiría formas de visualizar información relativa a las reglas del juego de forma rápida, así como métodos que permitan a los jugadores crear sus propios grupos e incluir toda la información personalizada que deseen en ellos.

El trabajo se ha dividido en tres fases principales. La primera fase ha sido la de planificación, en la que se han detallado los requisitos que debe satisfacer la aplicación y el diseño de esta. La segunda, que comprende la mayor parte del tiempo, ha sido la de desarrollo, en la que se han implementado todas las funcionalidades descritas en los requisitos. Finalmente, las últimas semanas de trabajo se han dedicado al desarrollo de pruebas y documentación de la aplicación.

Índice

Resumen	2
Índice	3
1. Introducción	4
2. Análisis del problema	5
2.1. Contexto	5
2.2. Fuentes de datos	6
2.3. Requisitos y Casos de Uso	6
2.3.1. Requisitos	6
2.3.2. Casos de uso	8
2.4. Interfaces de usuario	9
2.4.1. Relación interfaz con requisitos	10
3. Diseño de la solución	11
3.1. Arquitectura	11
3.1.1. Modelo de datos	11
3.1.2. Paquetes y clases	12
3.1.3. Componentes y conectores	16
3.1.4. Despliegue	19
3.2. Implementación	20
3.2.1. Funcionalidades finales	23
3.3. Pruebas	24
3.3.1. Pruebas de extremo a extremo	24
3.3.2. Pruebas de carga	24
4. Gestión del proyecto	27
5. Conclusiones y trabajo futuro	29
5.1. Conclusiones	29
5.2. Plan de futuro	29
Bibliografía	31
Lista de Figuras	32
Lista de Tablas	33
Anexo I - Casos de Uso	34
Anexo II - Bocetos de la GUI	49
Anexo III - Manual de Usuario	64

1. Introducción

Dragones y Mazmorras [1] es un juego de rol en el que varias personas se reúnen en sesiones para desarrollar una historia interactiva en la que todos participan. Cada grupo tiene al menos un *Master*, que es quien mantiene la dinámica de juego y prepara el contexto de la partida.

El conjunto de varias sesiones de juego con un mismo grupo se suele denominar “campaña”, y suelen tener ambientaciones fantásticas. Cada jugador interpreta a uno o más personajes, cuyos datos apunta en una ficha. A lo largo de las partidas, los personajes interactúan con un amplio abanico de enemigos, lugares, objetos mágicos y hechizos.

El proyecto propuesto surge de la idea de realizar una aplicación web que sirva de “asistente” para los jugadores de *Dragones y Mazmorras* 5ª edición. Dado que hay muchas aplicaciones ya existentes similares, cuya función principal es mostrar un listado de las reglas y componentes oficiales del juego de forma ordenada, se decidió ampliar las capacidades de la aplicación.

De esta forma, el proyecto se presenta como un sitio web que permite a los jugadores gestionar sus partidas de forma eficiente, permitiendo organizarse en grupos donde crear contenido personalizado, consultar toda la información relativa a sus personajes mediante accesos rápidos y crear entradas en una wiki propia para cada grupo.

El resto de este documento se ha estructurado de la siguiente forma:

- En el capítulo 2 se realizará un análisis completo del alcance del proyecto, describiendo el problema a tratar y listando los requisitos y casos de uso del sistema, junto a bocetos de la interfaz.
- El capítulo 3 se centrará en describir el diseño del sistema, mediante diagramas arquitecturales y algunos detalles de implementación.
- El capítulo 4 tratará cómo se ha llevado a cabo la gestión del trabajo y las metodologías usadas.
- En el capítulo 5 se expondrán las conclusiones tras finalizar el proyecto, y se propondrán posibles mejoras de la aplicación en un futuro.

2. Análisis del problema

Se va a desarrollar una aplicación Web que sirva de asistente para jugar a *Dragones y Mazmorras 5ª edición (D&D 5e)*. Esta aplicación tendrá varias funcionalidades que facilitarán al jugador montar y jugar sus partidas junto a otra gente, como poder consultar de forma rápida los componentes (hechizos, objetos y rasgos) que constan en el reglamento oficial.

Además, la aplicación permitirá a los usuarios juntarse en grupos, donde podrán crear sus propios componentes y fichas de personaje. Las fichas de personaje contendrán toda la información relativa a un personaje de jugador necesaria para jugar una partida. Además, estas fichas serán editables y permitirán añadir hechizos y objetos al libro de hechizos y al inventario del personaje, respectivamente, facilitando un acceso rápido a la información de dichos componentes.

Cada grupo contará con una wiki donde cualquier integrante podrá crear nuevas entradas con la información que considere, creando así un repositorio donde almacenar información sobre sus partidas, personajes o historias.

El usuario que cree un grupo será considerado el *Master* y, además de las funcionalidades anteriores, podrá crear sus propias notas (similares a las entradas de la wiki), que podrá utilizar a modo de apuntes para planificar las sesiones de juego y no serán visibles para el resto de integrantes del grupo.

2.1. Contexto

Como se ha mencionado anteriormente, hay varias aplicaciones que cumplen las necesidades básicas de un “asistente”, permitiendo consultar la información oficial del juego en forma de listados y filtros. En este sentido, *RollAgain* no es un producto innovador.

Sin embargo, la gran mayoría se centran en listar información ya existente y no ofrecen al usuario la posibilidad de crear sus componentes personalizados (y mucho menos de formar grupos y poder compartir dichos componentes). Además, la posibilidad de crear entradas en la wiki de cada grupo añade al proyecto una funcionalidad de “Diario de juego” que otras aplicaciones no poseen.

A continuación se listan algunas de las alternativas ya existentes en el mercado:

- **Roll20¹**: Plataforma muy popular cuyo principal uso es el alojamiento completo de partidas en línea. Permite al usuario organizarse en grupos y cuenta con opciones de tablero muy completas.
- **D&DBeyond²**: Sitio oficial de *Dungeons & Dragons*, que ofrece un listado de las reglas del juego base. Además, cuenta con una herramienta de creación de personajes y otros aspectos de pago.
- **5e Companion App³**: Aplicación para *Android* e *iOS* que permite el acceso rápido a las reglas del juego. También permite llevar la cuenta de combates y de los estados de personajes y enemigos.

2.2. Fuentes de datos

Se ha recurrido al SRD (*Systems Reference Document*) de *Dragones y Mazmorras 5ª edición*, que consiste en un documento bajo licencia abierta que contiene toda la información básica del reglamento oficial del juego.

La aplicación no bebe directamente de este documento, sino de una API externa que recoge la información del documento y lo organiza en una base de datos: *dnd5eapi* [2].

Para mejorar las prestaciones y evitar información redundante, se ha realizado un volcado de los archivos JSON necesarios para este proyecto y se han importado los datos pertinentes a la base de datos local, sirviendo nosotros mismos la API.

2.3. Requisitos y Casos de Uso

2.3.1. Requisitos

Definiciones:

- **Componente oficial**: Cualquier elemento que forma parte del reglamento del juego base o de sus expansiones: reglas, hechizos, objetos, criaturas... (Similar a la información que se encontraría en un libro de reglas oficial).
- **Componente personalizado**: Cualquier elemento de juego que no pertenece al reglamento oficial, y en su lugar ha sido creado por los integrantes de un grupo.
- **Ficha de personaje**: Documento que guarda la información principal del personaje de un jugador: características, atributos, hechizos...

¹ <https://roll20.net/>

² <https://dndbeyond.net>

³ <https://play.google.com/store/apps/details?id=com.blastervla.ddencountergenerator>

- **Grupo:** Unidad de juego que puede constar de 1 a 8 integrantes. Cada uno consta de un código de invitación que se podrá usar para que nuevos usuarios se unan. El creador de un grupo será considerado el *Master*.
- **Máster:** Jugador que, a diferencia del resto, no toma el papel de un personaje, sino que prepara las sesiones de juego y guía al resto de jugadores a través de la historia. En esta aplicación el máster de un grupo es su creador.

Requisitos funcionales:

- **1. Login y registro**

- **1.a:** El usuario podrá registrarse mediante un correo, un nombre de usuario y una contraseña.
- **1.b:** El usuario podrá acceder al contenido de la aplicación identificándose mediante su correo y contraseña.

- **2. Grupos**

- **2.a:** El usuario podrá crear un grupo o unirse a un grupo existente mediante un código único.
- **2.b:** El usuario podrá acceder a una pantalla donde crear su propia ficha de personaje y poder descargarla en formato PDF.
- **2.c:** El usuario podrá acceder a la información de otros usuarios del grupo y sus fichas de personaje.
- **2.d:** El usuario podrá acceder a una pantalla de creación de componentes, donde podrá elaborar sus componentes personalizados que otros miembros del grupo podrán consultar.
- **2.e:** El usuario podrá visualizar y editar los componentes personalizados del grupo.
- **2.f:** El usuario podrá acceder a una pantalla donde realizar tiradas de dados visibles por el resto del grupo.

- **3. Componentes**

- **3.a:** El usuario podrá acceder a un listado de componentes del juego.
- **3.b:** El usuario podrá ordenar los componentes según sus parámetros, además de poder utilizar un buscador.
- **3.c:** El usuario podrá asignar componentes a sus fichas de personaje, para poder acceder fácilmente a ellos sin la necesidad de buscarlos cada vez que los necesite consultar.

- **4. Wiki**

- **4.a:** Cada grupo dispondrá de una wiki en la que los usuarios podrán consultar información personalizada relativa a sus grupos o campañas.
- **4.b:** Los usuarios de un grupo podrán añadir nuevas entradas personalizadas a la wiki o editar las que ya existen.

- **5. Máster**

- **5.a:** El máster de un grupo tendrá acceso a una pantalla de diario, similar a la wiki pero que el resto de miembros del grupo no podrán ver, para planificar sesiones de juego futuras.
- **5.b:** El máster de un grupo podrá añadir nuevas notas a su diario o editar la que ya existen.

2.3.2. Casos de uso

En la siguiente figura se muestran los principales casos de uso de la aplicación:

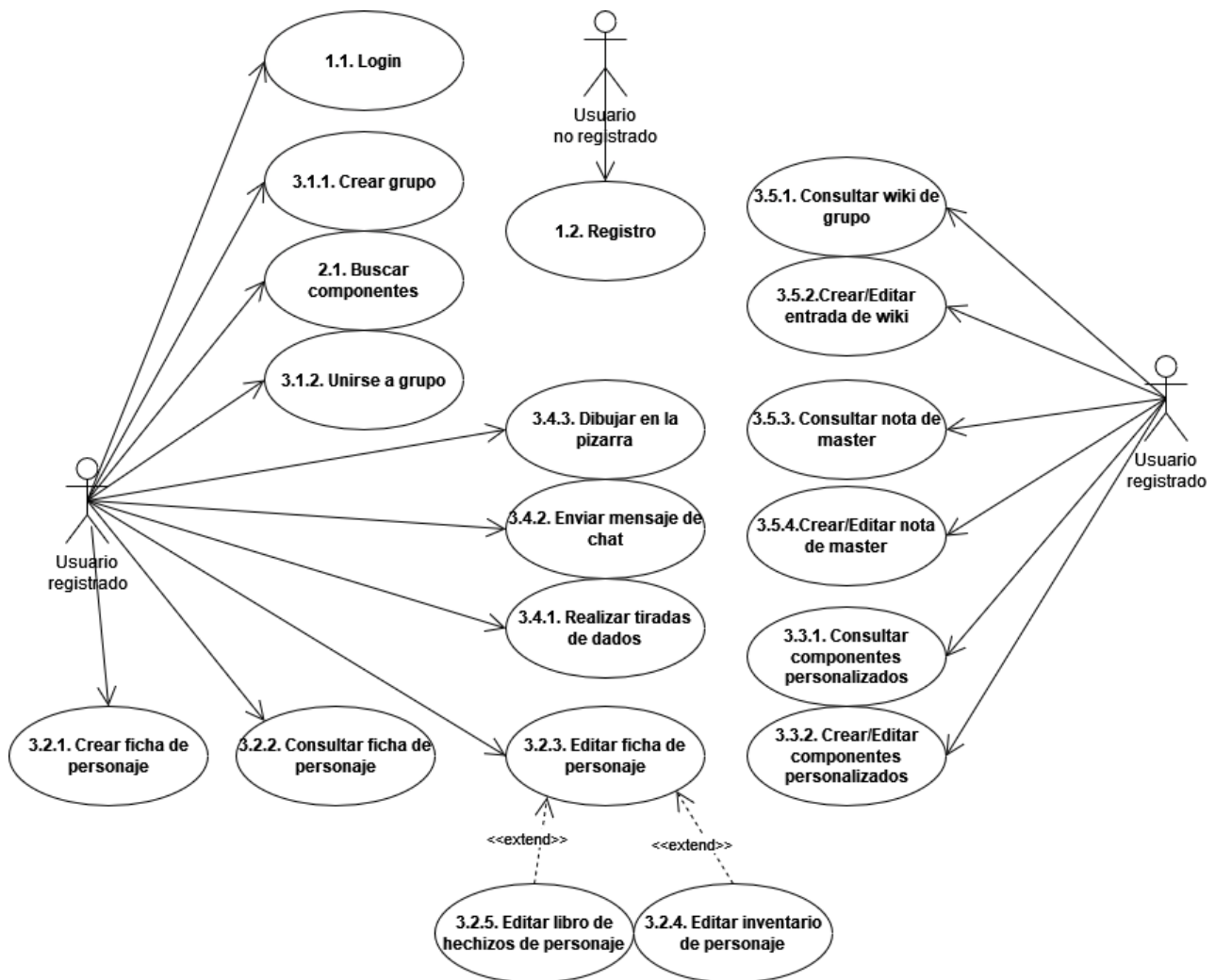


Figura 2.1: Diagrama de Casos de Uso.

Estos casos de uso se detallan uno a uno en el [Anexo I](#).

2.4. Interfaces de usuario

Durante la fase de diseño inicial, se realizó un esquema general que mostrase las distintas pantallas de la aplicación y las navegaciones entre ellas. Posteriormente se usarían estos bocetos para implementar la GUI en la aplicación final (siendo fiel a los elementos y navegación de estos pero con un nuevo diseño).

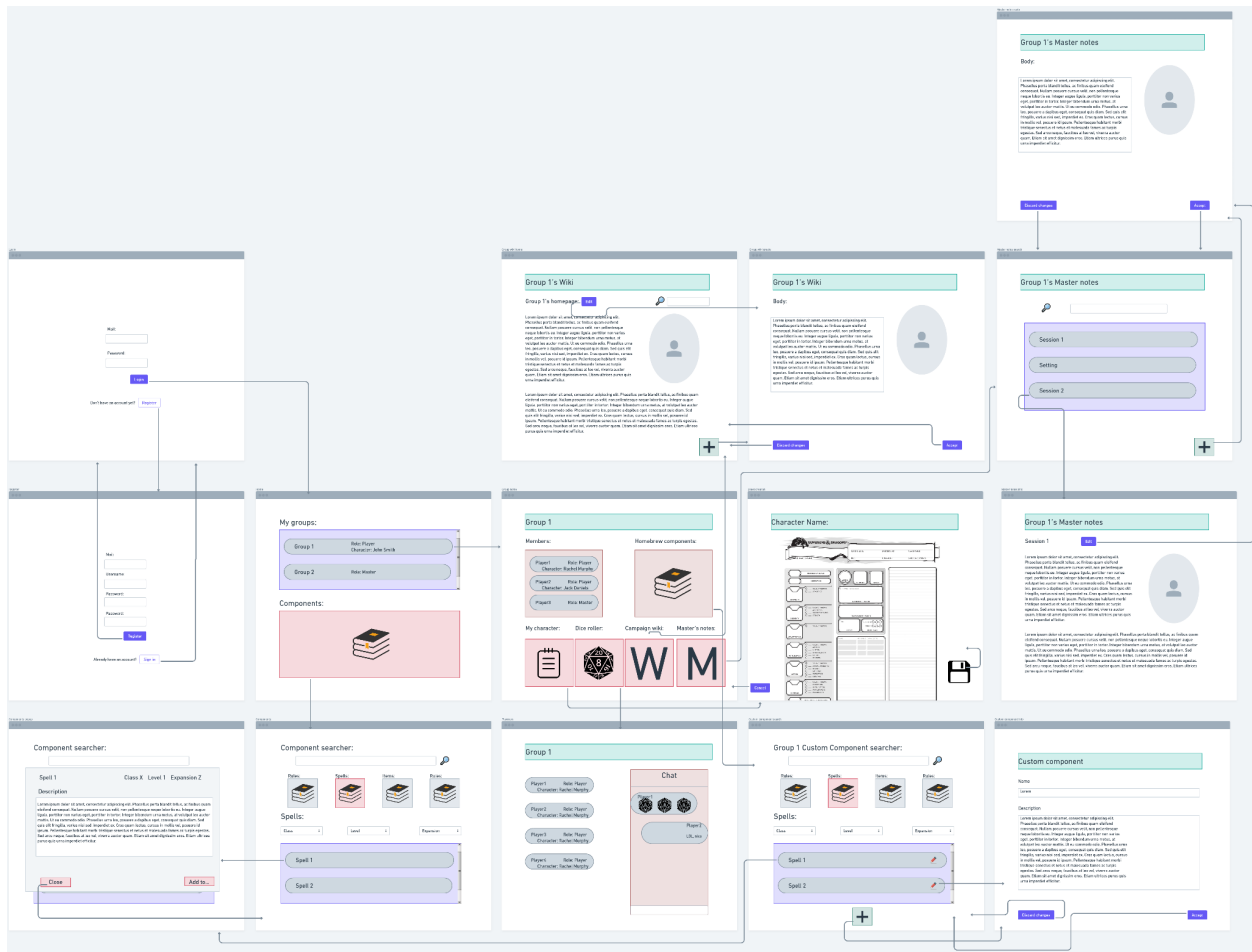


Figura 2.2: Boceto de Navegación Inicial

También se puede consultar este esquema de forma interactiva en el siguiente enlace:

<https://whimsical.com/tfg-RRFg566ST7Vtci4htrUTju>

2.4.1. Relación interfaz con requisitos

En el [Anexo II](#) se describen uno a uno los bocetos de la GUI y se relacionan con los casos de uso enumerados en la sección 2.3.2.

3. Diseño de la solución

En esta sección se explicarán las características más destacables del diseño del sistema, describiendo la arquitectura y tecnologías usadas. Además, se detallarán los detalles de implementación más destacables.

3.1. Arquitectura

Se ha escogido una arquitectura *3-tier*, distinguiéndose 3 partes principales: El cliente, el servidor y la base de datos. De esta forma, se establece una comunicación bi-direccional entre el cliente y el servidor Web, y entre el servidor Web y la base de datos.

El proyecto se ha desarrollado en el stack **MEAN** [3], que como base de datos utiliza **MongoDB** [4], un sistema no-SQL orientado a documentos; **Express** [5] como framework del servidor web, facilitando el desarrollo de la API; **Angular** [6] para el desarrollo del cliente web y **Node** [7] como entorno de ejecución del servidor. De esta forma, la inmensa mayoría de las líneas de código de la aplicación se han escrito en *JavaScript/Typescript*.

La base de datos se ha desplegado mediante **MongoDB Atlas** [8], que facilita el hosting gratuito en la nube. Sin embargo, dadas las limitaciones de espacio de esta solución, se ha utilizado también un almacenamiento *cloud* en **Firebase** [9], una plataforma integrada con la nube de *Google* que ofrece varias herramientas multiplataforma, para guardar tipos de datos más pesados (en este caso, las imágenes) mediante el servicio *Firebase Cloud Storage*.

3.1.1. Modelo de datos

La base de datos almacenará 7 entidades esenciales para el sistema:

- Los **usuarios**, que contendrán la información de identificación de cada persona que use el sistema.
- Los **grupos**, que agruparán usuarios y contendrán referencias a componentes personalizados, personajes y entradas de la wiki.
- Los tres tipos de componentes (**hechizos**, **equipamiento** y **rasgos**), que podrán ser oficiales o personalizados.
- Los **personajes**, que contendrán información variada y referencias a componentes.
- Las entradas de **wiki**, que cada grupo podrá crear. Las notas de *Master* son una especialización de este tipo que solo el *Master* de cada grupo podrá ver.

A continuación se muestra un diagrama de clases en el que se detallan los atributos y las relaciones de los datos:

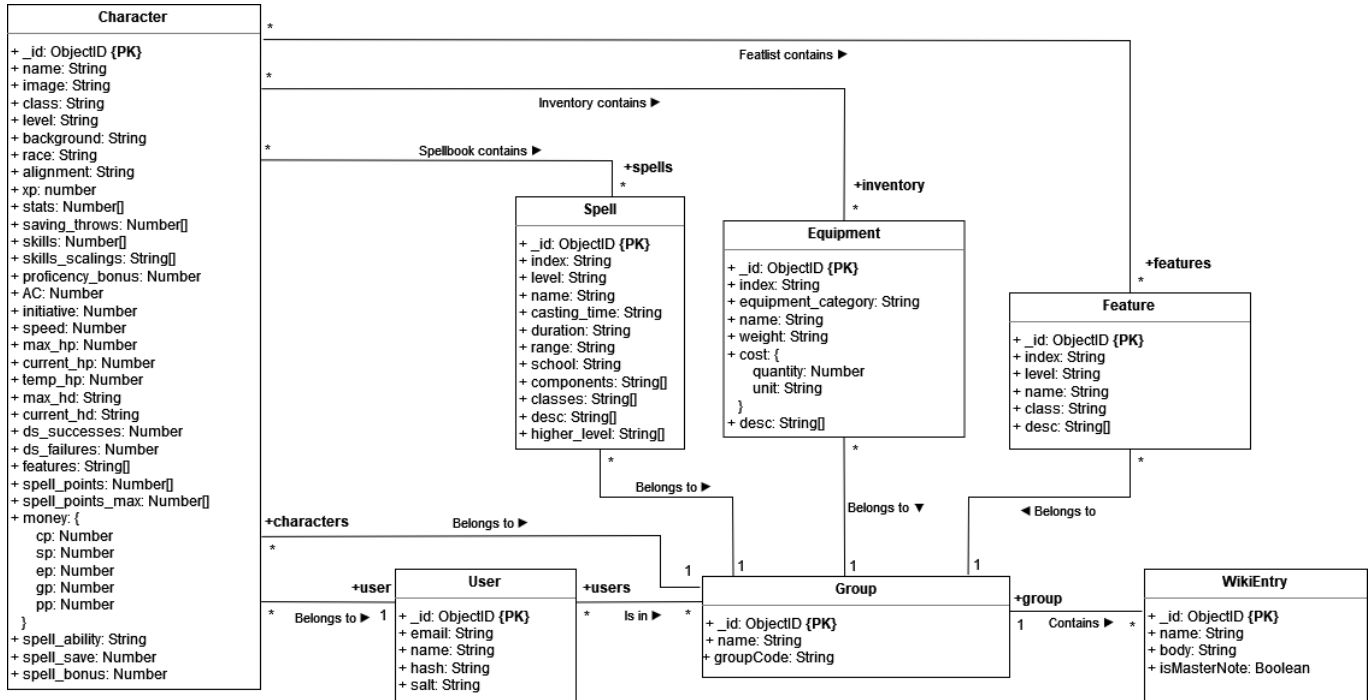


Figura 3.2: Diagrama de Clases del modelo de datos

Se han simplificado los atributos de los personajes para facilitar la visibilidad, pues las fichas de personaje almacenan gran cantidad de cadenas de texto y valores numéricos.

3.1.2. Paquetes y clases

La siguiente figura representa los paquetes más relevantes que componen el back-end de la aplicación:

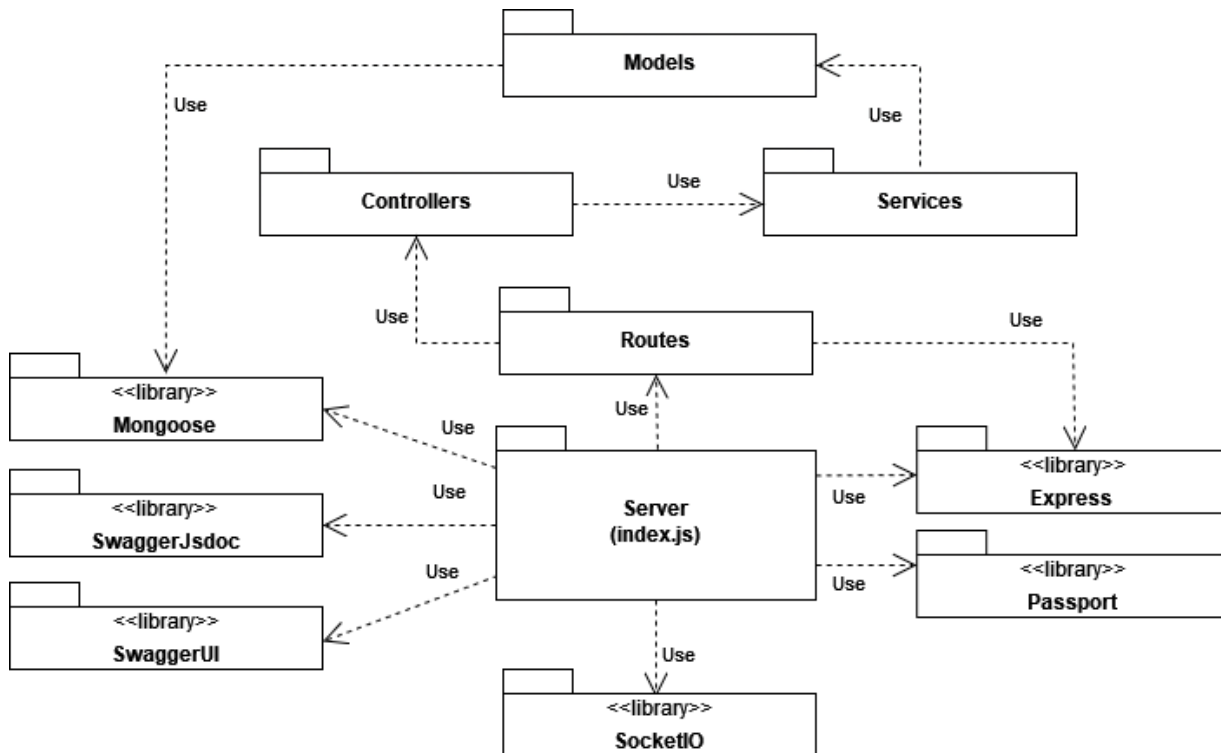


Figura 3.3: Diagrama de Paquetes del back-end

Como se puede observar en el diagrama, todos los paquetes del back-end se organizan en torno al servidor (servido en el fichero *index.js*). Esta clase hace uso de todas las interfaces de la API REST (las clases de la API se detallarán más adelante) y de las dependencias necesarias. Entre los paquetes utilizados más relevantes se encuentran:

- **Mongoose** [10], que adapta los métodos CRUD de la base de datos a un lenguaje relacional más amigable.
- **Socket.io** [11], que permite trabajar con WebSockets para escuchar eventos procedentes de unos clientes y redirigirlos a otros.
- **Swagger** [15] (*SwaggerUI* y *SwaggerJsdoc*), que permite servir un listado de las llamadas en una ruta de la web, y acompañarlas de descripciones y ejemplos.
- **Passport** [18], que actúa como plataforma de autenticación y permite generar tokens de sesión para garantizar el acceso seguro a la aplicación.

Se ha seguido un patrón Modelo-Vista-Controlador (*MVC*), que separa los componentes según su funcionalidad (representar información, aplicar la lógica necesaria o servirla al cliente). De esta forma, cada entidad se corresponde con 4 módulos:

- Un fichero de enrutado, que actúa como API REST, en el que se definen todas las llamadas que acepta el servidor por el protocolo *HTTPS*.
- Un fichero *controller*, que procesa las peticiones y se encargará de llamar a los correspondientes servicios con los parámetros adecuados.

- Un fichero *service*, que contendrá la lógica del negocio y las funciones encargadas de acceder a la base de datos.
- Un fichero *model*, en el que se define el *schema* de MongoDB con todos los atributos relativos a una entidad.

A continuación se muestra un diagrama de clases completo del servidor, incluyendo las funciones de las rutas, controladores y servicios para cada entidad.

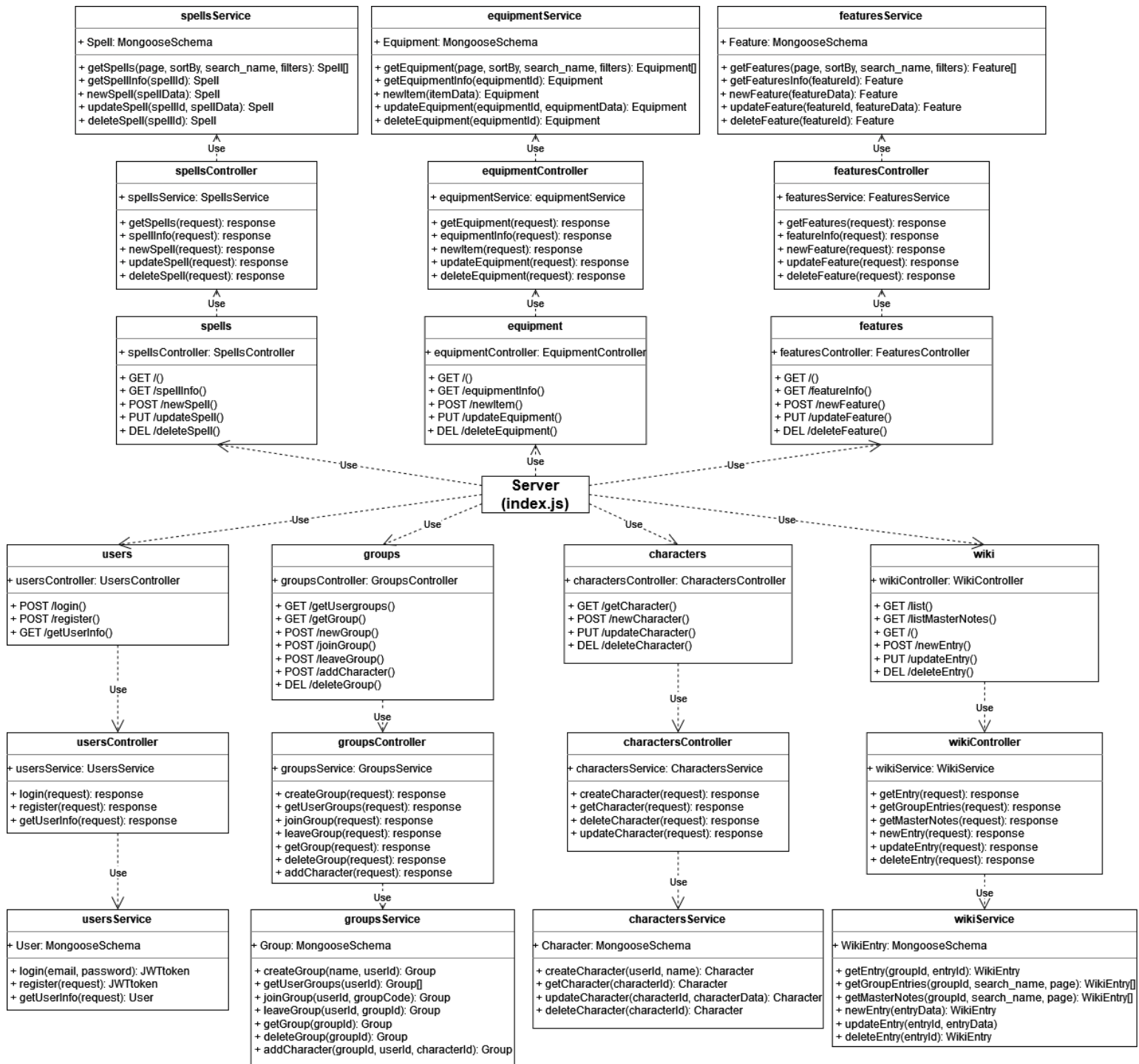


Figura 3.4: Diagrama de Clases del back-end

Respecto al front-end, las unidades funcionales básicas serán los *Componentes* (refiriéndonos al tipo 'Component' de Angular), que podrán ser páginas en su totalidad o parte de ellas. Los

Componentes exportados bajo una ruta en el componente *Routing* serán los que representen una página, y podrán usar a su vez otros *Componentes*.

A su vez, algunos de estos *Componentes* pondrán en marcha *Servicios* que interactuarán con el back-end (WebSockets) y otros servicios externos (la base de datos de imágenes). Cada componente estará formado por un fichero html con el contenido, un fichero css con los estilos y un archivo *TypeScript* con toda la lógica necesaria.

Para elaborar la interfaz de usuario se ha utilizado la librería **Bootstrap** [19], un framework que facilita un uso sencillo de hojas de estilos con muchas clases de css personalizables.

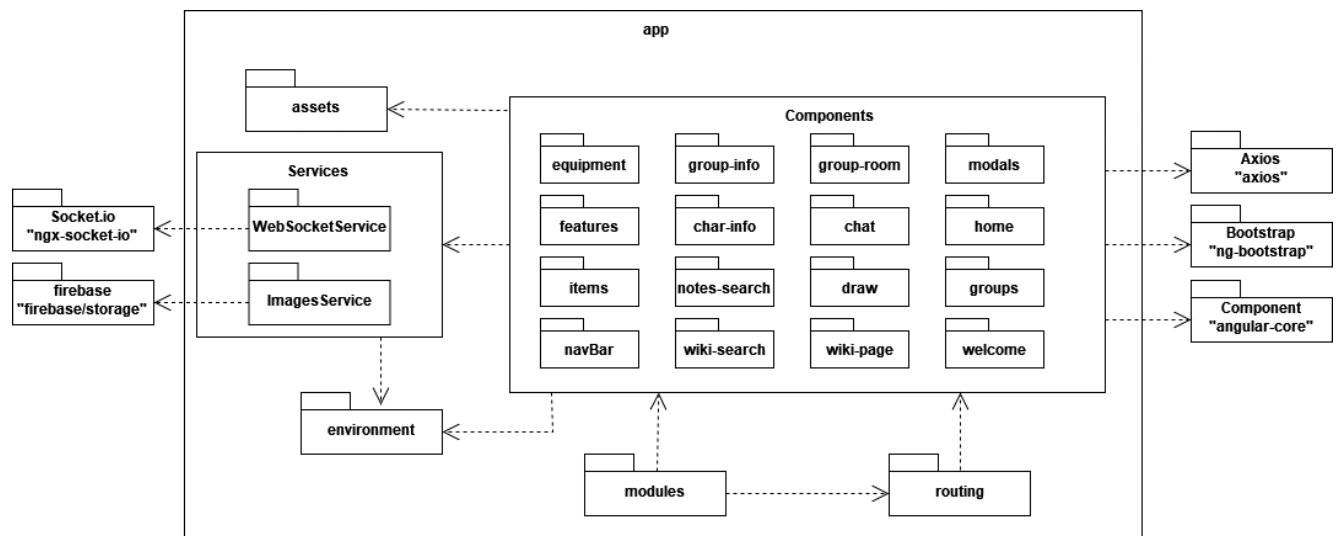


Figura 3.5: Diagrama de Paquetes Front-end

3.1.3. Componentes y conectores

El diagrama de la Figura 3.6 muestra los 3 tiers de la aplicación: front-end (azul), back-end (amarillo) y bases de datos (rojo).

Los *Componentes* del front-end enviarán peticiones HTTPS al servidor Web, que, una vez reconocidas por un componente de rutas (habrá uno por cada recurso expuesto en la API REST) será procesado por los *Controladores*. En función de los resultados de este procesamiento, el controlador devolverá un código HTTP distinto. Los *Controladores* harán uso de los *Servicios* para gestionar la lógica de dominio necesaria. Además, estos servicios tendrán acceso a los modelos de datos y podrán ejecutar operaciones CRUD sobre ellos (mediante el ODM *Mongoose*).

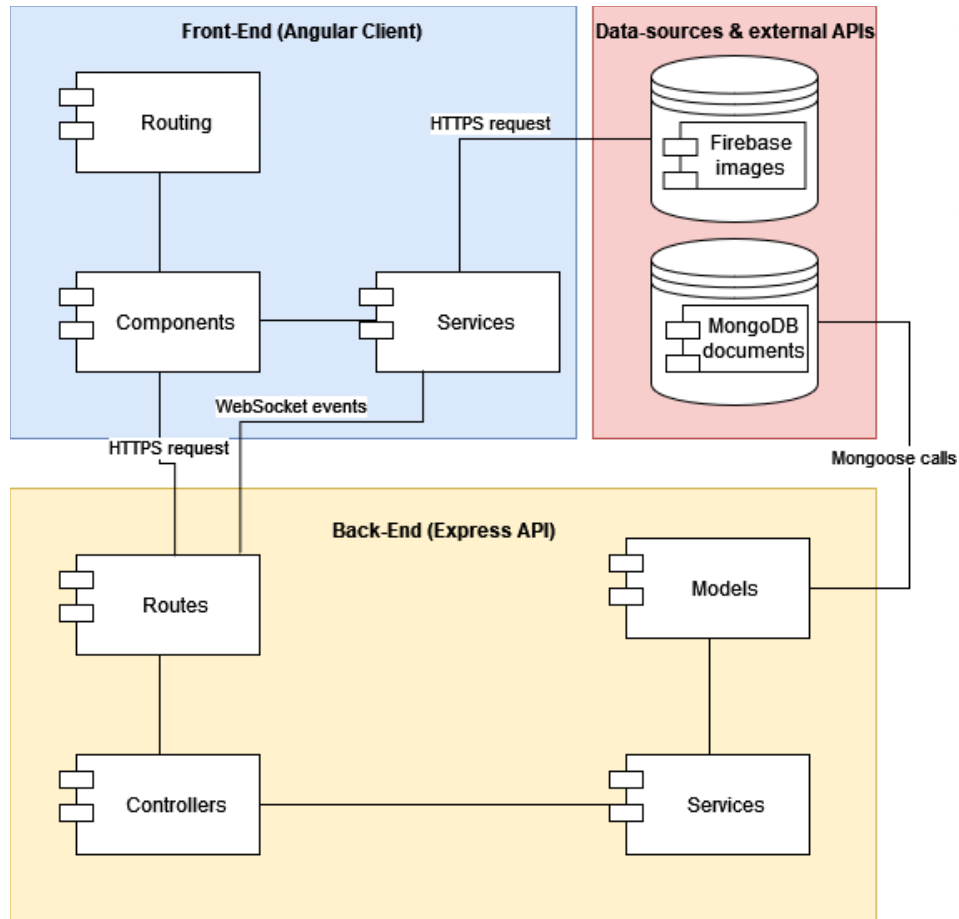


Figura 3.6: Diagrama de Componentes del Sistema

En el diagrama de la Figura 3.7 se puede observar un ejemplo de interacción entre un cliente y el servidor web que ilustra el patrón típico de interacción entre los tipos de componentes mostrados en la Figura 3.6:

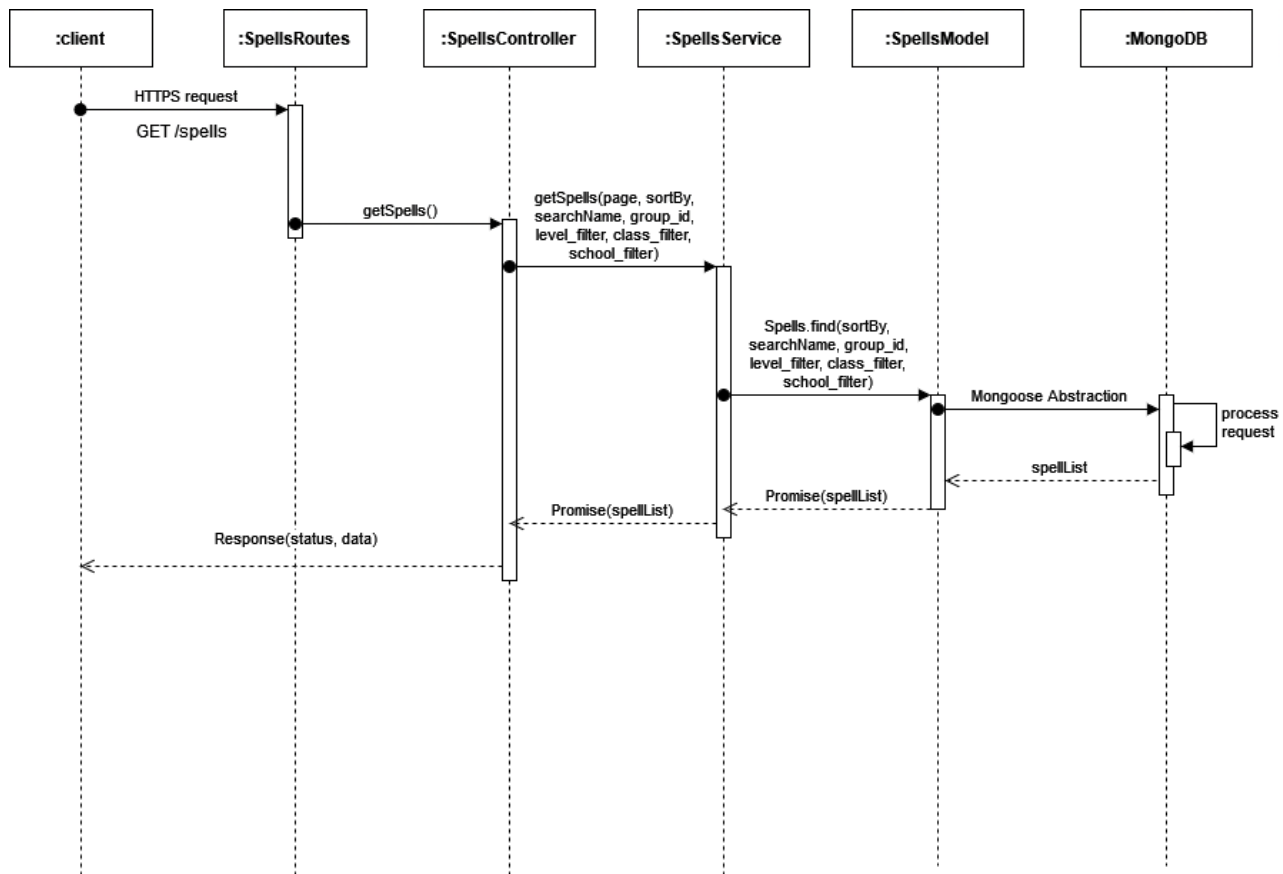


Figura 3.7: Un ejemplo de diagrama de secuencia del Servidor: obtener listado de hechizos

El cliente realiza una llamada HTTPS al punto `/spells`, que debería devolver una lista de los hechizos presentes en la base de datos (con los filtros que se desee). La API REST procesa la llamada mediante el controlador `SpellsController`, que devolverá el código de estado y la carga de datos correspondientes al cliente. Para obtener estos datos, el controlador hace uso del servicio `SpellsService` que, en función de los parámetros de la llamada, invocará los métodos correspondientes sobre el esquema del tipo de dato `Spell`, mediante el ODM `Mongoose`.

3.1.4. Despliegue

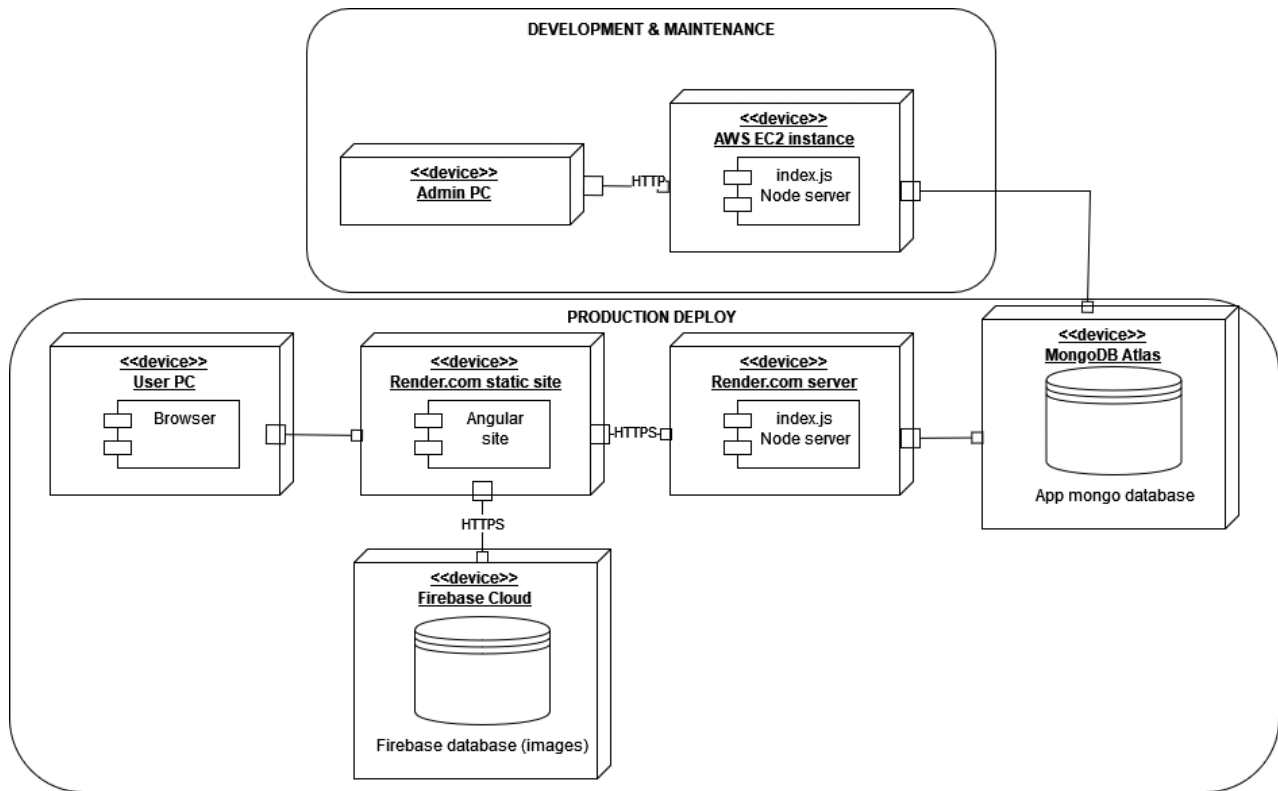


Figura 3.8: Diagrama de Despliegue del Sistema

Como se ha mencionado en el resumen arquitectural, el front y el back se han desplegado con la herramienta *render*. De esta forma, al acceder a la aplicación, el navegador del usuario final cargará el sitio estático correspondiente al front-end.

La versión de producción del servidor Web desplegado atenderá peticiones HTTPS únicamente del dominio establecido para el cliente web, asegurando un acceso seguro a la API. Además, se desplegará un segundo servidor Web en una máquina de Amazon AWS que permite el acceso manual a la API.

Además, se ha configurado una instancia elástica de **AWS EC2** [13], en la que se ha puesto en marcha una copia del servidor Web. La dirección de esta máquina será sólo accesible con las IPs añadidas manualmente. De esta forma, se podrán hacer llamadas directas a la API para tareas de mantenimiento o testing. Esta máquina habrá sido configurada de tal manera que solo acepte paquetes de entrada de las IPs de la red del administrador, y su propósito será llevar a cabo pruebas durante la fase de desarrollo y realizar tareas de mantenimiento.

El despliegue del cliente y el servidor Web se ha realizado mediante la herramienta *render* [12], que ofrece estancias gratuitas de hosting y garantiza una dirección DNS y un certificado SSL para poder establecer una comunicación por protocolo HTTPS.

Para desplegar la API secundaria en AWS se ha utilizado **GitHub Actions** [14], una herramienta de *GitHub* que nos permite ejecutar scripts (archivos *.yml*) en máquinas remotas de forma automática (en este caso, cada vez que se actualiza el repositorio).

La base de datos se desplegará sobre MongoDB Atlas, bajo un servicio gratuito de hosting cloud. Será accesible mediante una clave única, disponible para el servidor Web y para el administrador del sistema. Además, se desplegará una base de datos Firebase que se usará únicamente para almacenar las imágenes de los usuarios, y será accesible directamente mediante llamadas HTTPS por el cliente Web.

3.2. Implementación

En el apartado anterior se ha explicado la arquitectura del sistema y se ha hablado de los componentes que lo forman. Sin embargo, no se ha profundizado en cómo funcionan dichos componentes, por lo que a continuación se explicarán los detalles más relevantes.

Llamadas HTTP

En el caso del front, se ha mencionado que cada componente tiene un fichero en *TypeScript* que contiene toda la lógica de una pantalla del cliente Web. Esto incluye los controladores de los distintos elementos HTML de la página y las correspondientes llamadas a la API del back-end. Dichas llamadas se han realizado utilizando el paquete *Axios*, que facilita la formulación de las peticiones y la gestión de las respuestas de las llamadas HTTP.

Se barajó incluir todas las llamadas a la API bajo funciones en ficheros externos a los componentes. Sin embargo, dado que el tamaño de estos componentes no es muy grande, se optó finalmente por incluir las llamadas HTTP en la lógica de sus respectivos componentes, puesto que no dificultan la lectura del código.

Uso de WebSockets

Se ha utilizado la librería *Socket.io* para gestionar todas las tareas de comunicación bi-direccional que ofrece la aplicación. Todas estas tareas corresponden a la pantalla de Sala de Juego, que contiene un canvas de dibujo y un chat.

Se ha configurado el back-end para que acepte eventos de varios tipos (dibujar, limpiar canvas, enviar mensaje, unirse a la sala...), de tal forma que al recibir uno de estos eventos del cliente, el servidor envía un nuevo evento a los clientes que deban de recibirlo. Para ello, cuando un usuario entra en una sala de juego, el servidor añade su ID de dispositivo a un grupo de sockets, sobre el cual se retransmitirán los eventos que emita.

En el front-end se ha creado un servicio *web-socket.service.ts* que se inyecta en el cliente cuando carga el componente de sala de juego. En este fichero se declaran una serie de objetos *emitters* junto a sus correspondientes funciones de escucha y emisión, que se encargan de gestionar los distintos tipos de eventos disponibles.

```
outEven: EventEmitter<any> = new EventEmitter();
drawCallback: EventEmitter<any> = new EventEmitter();
eraseCallback: EventEmitter<any> = new EventEmitter();
chatMsgCallback: EventEmitter<any> = new EventEmitter();
newConnectionCallback: EventEmitter<any> = new EventEmitter();
userDisconnectCallback: EventEmitter<any> = new EventEmitter();
updateUserListCallback: EventEmitter<any> = new EventEmitter();
```

Figura 3.2.1: Fragmento Declaración de Eventos WebSockets

Almacenamiento de imágenes

Para almacenar las imágenes de las fichas de personaje, se ha creado un repositorio en *Firebase*. Esta plataforma, mediante un proyecto de tipo *Storage*, permite usar una base de datos en la nube que permite almacenar ficheros de cualquier tipo.

A diferencia de la base de datos de la aplicación (MongoDB), es el front-end el que realiza directamente las llamadas al servidor de *Firebase* cuando se sube una nueva imagen. Al procesar la petición, el servidor *Firebase* devuelve una URL que contiene la imagen. Esta URL es la que se almacena en la base de datos MongoDB, en su respectiva ficha de personaje.

Para ello, se ha creado un servicio *storage.service.ts* en el front-end al que se llama al subir una nueva foto. Este servicio se encarga de realizar las llamadas HTTP necesarias a la instancia de *Firebase* para subir la imagen y obtener el enlace, mediante el paquete *firebase* habilitado para *Angular*.

Integración y despliegue continuos

Como se ha mencionado, se ha utilizado la herramienta *render* para el despliegue del front y del back de producción. Este sitio cuenta con la opción de asociar un repositorio de GitHub a cada proyecto, para lanzar las acciones que se deseen con cada commit en la rama que se especifique.

Se ha configurado de tal forma que se hace un *build* y *run* del back y del front cuando se actualiza la rama *main* de los repositorios. De esta forma, el front y el back están continuamente funcionando en las siguientes URLs:

<https://rollagain-api.onrender.com>
<https://rollagain.onrender.com>

En el caso del despliegue del *backup* de la API en AWS, se ha configurado un workflow mediante un fichero "Push-to-EC2.yml" en GitHub Actions, que se lanza cada vez que se realice un commit sobre la rama *main* del backend. Este workflow se encarga de comprobar la validez del código y realizar un *pull* del nuevo código en la instancia de EC2. Finalmente se instalan los nuevos paquetes si fuera necesario y la nueva versión se pone en marcha.

```
steps:
  - name: Checkout the code
    uses: actions/checkout@v1

  - name: Deploy to my EC2 instance
    uses: easingthemes/ssh-deploy@v2.1.5
    env:
      SSH_PRIVATE_KEY: "${{ secrets.EC2_SSH_KEY }}"
      SOURCE: "./"
      REMOTE_HOST: "ec2-13-38-98-85.eu-west-3.compute.amazonaws.com"
      REMOTE_USER: "ubuntu"
      TARGET: "/home/ubuntu/TFG_BACK"
  - name: App npm install
    run: npm install
```

Figura 3.2.2: Pasos Despliegue GitHub Actions

Se ha utilizado el paquete *pm2* de ubuntu para mantener el *daemon* del servidor Web en funcionamiento aunque se cierre la consola de AWS.

Documentación de la API

Para documentar la API se ha utilizado el paquete *Swagger*, que permite generar una ruta del back-end en la que poder consultar información sobre las llamadas disponibles.

La información presente en esta página de documentación se especifica como un comentario previo a cada una de las funciones del fichero de rutas de cada tipo de datos.

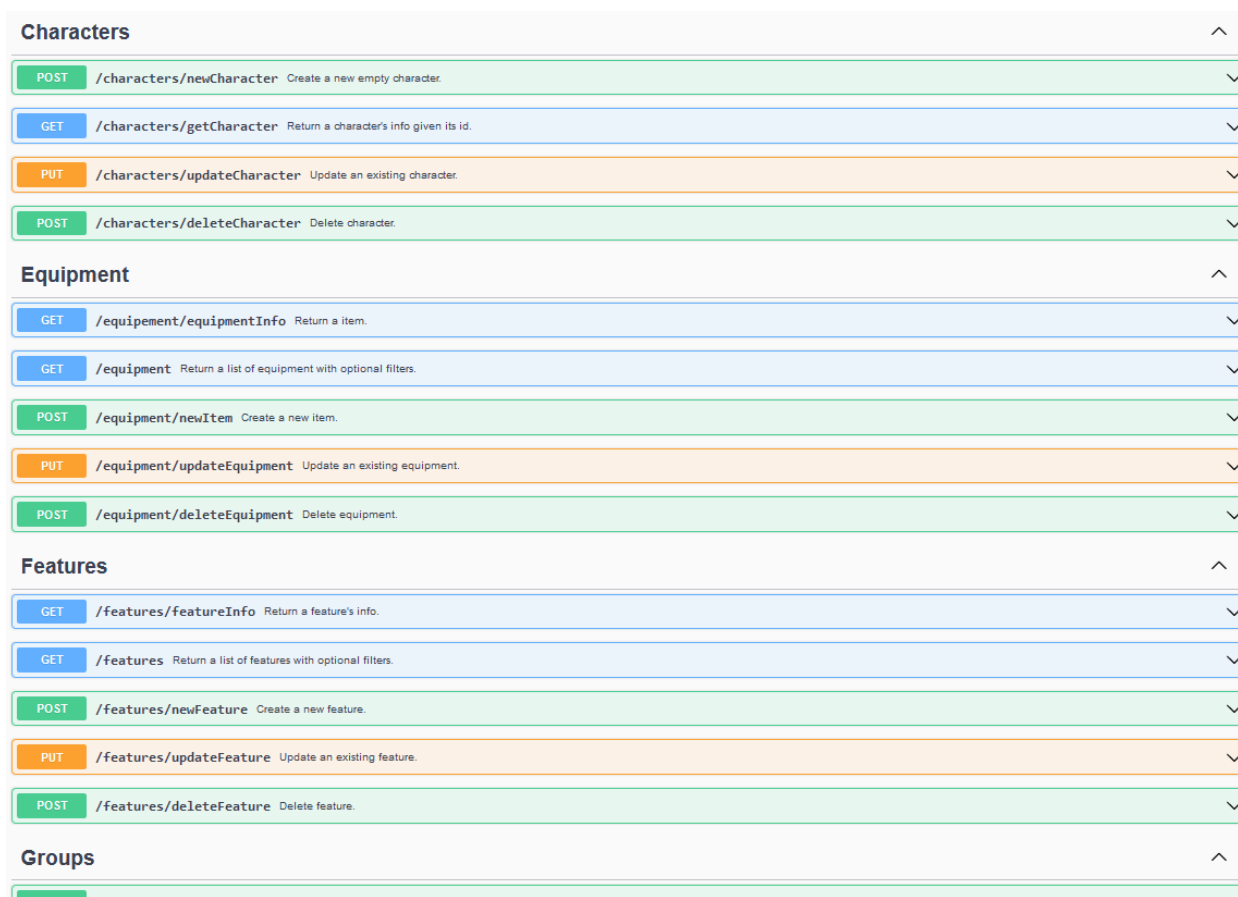


Figura 3.2.3: Pantallazo de Documentación en Swagger

Esta documentación es accesible en el siguiente enlace:

<https://rollagain-api.onrender.com/api-docs/>

3.2.1. Funcionalidades finales

La GUI final de la aplicación se ha basado en la propuesta en la fase de análisis (sección 2.2), incluyendo las vistas clave y los elementos de navegación pertinentes. Sin embargo, el diseño se ha ido modificando a lo largo del desarrollo, distribuyendo los elementos funcionales de la forma más cómoda para el usuario final.

El [Anexo III](#) comprende el manual de usuario, incluyendo capturas de la GUI final de la aplicación.

3.3. Pruebas

En esta sección se describirán las pruebas automáticas que se han elaborado para garantizar la calidad del usuario y comprobar los límites del sistema.

3.3.1. Pruebas de extremo a extremo

Para asegurar la calidad de la experiencia de usuario, se han elaborado una serie de tests extremo a extremo (e2e) que ejecutan los casos de uso listados en la sección 2.1.1. Mediante estos tests queremos comprobar que todos los flujos de acciones de estos casos de uso se desarrollan correctamente y que el front-end y el back-end se comunican de forma correcta. Para ello, se ha utilizado la herramienta **Cypress** [16], ejecutando un robot que interactúa con los distintos elementos de la aplicación, simulando a un humano.

Además, durante la fase de desarrollo se han realizado tests manuales de las funcionalidades implementadas antes de dar el código como bueno y subirlo a la rama principal del proyecto. En la última fase de trabajo se ha puesto especial énfasis en este procedimiento, llevando al límite los casos de uso y forzando situaciones poco comunes en la navegación por el sitio Web. Además, se ha comprobado que todo el contenido de las pantallas sea debidamente visible independientemente de la resolución del dispositivo.

3.3.2. Pruebas de carga

Se ha automatizado un procedimiento para testear los límites de carga del sistema, simulando una interacción incremental con el servidor Web de la aplicación. Para ello, se ha utilizado la herramienta **Apache jMeter** [17], que consiste en un cliente que permite especificar una serie de llamadas HTTP que serán enviadas al servidor por un número indicado de clientes web.

Se ha configurado de tal manera que se simulará el uso de la aplicación con un número incremental de usuarios, empezando en 1 y añadiendo uno nuevo cada medio segundo. Cada usuario lanzará una petición cada dos segundos, hasta que la prueba concluya.

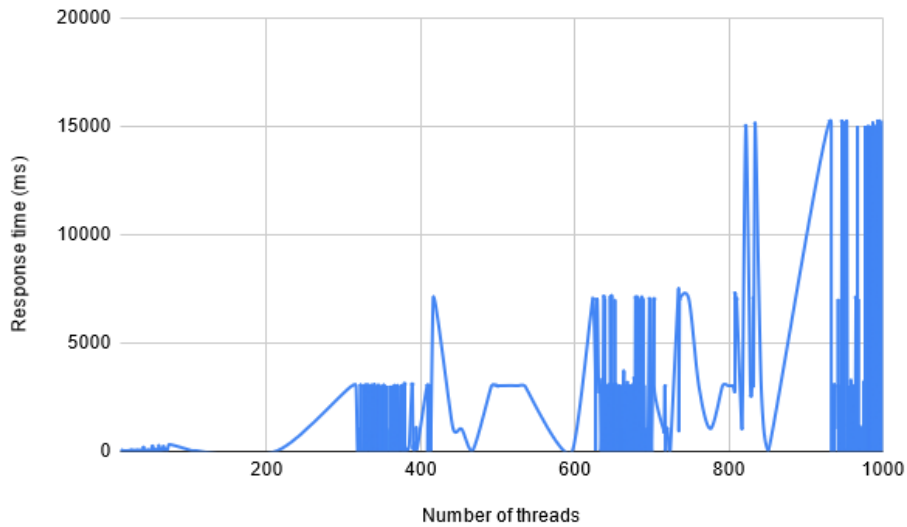


Figura 3.3.1: Gráfica de Sobrecarga con una instancia

En los resultados del test se observa que con un número reducido de usuarios (1 a 300), el tiempo de respuesta es de una media de 40 ms. Sin embargo, al llegar a los 316 usuarios concurrentes, se empiezan a observar valores de respuesta de 3 segundos. Conforme seguimos aumentando los clientes este valor va aumentando más y más, llegando a haber peticiones con más de 15 segundos de tiempo de respuesta con entre 800 y 1000 usuarios.

Si se quisiera aumentar la capacidad de usuarios concurrentes del sistema se podrían desplegar varias instancias del servidor bajo un clúster. Configurando las opciones de escalabilidad en el panel de control de *render*, se ha establecido un balanceador de carga para 4 instancias paralelas del servidor.

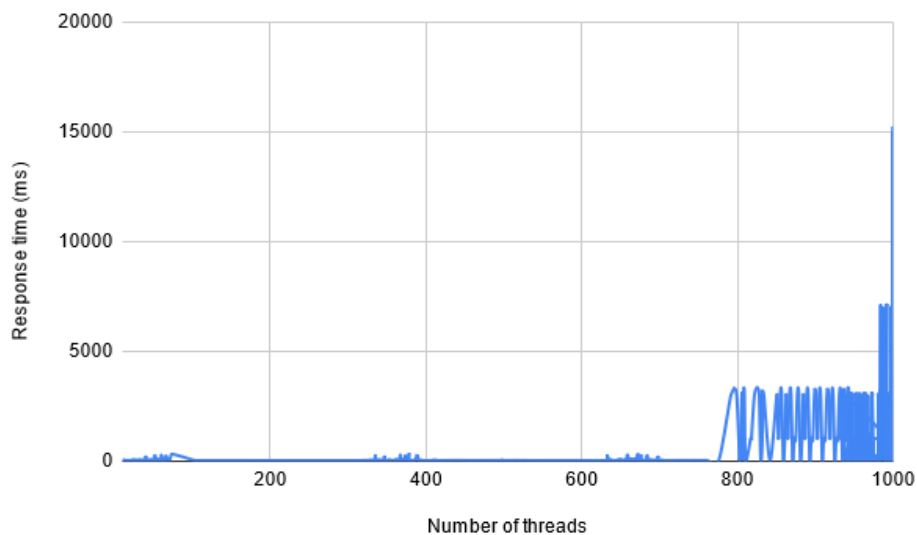


Figura 3.3.2: Gráfica de Sobrecarga con 4 instancias concurrentes

Se puede observar que con 4 clusters el rendimiento es notablemente mejor, observando el primer tiempo pesado (3100 ms) al superar los 780 usuarios concurrentes. El número total de instancias bajo el clúster se podría aumentar conforme a la cantidad de usuarios mensuales que visiten el sitio, o bien automatizarse configurando un sistema de auto-escalado.

Como conclusión, se ha comprobado que el sistema es escalable horizontalmente y que la capacidad de servicio de la API es proporcional al número de instancias atendiendo peticiones simultáneamente.

4. Gestión del proyecto

El proyecto se ha dividido en 3 fases principales: *planificación, desarrollo y redacción y pruebas*.

- La fase de **planificación** englobó todas las tareas previas a la puesta en marcha de las funcionalidades de la aplicación. Esto incluye la elaboración de la propuesta del proyecto, la puesta en contacto con el tutor del trabajo y la redacción de los debidos requisitos, casos de uso, diagramas y bocetos. Además, en esta fase se crearon los correspondientes repositorios de GitHub y se configuraron los mecanismos de CI/CD pertinentes.
- La fase de **desarrollo** comprende el global de la elaboración de todas las funcionalidades de la aplicación, de tal forma que esta fase terminaría cuando la aplicación propuesta fuese completamente usable. Esto incluye el desarrollo completo de la API y del cliente Web, y la configuración y mantenimiento del resto de elementos del sistema.
- La última fase del proyecto se ha enfocado en realizar **tests**, tanto manuales como automáticos, para garantizar la calidad de las funcionalidades ofrecidas por la aplicación. Esta fase corresponde a las últimas semanas de trabajo y, dado que las pruebas han permitido encontrar errores en la aplicación, se han prolongado tareas de desarrollo hasta esta fase. Además, en la fase final también se ha dedicado gran parte del tiempo a la **redacción de la memoria** (aunque esta se ha ido trabajando a lo largo de las dos primeras fases). Cabe destacar que otras tareas de documentación como la descripción de las llamadas de la API se realizó durante el desarrollo.

Respecto a la metodología usada, inicialmente se planteó usar sistemas establecidos como Scrum o Kanban; sin embargo, se descartaron por dos motivos principales. El primero es que estas metodologías están orientadas principalmente al reparto de tareas en equipos.

El segundo motivo es el lapso de tiempo en el que se ha desarrollado el proyecto, puesto que, pese a que el tiempo entre el comienzo y el final de proyecto ha sido amplio (+6 meses), la frecuencia de trabajo ha variado al no ser un trabajo de tiempo completo.

Al final se han utilizado algunos procedimientos correspondientes a Scrum, como la división de cada tarea en tareas lo más pequeñas posibles y la asignación de puntos de historia, agrupando todas ellas en una pila. Sin embargo, esta pila no ha sido utilizada tanto para llevar una gestión continua del proyecto, sino como una forma de obtener una visión global de los límites del proyecto. Por lo tanto, tras establecer las funcionalidades principales, la pila quedó en desuso.

De esta forma, se ha usado un modelo de desarrollo iterativo en el que se han ido poniendo en producción versiones utilizables de la aplicación, a la que se han ido incorporando nuevas funcionalidades completas en cada iteración.

Al final de cada sesión de trabajo, se han ido anotando los tiempos y las tareas realizadas en una ficha de *Excel*, para saber cuánto tiempo se ha dedicado en cada frente del proyecto. El tiempo total dedicado ha sido de 320 horas, distribuidas de la siguiente manera:

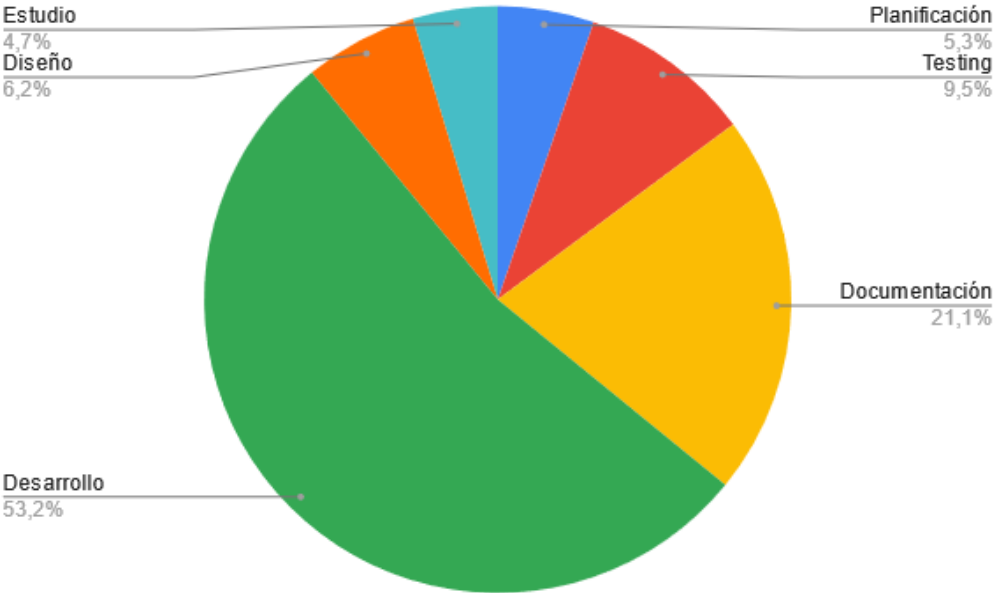


Figura 4.1. Gráfico Circular de Reparto de Esfuerzos

5. Conclusiones y trabajo futuro

Este apartado se ha elaborado tras finalizar la mayor parte del trabajo, e intenta recopilar las conclusiones finales y las impresiones que se han obtenido durante el transcurso de este. Además, se incluirán algunas de las ideas consideradas como posibles nuevas funcionalidades pero que no estaban dentro del alcance del proyecto.

5.1. Conclusiones

Se ha cumplido con los objetivos propuestos inicialmente. Se ha desarrollado una aplicación web usando tecnologías del *stack MEAN*, que cumple con los requisitos elaborados en la fase de diseño.

El sistema desarrollado permite consultar las reglas básicas del juego fácilmente. Además, los usuarios se pueden juntar en grupos y crear sus propias reglas que podrán compartir entre ellos. Cada usuario puede crear sus propios personajes y editar sus características, algunas de las cuales estarán conectadas con otros elementos del juego, permitiendo a los usuarios crear accesos rápidos a otros componentes sin necesitar cerrar sus hojas de personaje.

Se ha integrado una tecnología de comunicación bidireccional para crear un chat en vivo con el que los integrantes de un grupo podrán comunicarse si fuera necesario y realizar tiradas de dados.

Además, se han realizado las pruebas pertinentes para garantizar el correcto funcionamiento del sistema, así como tests de volumen que nos han permitido determinar que la aplicación es escalable horizontalmente.

Cabe destacar que ya se había utilizado el *stack* de tecnologías en proyectos grupales previos, facilitando la labor de aprendizaje inicial. No obstante, ha sido necesario el aprendizaje de nuevas tecnologías no utilizadas previamente o que no habían sido exploradas tan a fondo en otros proyectos, como por ejemplo:

- La manipulación avanzada de algunos elementos gráficos de *Angular*.
- La elaboración de suites de tests de estrés automáticos mediante *jMeter*.
- La configuración de un sistema de despliegue automático en Amazon AWS.
- La integración de un repositorio de almacenamiento cloud como *Firebase*.
- La elaboración de una documentación para proyectos de este calibre.

5.2. Plan de futuro

Pese a que el resultado de este trabajo se ha considerado suficiente con los objetivos propuestos en la fase de planificación y diseño, a lo largo del desarrollo han surgido ideas de funcionalidades

que aportarían aspectos interesantes a la aplicación, pero se han descartado por no entrar en los márgenes de este trabajo.

Sin embargo, si se continuase el desarrollo de la aplicación, se intentarían añadir algunos de los siguientes elementos:

- Más elementos para facilitar la labor del *Master* de un grupo, como una lista de posibles enemigos (los cuales podría añadir mediante accesos rápidos a sus notas de sesión) o un planificador de encuentros de combate en el que llevar un conteo del estado de los personajes de los jugadores y los enemigos.
- Ampliar el carácter social de la aplicación, permitiendo a los usuarios editar sus perfiles y añadir amigos a una lista, desde la que puedan contactar rápidamente con otros usuarios e invitarlos a grupos de juego.
- Permitir a los grupos compartir de forma pública sus componentes personalizados, permitiendo a otros grupos explorar nuevos componentes que añadir a sus sesiones de juego.
- Un tablero interactivo en la sala de juego, que permitiese a los usuarios colocar y mover sus propias fichas, facilitando al grupo prescindir de elementos físicos y permitiéndoles desarrollar una partida de forma virtual.
- Siguiendo la posibilidad de partidas virtuales, se podría añadir un chat de voz/vídeo que facilitase el transcurso de partidas remotas.

Estos dos últimos elementos darían a la aplicación la calidad de “Plataforma de juego” más que “Asistente”, pues podría sustituir completamente el uso de cualquier otro dispositivo físico o electrónico. Sin embargo, se seguiría pudiendo usar la aplicación para consultar los componentes deseados y organizar las fichas de personajes que se usarán en sesiones no virtuales.

Además, se podría ampliar la aplicación e importar nuevas fuentes de datos que contemplen el uso de otros sistemas de juego de rol similares, como *Pathfinder*⁴, *Cyberpunk*⁵ o *Call of Cthulhu*⁶. Sin embargo, es posible que fuera más sencillo y cómodo para el usuario final disponer de estas funcionalidades en plataformas separadas, pues el contenido de las normas y las fichas de personaje podrían variar demasiado.

⁴ <https://paizo.com/pathfinder>

⁵ <https://rtalsoriangames.com/cyberpunk/>

⁶ <https://www.chaosium.com/call-of-cthulhu-rpg/>

Bibliografía

- [1] *Sitio oficial de Dragones y Mazmorras*, Wizards of the Coast. <https://dnd.wizards.com/es>, accedida el 17 de enero de 2022
- [2] *API d&d 5ª edición usada*, 5e bits. <http://www.dnd5eapi.co/>, accedida el 5 de enero de 2022
- [3] *Definición del stack MEAN*, IBM. <https://www.ibm.com/topics/mean-stack>, accedida el 16 de mayo de 2022
- [4] *Documentación de MongoDB*, MongoDB Inc. <https://www.mongodb.com/docs/>, accedida el 5 de enero de 2022
- [5] *Documentación Express*, OpenJS Foundation. <https://expressjs.com/en/4x/api.html>, accedida el 7 de enero de 2022
- [6] *Documentación Angular*, Google. <https://angular.io/docs>, accedida el 5 de enero de 2022
- [7] *Documentación Node*, OpenJS Foundation. <https://nodejs.org/en/docs>, accedida el 5 de enero de 2022
- [8] *MongoDB Atlas*, MongoDB Inc. <https://www.mongodb.com/atlas>, accedida el 5 de enero de 2022
- [9] *Documentación Firebase*, Google. <https://firebase.google.com/docs?hl=es-419>, accedida el 30 de abril de 2022
- [10] *Documentación Mongoose*, Automattic. <https://mongoosejs.com/docs>, accedida el 5 de enero de 2022
- [11] *Documentación socket.io*, Socket.IO. <https://socket.io/docs/v4/>, accedida el 7 de marzo de 2022
- [12] *Documentación render*. <https://render.com/docs>, accedida el 16 de mayo de 2022
- [13] *Sitio AWS EC2*, Amazon. <https://aws.amazon.com/es/ec2/>, accedida el 13 de abril de 2022
- [14] *Documentación GitHub Actions*, GitHub Inc. <https://docs.github.com/en/actions>, accedida el 13 de abril de 2022
- [15] *Documentación Swagger*, SmartBear Software. <https://swagger.io/solutions/api-documentation/>, accedida el 14 de enero de 2022
- [16] *Documentación Cypress*, Cypress.io. <https://docs.cypress.io>, accedida el 6 de mayo de 2022
- [17] *Manual Apache jMeter*, Apache Software Foundation. <https://jmeter.apache.org/usermanual/index.html>, accedida el 15 de mayo de 2022
- [18] *Documentación Passport*, Jared Hanson. <https://www.passportjs.org/docs/>, accedida el 22 de mayo de 2022
- [19] *Documentación Bootstrap*, Bootstrap Team. <https://getbootstrap.com/docs/5.3>, accedida el 22 de mayo de 2022

Lista de Figuras

- [Figura 2.1: Diagrama general de Casos de Uso.](#)
- [Figura 2.2: Boceto de Navegación Inicial](#)
- [Figura 3.1: Esquema de Tecnologías Usadas](#)
- [Figura 3.2: Diagrama de Clases del modelo de datos](#)
- [Figura 3.3: Diagrama de Paquetes del Back-end](#)
- [Figura 3.4: Diagrama de Clases API](#)
- [Figura 3.5: Diagrama de Paquetes Front-end](#)
- [Figura 3.6: Diagrama de Componentes del Sistema](#)
- [Figura 3.7: Ejemplo diagrama de secuencia del Servidor](#)
- [Figura 3.8: Diagrama de Despliegue del Sistema](#)
- [Figura 3.2.1: Fragmento Declaración de Eventos WebSockets](#)
- [Figura 3.2.2: Pasos Despliegue GitHub Actions](#)
- [Figura 3.2.3: Pantallazo de Documentación en Swagger](#)
- [Figura 3.3.1: Gráfica de Sobrecarga con una instancia](#)
- [Figura 3.3.2: Gráfica de Sobrecarga con 4 instancias concurrentes](#)
- [Figura 4.1. Gráfico Circular de Reparto de Esfuerzos](#)
- [Figura B.1: Boceto Pantalla de Login](#)
- [Figura B.2: Boceto Pantalla de Registro](#)
- [Figura B.3: Boceto Pantalla de Inicio](#)
- [Figura B.4: Boceto Pantalla de Búsqueda de Componentes](#)
- [Figura B.5: Boceto Pantalla de Información de Componente](#)
- [Figura B.6: Boceto Pantalla de Información de Grupo](#)
- [Figura B.7: Boceto Pantalla de Ficha de Personaje](#)
- [Figura B.8: Boceto Pantalla de Sala de Grupo](#)
- [Figura B.9: Boceto Pantalla de Búsqueda de Componentes Personalizados](#)
- [Figura B.10: Boceto Pantalla de Creación de Componentes Personalizados](#)
- [Figura B.11: Boceto Pantalla de Entrada de Wiki](#)
- [Figura B.12: Boceto Pantalla de Creación de Entrada de Wiki](#)
- [Figura B.13: Boceto Pantalla de Búsqueda de Notas de Master](#)
- [Figura B.14: Boceto Pantalla de Nota de Master](#)
- [Figura B.15: Boceto Pantalla de Creación de Nota de Master](#)
- [Figura C.1: Pantalla Final de Login](#)
- [Figura C.2: Pantalla Final de Registro](#)
- [Figura C.3: Notificación Errores de Registro](#)
- [Figura C.3: Notificación Errores de Registro](#)
- [Figura C.4: Pantalla Final de Inicio](#)
- [Figura C.5: Pantalla Final de Búsqueda de Componentes](#)
- [Figura C.6: Desplegable Información de Componente](#)
- [Figura C.7: Pantalla Final de Grupos](#)
- [Figura C.8: Pantalla Final de Información de Grupo](#)

- [Figura C.9: Pantalla Final de Ficha de Personaje](#)
- [Figura C.10: Pantalla Final de Edición de Ficha de Personaje](#)
- [Figura C.11: Desplegables Inventario y Libro de Hechizos de Personaje](#)
- [Figura C.11: Desplegables Inventario y Libro de Hechizos de Personaje](#)
- [Figura C.12: Desplegable Rasgos de Personaje](#)
- [Figura C.13: Desplegable Añadir Componente a Personaje](#)
- [Figura C.14: Pantalla Final de Búsqueda de Componentes Personalizados](#)
- [Figura C.15: Pantalla Final de Creación/Edición de Componentes](#)
- [Figura C.16: Pantalla Final de Sala de Juego](#)
- [Figura C.17: Desplegable Tiradas de Dados](#)
- [Figura C.18: Pantalla Final de Búsqueda de Entradas de Wiki](#)
- [Figura C.19: Pantalla Final de Entrada de Wiki](#)
- [Figura C.20: Pantalla Final de Edición de Entrada de Wiki](#)
- [Figura C.21: Pantalla Final de Búsqueda de Notas de Master](#)

Lista de Tablas

- [Tabla A.1: Descripción Caso de Uso de Login.](#)
- [Tabla A.2: Descripción Caso de Uso de Registro](#)
- [Tabla A.3: Descripción Caso de Uso de Búsqueda de Componentes](#)
- [Tabla A.4: Descripción Caso de Uso de Creación de Grupos](#)
- [Tabla A.5: Descripción Caso de Uso de Unión a Grupo](#)
- [Tabla A.6: Descripción Caso de Uso de Creación de Personajes](#)
- [Tabla A.7: Descripción Caso de Uso de Acceso a Ficha de Personaje](#)
- [Tabla A.8: Descripción Caso de Uso de Edición de Datos de Personaje](#)
- [Tabla A.9: Descripción Caso de Uso de Edición de Inventario de Personaje](#)
- [Tabla A.10: Descripción Caso de Uso de Edición de Hechizos de Personaje](#)
- [Tabla A.11: Descripción Caso de Uso de Búsqueda de Componentes Personalizados](#)
- [Tabla A.12: Descripción Caso de Uso de Creación de Componentes Personalizados](#)
- [Tabla A.13: Descripción Caso de Uso de Realizar Tiradas de Dados](#)
- [Tabla A.14: Descripción Caso de Uso de Enviar Mensajes de Chat](#)
- [Tabla A.15: Descripción Caso de Uso de Dibujar en el Tablero de Grupo](#)
- [Tabla A.16: Descripción Caso de Uso de Consulta de Wiki de Grupo](#)
- [Tabla A.17: Descripción Caso de Uso de Creación/Edición de wiki de Grupo](#)
- [Tabla A.18: Descripción Caso de Uso de Consulta de Notas de Master](#)
- [Tabla A.19: Descripción Caso de Uso de Creación de Notas de Master](#)

Anexo I - Casos de Uso

Este anexo contiene un listado detallado de los Casos de Uso de la aplicación.

1.1. Login		
Descripción	El usuario se intentará identificar mediante correo y contraseña y el sistema le otorgará acceso si los datos son correctos.	
Precondición	El usuario no está identificado en la web.	
Flujo básico	Paso	Acción
	1	El usuario accede a cualquier enlace de la página.
	2	El sistema redirige a la pantalla de login y solicita correo electrónico y contraseña al usuario.
	3	El usuario introduce un correo y contraseña válidos y hace click en "login".
	4	El sistema concede acceso al usuario y redirige a la pantalla de inicio.
Flujos alternativos	3.1	Si el usuario introduce un usuario y contraseña no válidos,
	3.1.1	El sistema informa del error.
Postcondición	El sistema almacena el token del usuario y le permite realizar acciones de usuario hasta que expire su sesión.	
Comentarios	-	

Tabla A.1: Descripción Caso de Uso de Login.

1.2. Registro		
Descripción	El usuario se intentará registrar mediante correo, nombre y contraseña y el sistema creará una cuenta de usuario.	
Precondición	El usuario se encuentra en la página de login.	
Flujo básico	Paso	Acción
	1	El usuario hace click en "register"
	2	El sistema solicita al usuario correo, nombre de usuario

		y contraseña.
	3	El usuario introduce los datos correctos y hace click en "register".
	4	El sistema informa del correcto registro y redirige a la pantalla de login con los campos completados con los datos del nuevo usuario.
Flujos alternativos	3.1	Si el usuario introduce incorrectamente algún campo.
	3.1.1	El sistema informa del error y permanece en la página de registro.
Postcondición	El nuevo usuario ha sido creado y se podrá acceder al sistema con sus credenciales.	
Comentarios	-	

Tabla A.2: Descripción Caso de Uso de Registro

2.1. Buscar componentes		
Descripción	El usuario intentará acceder a información detallada sobre componentes oficiales del juego.	
Precondición	El usuario está identificado y se encuentra en la página de <i>home</i> .	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>Spells, Equipment</i> o <i>Features</i> .
	2	El sistema muestra un listado de todos los componentes del tipo elegido.
	3	El usuario hace click en cualquier entrada.
	4	El sistema muestra un desplegable con la información detallada sobre el componente.
Flujos alternativos	2.1	El usuario busca un componente por su nombre usando el campo de texto.
	2.2	El usuario hace click en cualquiera de los parámetros de la cabecera de la tabla para ordenar según un criterio.
	2.3	El usuario hace click en las flechas "<<" y ">>" para avanzar y retroceder página.
Postcondición	-	

Comentarios	-
--------------------	---

Tabla A.3: Descripción Caso de Uso de Búsqueda de Componentes

3.1.1. Crear un grupo		
Descripción	El usuario intentará crear un nuevo grupo de juego.	
Precondición	El usuario está identificado y se encuentra en la página de <i>home</i> .	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>My groups</i> .
	2	El sistema muestra un listado de los grupos a los que pertenece el usuario.
	3	El usuario introduce un nombre en el campo de texto de <i>Group name</i> y hace click en <i>Create group</i>
	4	El sistema crea el nuevo grupo e informa al usuario.
Flujos alternativos	-	-
Postcondición	El grupo ha sido creado y se mostrará en la pestaña <i>My groups</i> para los usuarios que pertenezcan a él.	
Comentarios	-	

Tabla A.4: Descripción Caso de Uso de Creación de Grupos

3.1.2. Unirse a un grupo		
Descripción	El usuario intentará acceder a un grupo de juego ya existente.	
Precondición	El usuario está identificado y se encuentra en la página de <i>home</i> .	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>My groups</i> .
	2	El sistema muestra un listado de los grupos a los que pertenece el usuario.
	3	El usuario introduce un nombre en el campo de texto de <i>Group code</i> y hace click en <i>Join group</i>
	4	El sistema añade al usuario al grupo y le informa.

Flujos alternativos	3.1	Si el código de grupo no pertenece a ningún grupo o el usuario ya está en el grupo:
	3.1.1	El sistema informa al usuario del error.
Postcondición	El usuario se ha unido al grupo y podrá acceder a él desde la página <i>My groups</i> .	
Comentarios	-	

Tabla A.5: Descripción Caso de Uso de Unión a Grupo

3.2.1. Crear ficha de personaje		
Descripción	El usuario intentará crear una nueva ficha de personaje en un grupo al que pertenezca.	
Precondición	El usuario está identificado y se encuentra en la página de <i>home</i> . El usuario pertenece a algún grupo.	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>My groups</i> .
	2	El sistema muestra un listado de los grupos a los que pertenece el usuario.
	3	El usuario hace click sobre el grupo en el que desee crear su personaje.
	4	El sistema muestra la página de información del grupo.
	5	El usuario hace click en el botón <i>New</i> de la sección <i>My Characters</i> .
	6	El sistema solicita al usuario un nombre para el personaje.
	7	El usuario introduce el nombre deseado y hace click en <i>Create</i> .
	8	El sistema muestra el nuevo personaje en la lista.
Flujos alternativos	-	-
Postcondición	El nuevo personaje queda registrado en el sistema.	
Comentarios	-	

Tabla A.6: Descripción Caso de Uso de Creación de Personajes

3.2.2. Consultar ficha de personaje		
Descripción	El usuario intentará consultar una ficha de personaje suya o de algún integrante de un grupo al que pertenezca.	
Precondición	El usuario está identificado y se encuentra en la página de <i>home</i> . El usuario pertenece a algún grupo. El usuario posee un personaje en dicho grupo.	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>My groups</i> .
	2	El sistema muestra un listado de los grupos a los que pertenece el usuario.
	3	El usuario hace click sobre el grupo en el que desee crear su personaje.
	4	El sistema muestra la página de información del grupo.
	5	El usuario hace click en su personaje bajo la sección <i>My characters</i> .
	6	El sistema muestra la pantalla de ficha de personaje con toda la información relativa a este.
Flujos alternativos	6.1	El usuario hace click en <i>inventory</i> .
	6.1.1	El sistema muestra las monedas y objetos del personaje.
	6.1.2	El usuario hace click en un objeto de la lista.
	6.1.3	El sistema muestra una ventana emergente con información detallada del objeto.
	6.2	El usuario hace click en <i>spellbook</i> .
	6.2.1	El sistema muestra el listado de hechizos del personaje.
	6.2.2	El usuario hace click en un hechizo de la lista.
	6.2.3	El sistema muestra una ventana emergente con información detallada del hechizo.
Postcondición	-	
Comentarios	-	

Tabla A.7: Descripción Caso de Uso de Acceso a Ficha de Personaje

3.2.3. Editar ficha de personaje		
Descripción	El usuario intentará editar una ficha de personaje que le pertenezca.	
Precondición	El usuario está identificado y se encuentra en la página de <i>home</i> . El usuario pertenece a algún grupo. El usuario posee un personaje en dicho grupo.	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>My groups</i> .
	2	El sistema muestra un listado de los grupos a los que pertenece el usuario.
	3	El usuario hace click sobre el grupo en el que desee crear su personaje.
	4	El sistema muestra la página de información del grupo.
	5	El usuario hace click en su personaje bajo la sección <i>My characters</i> .
	6	El sistema muestra la pantalla de ficha de personaje con toda la información relativa a este.
	7	El usuario hace click en el icono del lápiz al lado del nombre del personaje.
	8	El sistema permite al usuario editar todos los campos de su ficha de personaje.
	9	El usuario edita los parámetros deseados y hace click en <i>Save changes</i> .
	10	El sistema sale del modo edición e informa al usuario de que los cambios se han guardado.
Flujos alternativos	9.1	Si el usuario hace click en <i>Discard changes</i> :
	9.1.1	El sistema deshace los cambios realizados e informa al usuario.
Postcondición	Los cambios realizados se han almacenado correctamente.	
Comentarios	-	

Tabla A.8: Descripción Caso de Uso de Edición de Datos de Personaje

3.2.4. Editar inventario de personaje		
Descripción	El usuario intentará añadir o eliminar objetos del inventario de una ficha de personaje que le pertenezca.	
Precondición	El usuario se encuentra en la página de ficha de personaje, en el modo editar.	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>Inventory</i> .
	2	El sistema muestra el inventario del personaje.
	3	El usuario hace click en el botón con el símbolo +.
	4	El sistema muestra una ventana emergente con el listado de objetos.
	5	El usuario hace click en un objeto.
	6	El sistema muestra un desplegable con información detallada sobre el objeto.
	7	El usuario hace click en el botón <i>Add item to inventory</i> .
	8	El sistema actualiza el inventario con el nuevo objeto.
Flujos alternativos	5.1	Si el usuario hace click en <i>Custom items</i> :
	5.1.1	El sistema mostrará únicamente los objetos personalizados del grupo al que pertenece el personaje.
	4.1	Si el usuario hace click en el icono de borrado al lado de un objeto:
	4.1.1	El objeto se eliminará del inventario del personaje
	4.2	El usuario modifica las monedas actuales a su gusto.
Postcondición	Los cambios realizados se han almacenado correctamente.	
Comentarios	-	

Tabla A.9: Descripción Caso de Uso de Edición de Inventario de Personaje

3.2.5. Editar libro de hechizos de personaje	
Descripción	El usuario podrá añadir o eliminar hechizos del libro de hechizos de una ficha de personaje que le pertenezca.

Precondición	El usuario se encuentra en la página de ficha de personaje, en el modo editar.	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>Spellbook</i> .
	2	El sistema muestra el libro de hechizos del personaje.
	3	El usuario hace click en el botón <i>Add Spell</i> .
	4	El sistema muestra una ventana emergente con el listado de hechizos.
	5	El usuario hace click en un hechizo.
	6	El sistema muestra un desplegable con información detallada sobre el hechizo.
	7	El usuario hace click en el botón <i>Add spell to character's spellbook</i> .
	8	El sistema actualiza el libro de hechizos con el nuevo hechizo.
Flujos alternativos	5.1	Si el usuario hace click en <i>Custom spells</i> :
	5.1.1	El sistema mostrará únicamente los hechizos personalizados del grupo al que pertenece el personaje.
	4.1	Si el usuario hace click en el icono de borrado al lado de un hechizo:
	4.1.1	El hechizo se eliminará del inventario del personaje
	4.2	El usuario modifica los puntos de hechizo máximos y actuales a su deseo.
Postcondición	Los cambios realizados se han almacenado correctamente.	
Comentarios	-	

Tabla A.10: Descripción Caso de Uso de Edición de Hechizos de Personaje

3.3.1. Consultar componentes personalizados	
Descripción	El usuario intentará consultar los componentes de juego personalizados de un grupo.
Precondición	El usuario se encuentra en la página de información de grupo.

Flujo básico	Paso	Acción
	1	El usuario hace click en <i>Custom spells</i> , <i>Custom items</i> o <i>Custom features</i> .
	2	El sistema muestra el listado de componentes (similar al caso de uso 3).
	3	El usuario hace click en un componente.
Flujos alternativos	2.1	El usuario busca un componente por su nombre usando el campo de texto.
	2.2	El usuario hace click en cualquiera de los parámetros de la cabecera de la tabla para ordenar según un criterio.
	2.3	El usuario hace click en las flechas "<<" y ">>" para avanzar y retroceder página.
Postcondición	-	
Comentarios	La página mostrada es la misma que en el caso de uso 3, pero el contenido consiste únicamente de los componentes personalizados del grupo.	

Tabla A.11: Descripción Caso de Uso de Búsqueda de Componentes Personalizados

3.3.2. Crear/editar componentes personalizados		
Descripción	El usuario intentará crear o editar componentes personalizados para un grupo.	
Precondición	El usuario se encuentra en la página de información de grupo.	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>Custom spells</i> , <i>Custom items</i> o <i>Custom features</i> .
	2	El sistema muestra el listado de componentes (similar al caso de uso 3).
	3	El usuario hace click en <i>New spell</i> o <i>New item</i> .
4	El sistema muestra un desplegable con campos editables.	

	5	El usuario introduce la información del nuevo componente y hace click en <i>Save</i> .
	6	El sistema actualiza el listado de componentes personalizados con el componente recién creado.
Flujos alternativos	3.1	El usuario puede, en su lugar, hacer click sobre el icono del lápiz en el desplegable de un componente para editar un componente ya existente.
	5.1	El usuario puede hacer click en <i>Cancel</i> para cancelar la creación/edición del componente.
Postcondición	Los cambios realizados se han almacenado correctamente.	
Comentarios	-	

Tabla A.12: Descripción Caso de Uso de Creación de Componentes Personalizados

3.4.1. Realizar tiradas de dados		
Descripción	El usuario intentará realizar tiradas de dados visibles para todos los usuarios que se encuentren en una sala de juego.	
Precondición	El usuario se encuentra en la página de información de grupo.	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>Playroom</i> .
	2	El sistema muestra la pantalla de la sala de juego.
	3	El usuario hace click en el icono del dado.
	4	El sistema muestra un desplegable con posibles tiradas realizables.
	5	El usuario hace click en la combinación de tirada (núm. de dados + núm. de caras) que desea realizar.
	6	El sistema muestra en el chat la suma descompuesta y el valor total de la tirada de dados.
Flujos alternativos	-	-
Postcondición	Los cambios realizados se han almacenado correctamente.	
Comentarios	-	

Tabla A.13: Descripción Caso de Uso de Realizar Tiradas de Dados

3.4.2. Enviar mensaje de chat		
Descripción	El usuario intentará enviar un mensaje de chat para todos los usuarios que se encuentren en una sala de juego.	
Precondición	El usuario se encuentra en la página de información de grupo.	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>Playroom</i> .
	2	El sistema muestra la pantalla de la sala de juego.
	3	El usuario escribe en la caja de texto del chat y posteriormente pulsa <i>ENTER</i> .
	4	El sistema muestra el nuevo mensaje al final de la ventana de chat, visible por el resto de usuarios en la sala.
Flujos alternativos	-	-
Postcondición	Todos los usuarios conectados a la sala recibirán el mismo mensaje que ha escrito el usuario.	
Comentarios	-	

Tabla A.14: Descripción Caso de Uso de Enviar Mensajes de Chat

3.4.3. Dibujar en la pizarra		
Descripción	El usuario intentará hacer un garabato en la pizarra para todos los usuarios que se encuentren en la sala de juego.	
Precondición	El usuario se encuentra en la página de información de grupo.	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>Playroom</i> .
	2	El sistema muestra la pantalla de la sala de juego.
	3	El usuario arrastra el puntero haciendo click por encima de la pizarra.
	4	El sistema traza una línea por donde pase el puntero, que será visible para el resto de usuarios.

Flujos alternativos	-	-
Postcondición	Todos los usuarios conectados a la sala verán los mismos trazos reflejados en sus pizarras.	
Comentarios	-	

Tabla A.15: Descripción Caso de Uso de Dibujar en el Tablero de Grupo

3.5.1. Consultar wiki de grupo		
Descripción	El usuario intentará consultar una entrada de la wiki personalizada de un grupo.	
Precondición	El usuario se encuentra en la página de información de grupo.	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>Wiki</i> .
	2	El sistema muestra el listado de entradas de la wiki del grupo.
	3	El usuario hace click en una de las entradas.
	4	El sistema muestra la información relativa a la entrada de la wiki.
Flujos alternativos	2.1	El usuario busca una entrada por su nombre usando el campo de texto.
	2.2	El usuario hace click en las flechas "<<" y ">>" para avanzar y retroceder página.
Postcondición	-	
Comentarios	-	

Tabla A.16: Descripción Caso de Uso de Consulta de Wiki de Grupo

3.5.2. Crear/Editar entrada en la wiki de grupo		
Descripción	El usuario intentará crear o editar una entrada en la wiki personalizada de un grupo.	
Precondición	El usuario se encuentra en el listado de entradas de la wiki de un grupo..	
Flujo básico	Paso	Acción

	1	El usuario hace click en <i>New entry</i> .
	2	El sistema solicita un nombre al usuario.
	3	El usuario introduce un nombre y hace click en <i>Create</i> .
	4	El sistema crea la entrada y redirige a la página de la entrada.
	5	El usuario hace click en el botón con dibujo de lápiz.
	6	El sistema permite al usuario editar el cuerpo de la entrada.
	7	El usuario introduce la información pertinente y hace click en <i>Save changes</i> .
	8	El sistema guarda los cambios, sale del modo edición e informa al usuario.
Flujos alternativos	7.1	En lugar de guardar los cambios, el usuario hace click en <i>Discard changes</i> .
	7.2	El sistema descarta los cambios realizados y sale del modo edición.
Postcondición	La entrada se ha guardado correctamente en el sistema.	
Comentarios	-	

Tabla A.17: Descripción Caso de Uso de Creación/Edición de wiki de Grupo

3.5.3. Consultar notas de diario de <i>Master</i>		
Descripción	El usuario intentará consultar las notas de diario de un grupo del cual sea <i>Master</i> .	
Precondición	El usuario se encuentra en la página de información de grupo del cual es <i>Master</i> .	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>Master notes</i> .
	2	El sistema muestra el listado de notas.
	3	El usuario hace click en una de las nota.
	4	El sistema muestra la información relativa a la nota.

Flujos alternativos	2.1	El usuario busca una nota por su nombre usando el campo de texto.
	2.2	El usuario hace click en las flechas “<<” y “>>” para avanzar y retroceder página.
Postcondición	-	
Comentarios	-	

Tabla A.18: Descripción Caso de Uso de Consulta de Notas de Master

3.5.4. Crear/Editar nota de diario de Master		
Descripción	El usuario intentará crear o editar una nota de diario de un grupo del cual sea <i>Master</i> .	
Precondición	El usuario se encuentra en el listado de notas de <i>Master</i> del grupo (del cual es <i>Master</i>).	
Flujo básico	Paso	Acción
	1	El usuario hace click en <i>New entry</i> .
	2	El sistema solicita un nombre al usuario.
	3	El usuario introduce un nombre y hace click en <i>Create</i> .
	4	El sistema crea la nota y redirige a la página de la entrada.
	5	El usuario hace click en el botón con dibujo de lápiz.
	6	El sistema permite al usuario editar el cuerpo de la nota.
	7	El usuario introduce la información pertinente y hace click en <i>Save changes</i> .
	8	El sistema guarda los cambios, sale del modo edición e informa al usuario.
Flujos alternativos	7.1	En lugar de guardar los cambios, el usuario hace click en <i>Discard changes</i> .
	7.2	El sistema descarta los cambios realizados y sale del modo edición.
Postcondición	La nota se ha guardado correctamente en el sistema.	
Comentarios	-	

Tabla A.19: Descripción Caso de Uso de Creación de Notas de Master

Anexo II - Bocetos de la GUI

A continuación se muestran los distintos bocetos iniciales de la interfaz de usuario de la aplicación (realizados durante la fase de análisis y diseño). Además, se acompañará cada boceto con una descripción y un listado de los requisitos con los que se relaciona.

Pantalla de login

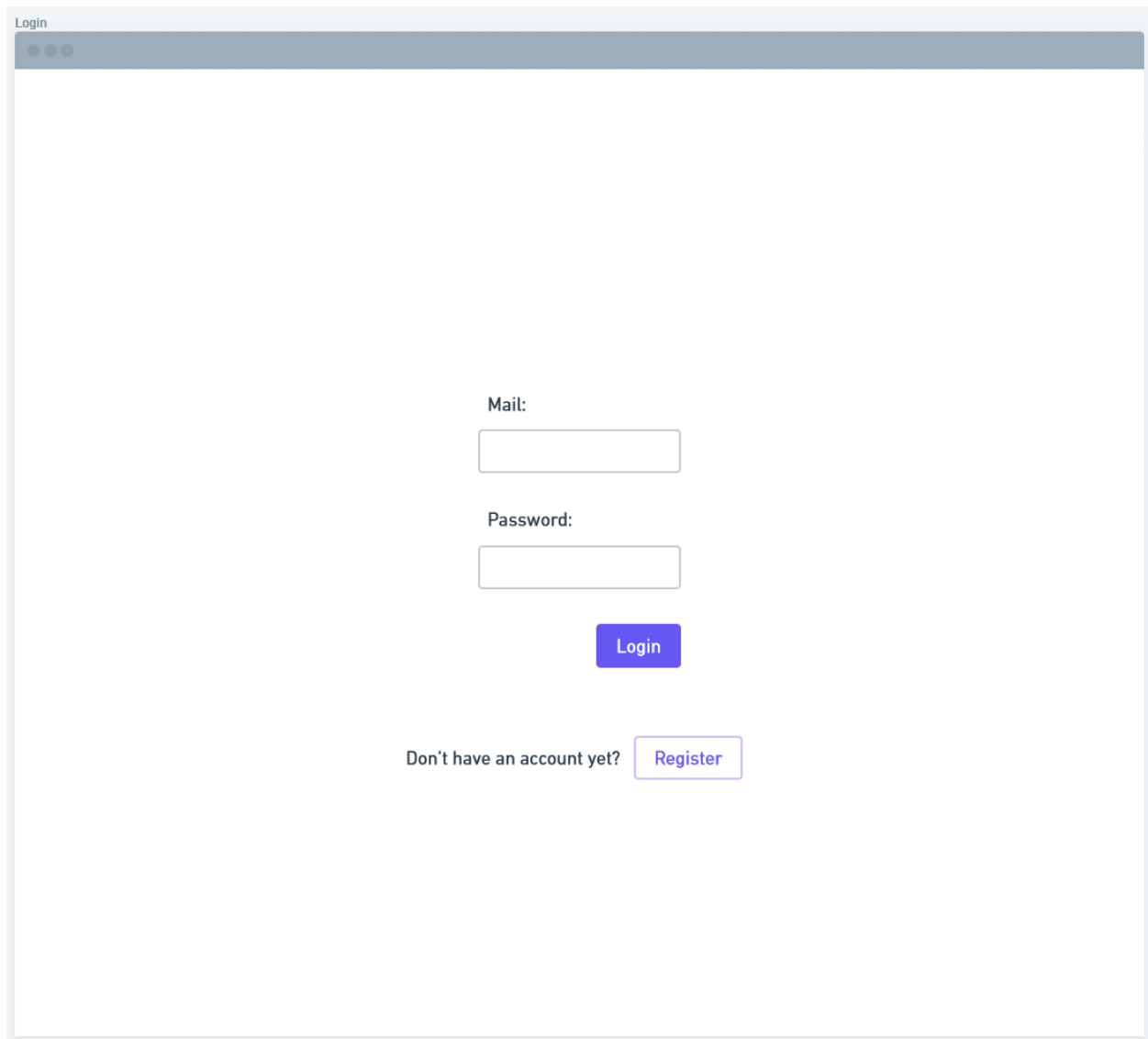


Figura B.1: Boceto Pantalla de Login

Descripción:

- Contiene dos campos de texto (correo y contraseña) y un botón de “Login” para que el usuario se identifique y acceda al contenido de la aplicación.

Requisitos relacionados:

- **1.b:** El usuario podrá acceder al contenido de la aplicación identificándose mediante su correo y contraseña.

Pantalla de registro

Register

Mail:

Username:

Password:

Password:

Already have an account?

Detailed description of the form: The image shows a wireframe of a registration page. At the top left, the word 'Register' is written in a small font. Below it is a dark grey header bar with three white circles on the left. The main content area is white and contains four text input fields stacked vertically. The first is labeled 'Mail:', the second 'Username:', the third 'Password:', and the fourth 'Password:'. Below the third and fourth fields is a blue button with the text 'Register'. At the bottom, there is the text 'Already have an account?' followed by a blue button with the text 'Sign in'.

Figura B.2: Boceto Pantalla de Registro

Descripción:

- Contiene cuatro campos de texto (correo, usuario, contraseña y confirmación de contraseña) y un botón de “Registro” que permitirá crear un nuevo usuario.

Requisitos relacionados:

- **1.a:** El usuario podrá registrarse mediante un correo, un nombre de usuario y una contraseña.

Pantalla principal

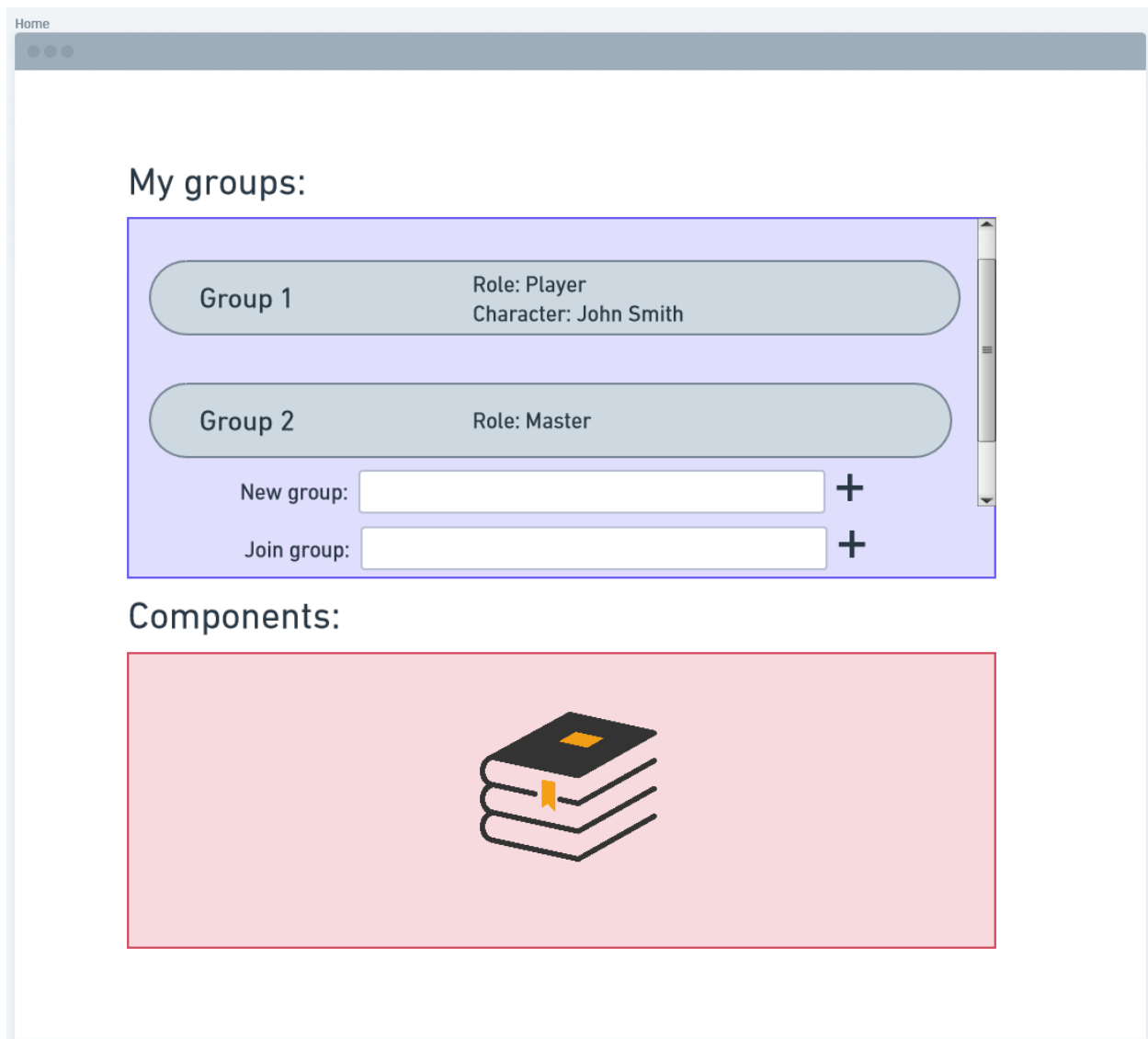


Figura B.3: Boceto Pantalla de Inicio

Descripción:

- Muestra un acceso rápido a los grupos a los que pertenece el usuario y a los componentes oficiales del juego. Además, mediante dos campos de texto, permite la creación rápida de grupos y la unión a grupos ya existentes.

Requisitos relacionados:

- **2.a:** El usuario podrá crear un grupo o unirse a un grupo existente mediante un código único.
- **3.a:** El usuario podrá acceder a un listado de componentes del juego.

Pantalla de búsqueda de componentes

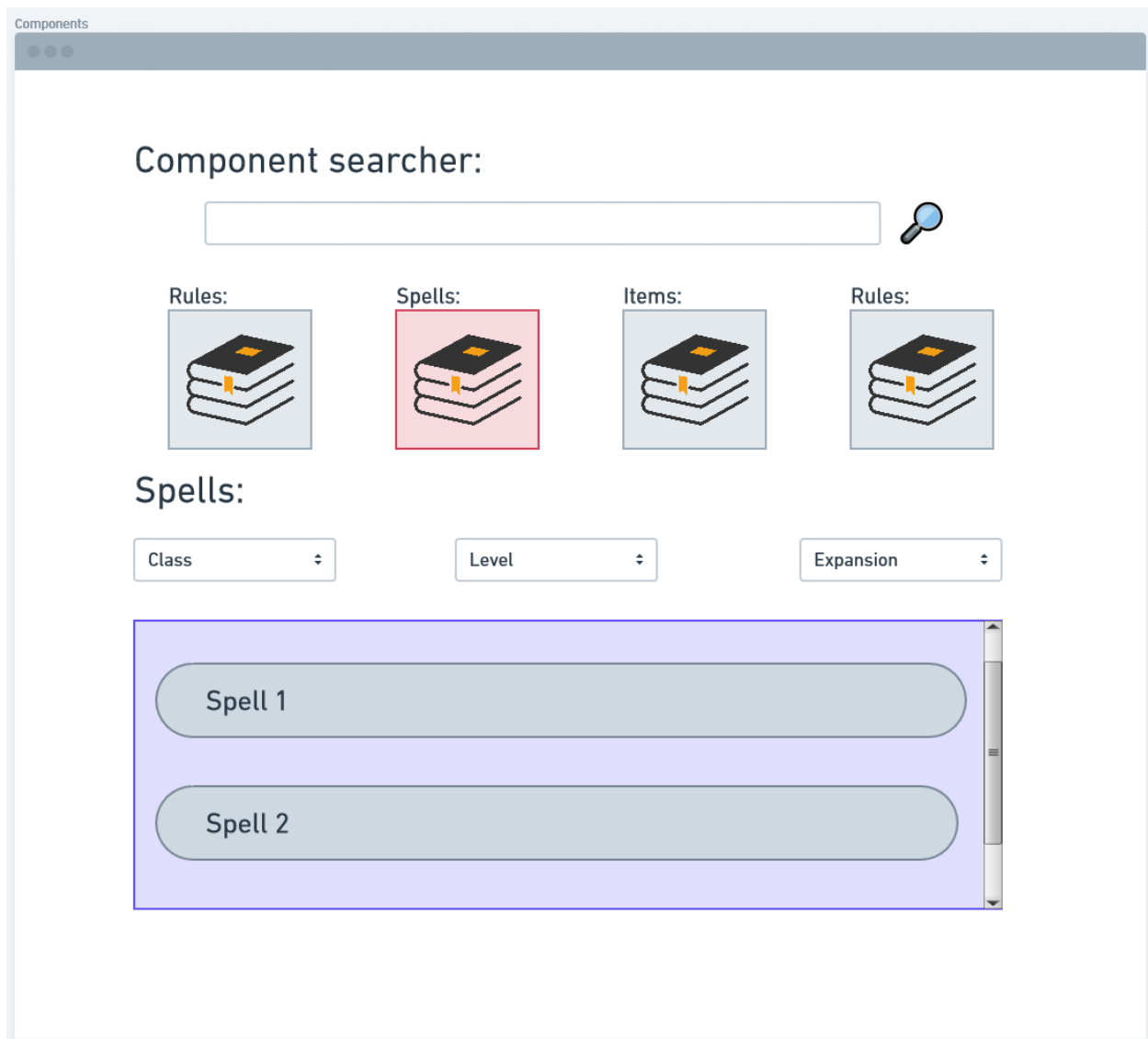


Figura B.4: Boceto Pantalla de Búsqueda de Componentes

Descripción:

- Contiene un selector del tipo de componente a consultar y un listado de estos, además de los pertinentes elementos de filtrado.

Requisitos relacionados:

- **3.a:** El usuario podrá acceder a un listado de componentes del juego.
- **3.b:** El usuario podrá ordenar los componentes según sus parámetros, además de poder utilizar un buscador.

Pantalla de información de componente

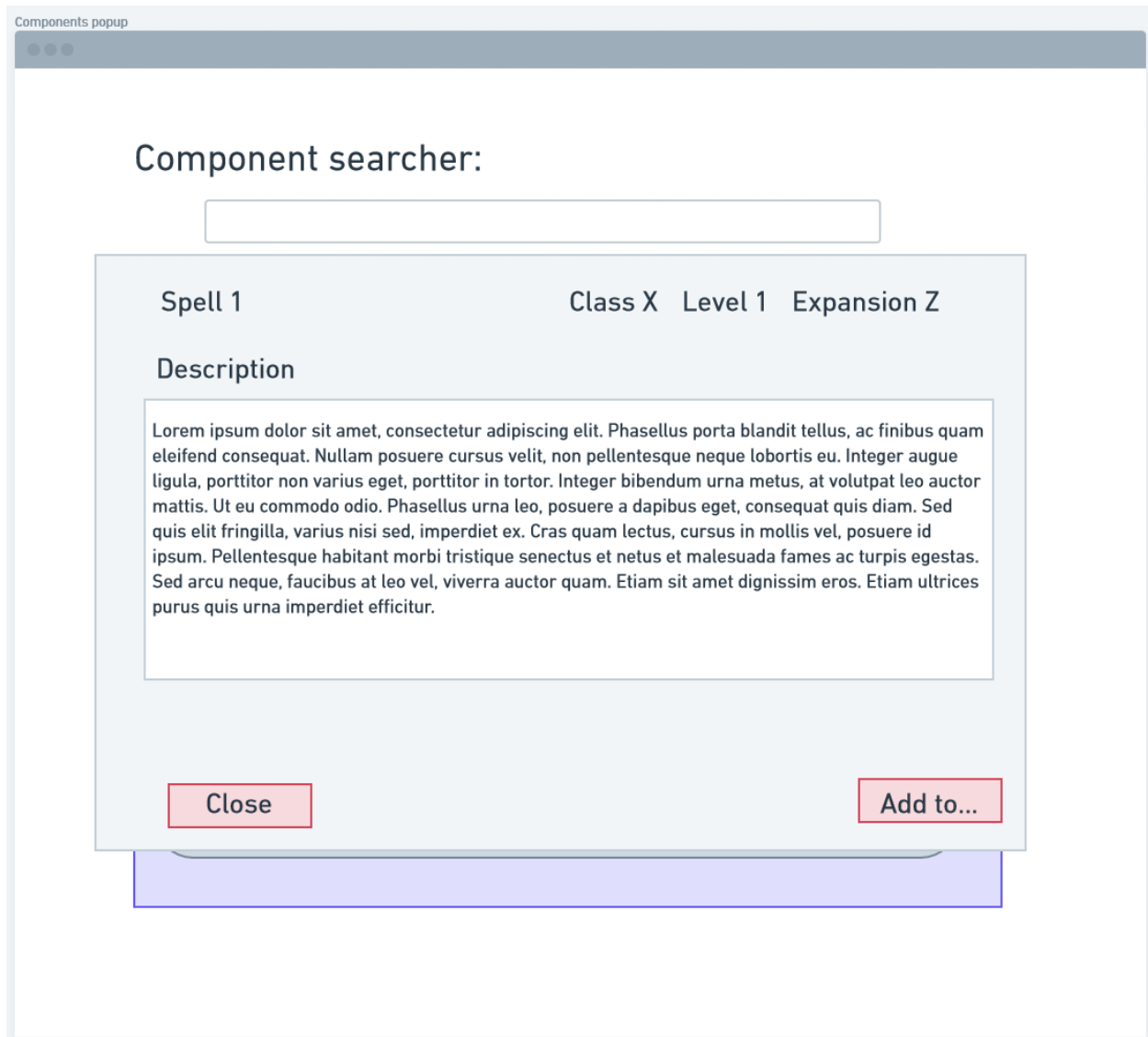


Figura B.5: Boceto Pantalla de Información de Componente

Descripción:

- Ventana emergente que se mostrará al seleccionar un componente. Contendrá la información relativa al componente y permitirá al usuario añadirlo a una ficha de personaje.

Requisitos relacionados:

- **3.a:** El usuario podrá acceder a un listado de componentes del juego.
- **3.c:** El usuario podrá asignar componentes a sus fichas de personaje, para poder acceder fácilmente a ellos sin la necesidad de buscarlos cada vez que los necesite consultar.

Pantalla de información de grupo

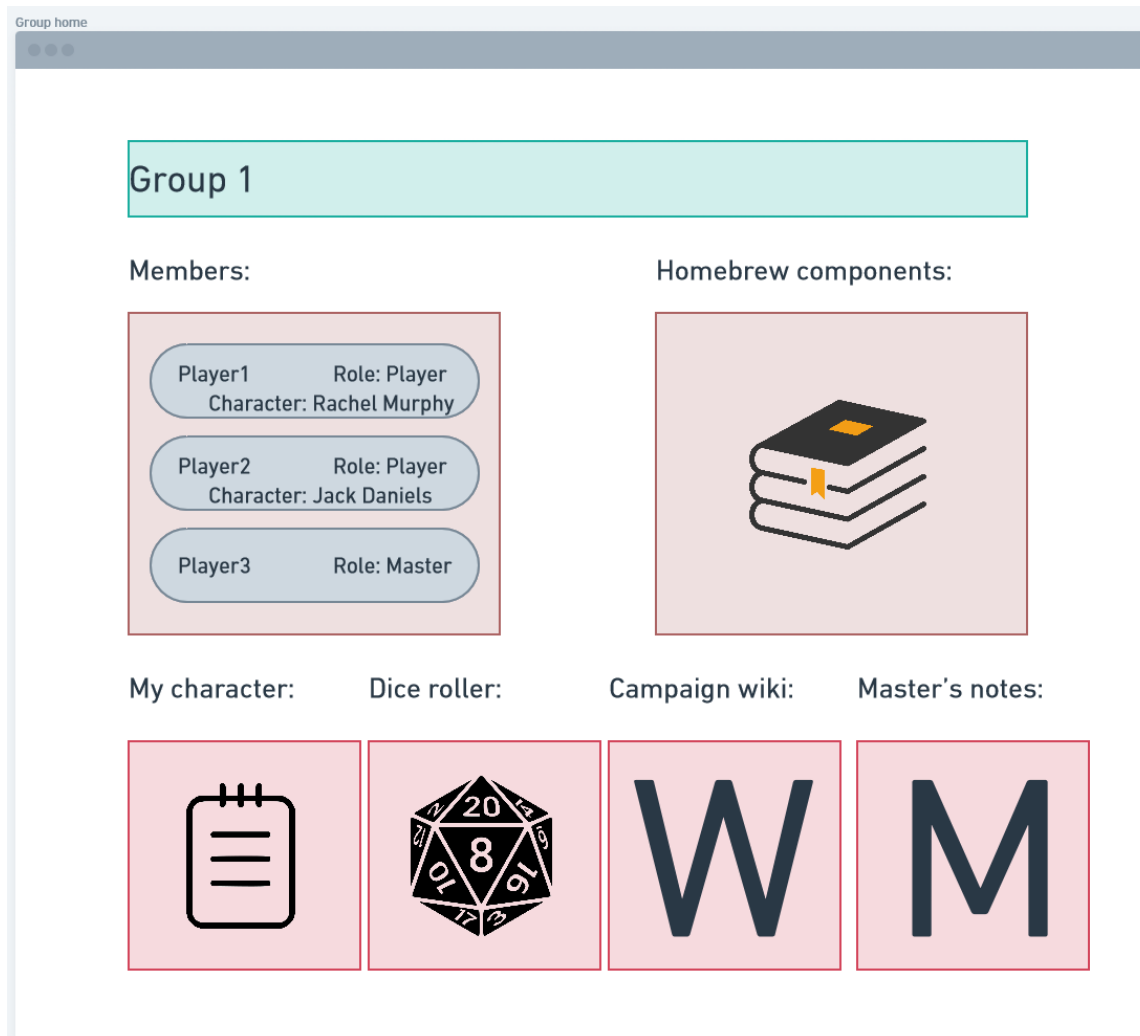


Figura B.6: Boceto Pantalla de Información de Grupo

Descripción:

- Muestra la información básica de un grupo (nombre e integrantes), y permite el acceso al resto de funcionalidades del grupo (personajes, sala de juego, wiki y componentes personalizados). Además, si el usuario es el *Master* del grupo, se mostrará un botón de acceso a las notas de *Master*.

Requisitos relacionados:

- **2.b:** El usuario podrá acceder a una pantalla donde crear su propia ficha de personaje y poder descargarla en formato PDF.
- **2.c:** El usuario podrá acceder a la información de otros usuarios del grupo y sus fichas de personaje.
- **2.d:** El usuario podrá acceder a una pantalla de creación de componentes, donde podrá elaborar sus componentes personalizados que otros miembros del grupo podrán consultar.
- **2.f:** El usuario podrá acceder a una pantalla donde realizar tiradas de dados visibles por

el resto del grupo.

Pantalla de ficha de personaje

Sheet creation

Character Name:

DUNGEONS & DRAGONS®

CLASS & LEVEL BACKGROUND PLAYER NAME

RACE ALIGNMENT EXPERIENCE POINTS

CHARACTER NAME

PROFICIENCY BONUS

INSPIRATION

SAVING THROWS
ATHLETICS

STRENGTH

SAVING THROWS
ACROBATICS
SLEIGHT OF HAND
STEALTH

DEXTERITY

SAVING THROWS

CONSTITUTION

SAVING THROWS
ARCANA
HISTORY
INVESTIGATION
NATURE
RELIGION

INTELLIGENCE

SAVING THROWS
ANIMAL HANDLING
INSIGHT
MEDICINE
PERCEPTION
SURVIVAL

WISDOM

SAVING THROWS
DECEPTION
INTIMIDATION
PERFORMANCE
PERSUASION

CHARISMA

ARMOR CLASS

INITIATIVE

SPEED

HIT POINT MAXIMUM

CURRENT HIT POINTS

TEMPORARY HIT POINTS

Total

HIT DICE

SUCCESSSES

FAILURES

DEATH SAVED

NAME

AGE

BONUS

DAMAGE TYPE

Cancel

Figura B.7: Boceto Pantalla de Ficha de Personaje

Descripción:

- Muestra toda la información relativa a un personaje y permite añadir referencias rápidas a componentes oficiales y personalizados del grupo.

Requisitos relacionados:

- **2.b:** El usuario podrá acceder a una pantalla donde crear su propia ficha de personaje y poder descargarla en formato PDF.
- **2.c:** El usuario podrá acceder a la información de otros usuarios del grupo y sus fichas de personaje.

Pantalla de sala de juego

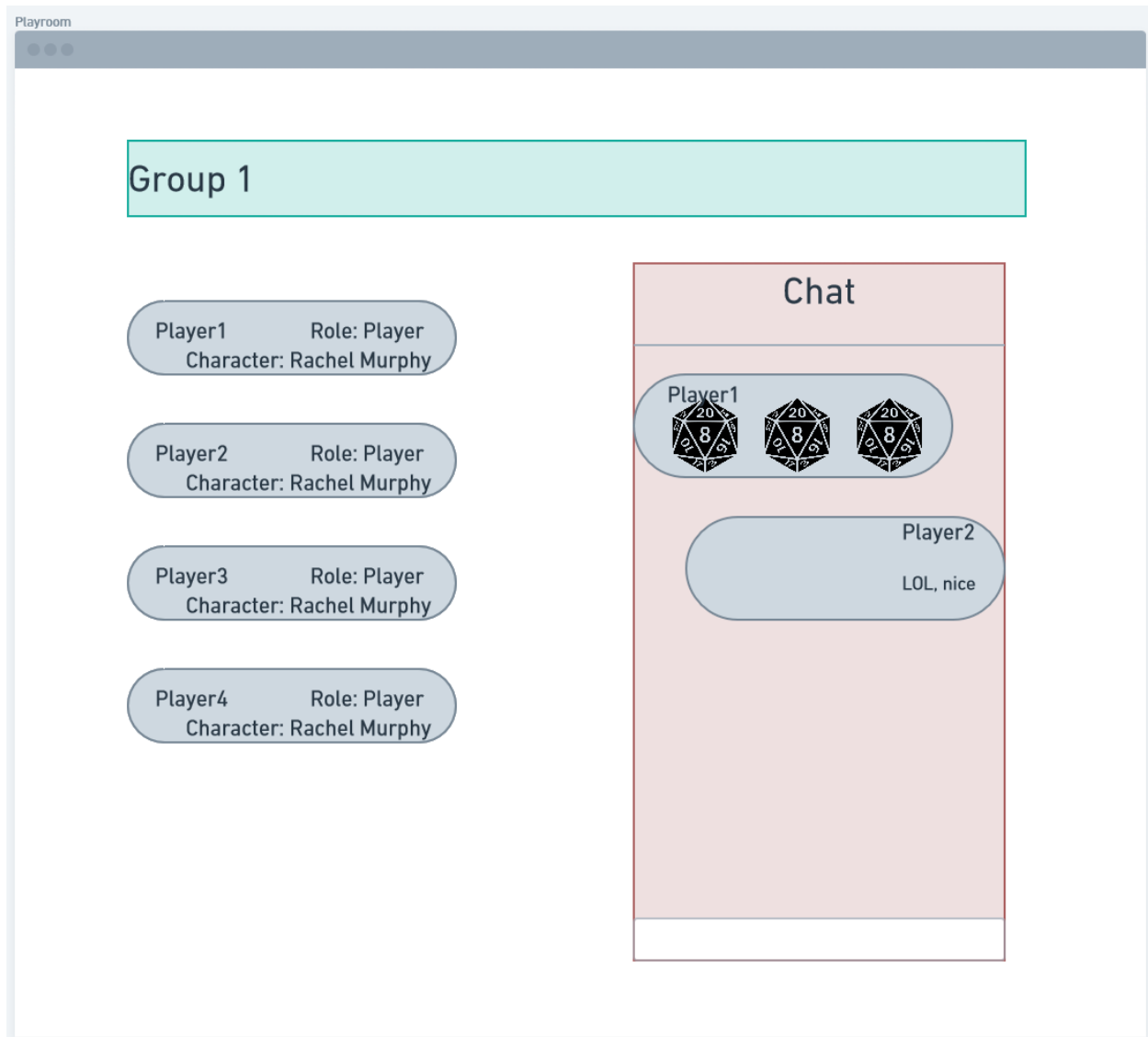


Figura B.8: Boceto Pantalla de Sala de Grupo

Descripción:

- Muestra una pantalla con un chat en vivo en el que los integrantes de un grupo podrán mandar mensajes y hacer tiradas de dados.

Requisitos relacionados:

- **2.f:** El usuario podrá acceder a una pantalla donde realizar tiradas de dados visibles por el resto del grupo.

Pantalla de buscar componentes personalizados

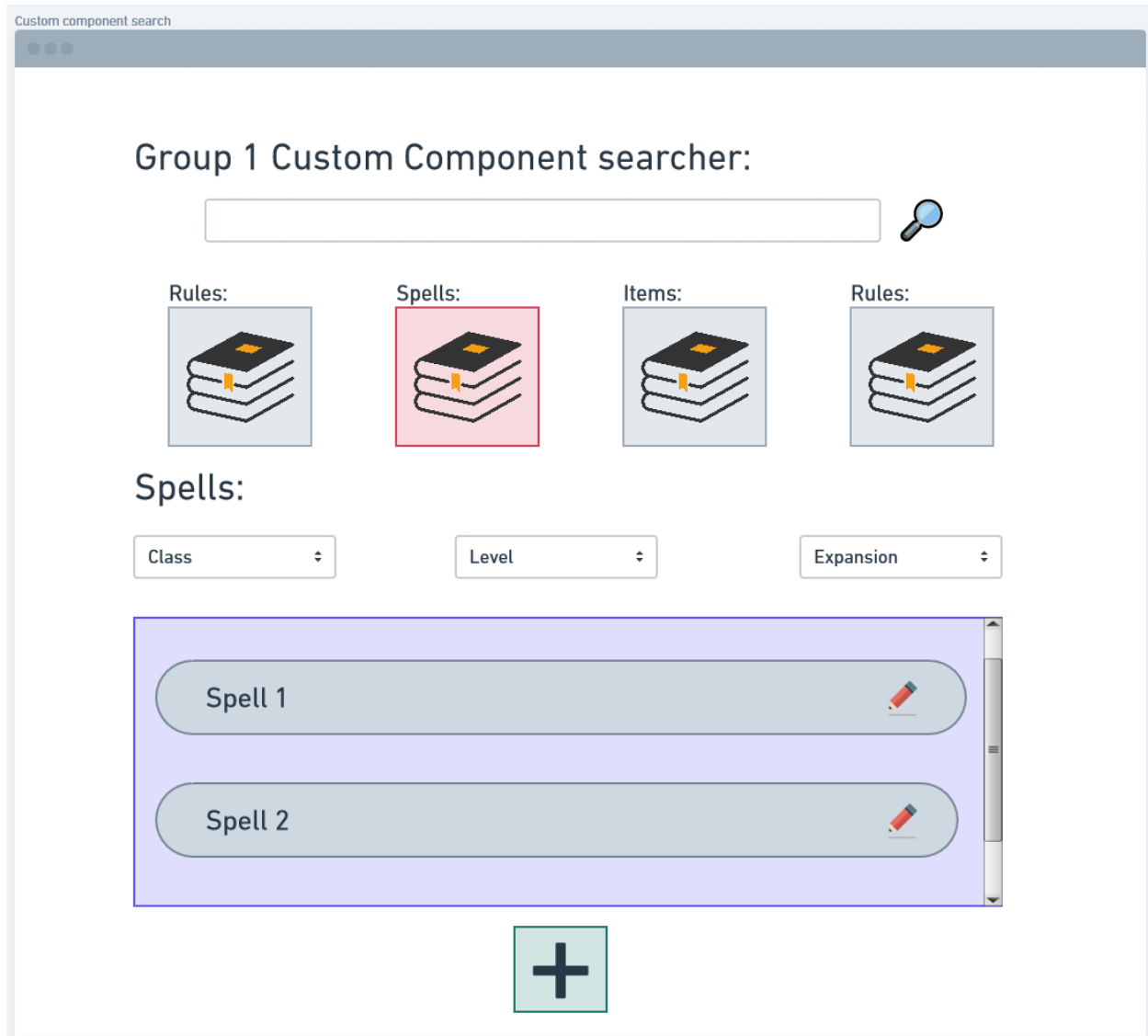


Figura B.9: Boceto Pantalla de Búsqueda de Componentes Personalizados

Descripción:

- Similar al buscador de componentes oficial, pero únicamente con los componentes personalizados de un grupo.

Requisitos relacionados:

- **2.d:** El usuario podrá acceder a una pantalla de creación de componentes, donde podrá elaborar sus componentes personalizados que otros miembros del grupo podrán consultar.

- **2.e:** El usuario podrá visualizar y editar los componentes personalizados del grupo.

Pantalla de información de componentes personalizados

Custom component info

Custom component

Name

Lorem

Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus porta blandit tellus, ac finibus quam eleifend consequat. Nullam posuere cursus velit, non pellentesque neque lobortis eu. Integer augue ligula, porttitor non varius eget, porttitor in tortor. Integer bibendum urna metus, at volutpat leo auctor mattis. Ut eu commodo odio. Phasellus urna leo, posuere a dapibus eget, consequat quis diam. Sed quis elit fringilla, varius nisi sed, imperdiet ex. Cras quam lectus, cursus in mollis vel, posuere id ipsum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed arcu neque, faucibus at leo vel, viverra auctor quam. Etiam sit amet dignissim eros. Etiam ultrices purus quis urna imperdiet efficitur.

Discard changes

Accept

Figura B.10: Boceto Pantalla de Creación de Componentes Personalizados

Descripción:

- Permite a los integrantes de un grupo crear nuevos componentes personalizados o editar los ya existentes.

Requisitos relacionados:

- **2.d:** El usuario podrá acceder a una pantalla de creación de componentes, donde podrá elaborar sus componentes personalizados que otros miembros del grupo podrán consultar.
- **2.e:** El usuario podrá visualizar y editar los componentes personalizados del grupo.

Pantalla de entrada de wiki

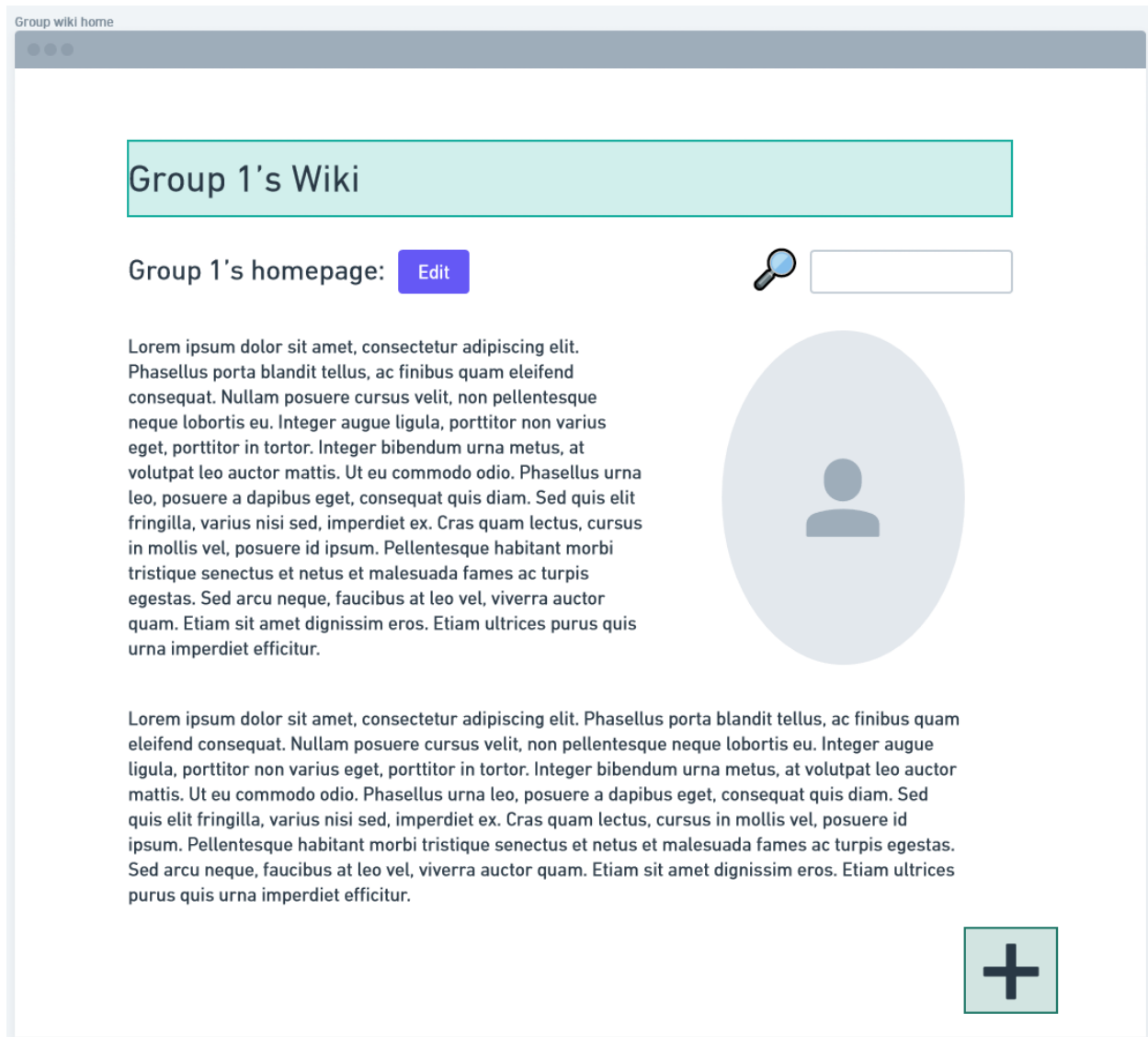


Figura B.11: Boceto Pantalla de Entrada de Wiki

Descripción:

- Muestra el contenido de una entrada de la wiki de un grupo. Permite el acceso a otras entradas mediante un buscador.

Requisitos relacionados:

- **4.a:** Cada grupo dispondrá de una wiki en la que los usuarios podrán consultar información personalizada relativa a sus grupos o campañas.

Pantalla de editar/crear entrada de wiki

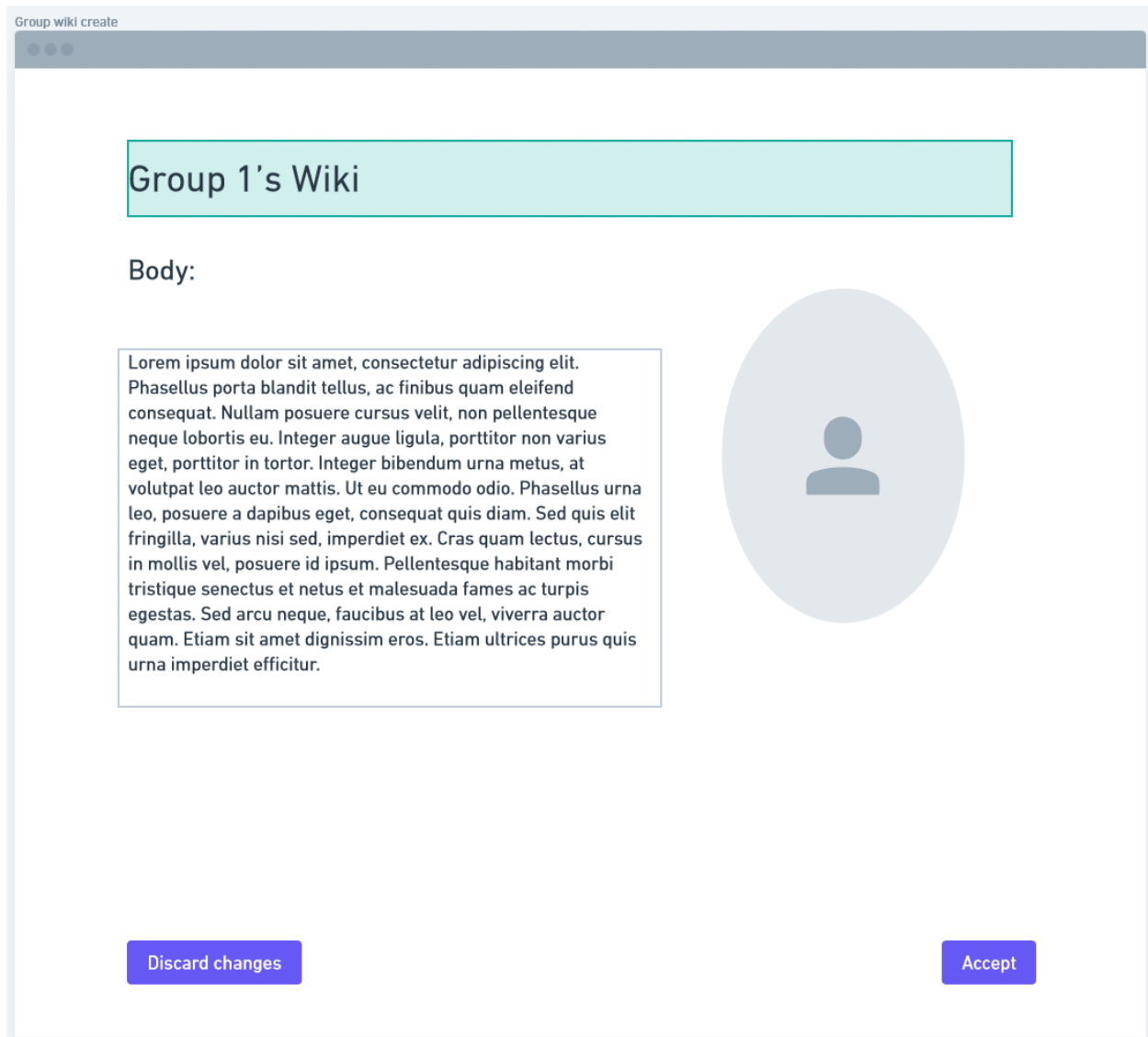


Figura B.12: Boceto Pantalla de Creación de Entrada de Wiki

Descripción:

- Muestra la información básica de un grupo (nombre e integrantes), y permite el acceso al resto de funcionalidades del grupo (personajes, sala de juego, wiki y componentes personalizados). Además, si el usuario es el *Master* del grupo, se mostrará un botón de acceso a las notas de *Master*.

Requisitos relacionados:

- **4.b:** Los usuarios de un grupo podrán añadir nuevas entradas personalizadas a la wiki o editar las que ya existen.

Pantalla de buscar notas de *Master*

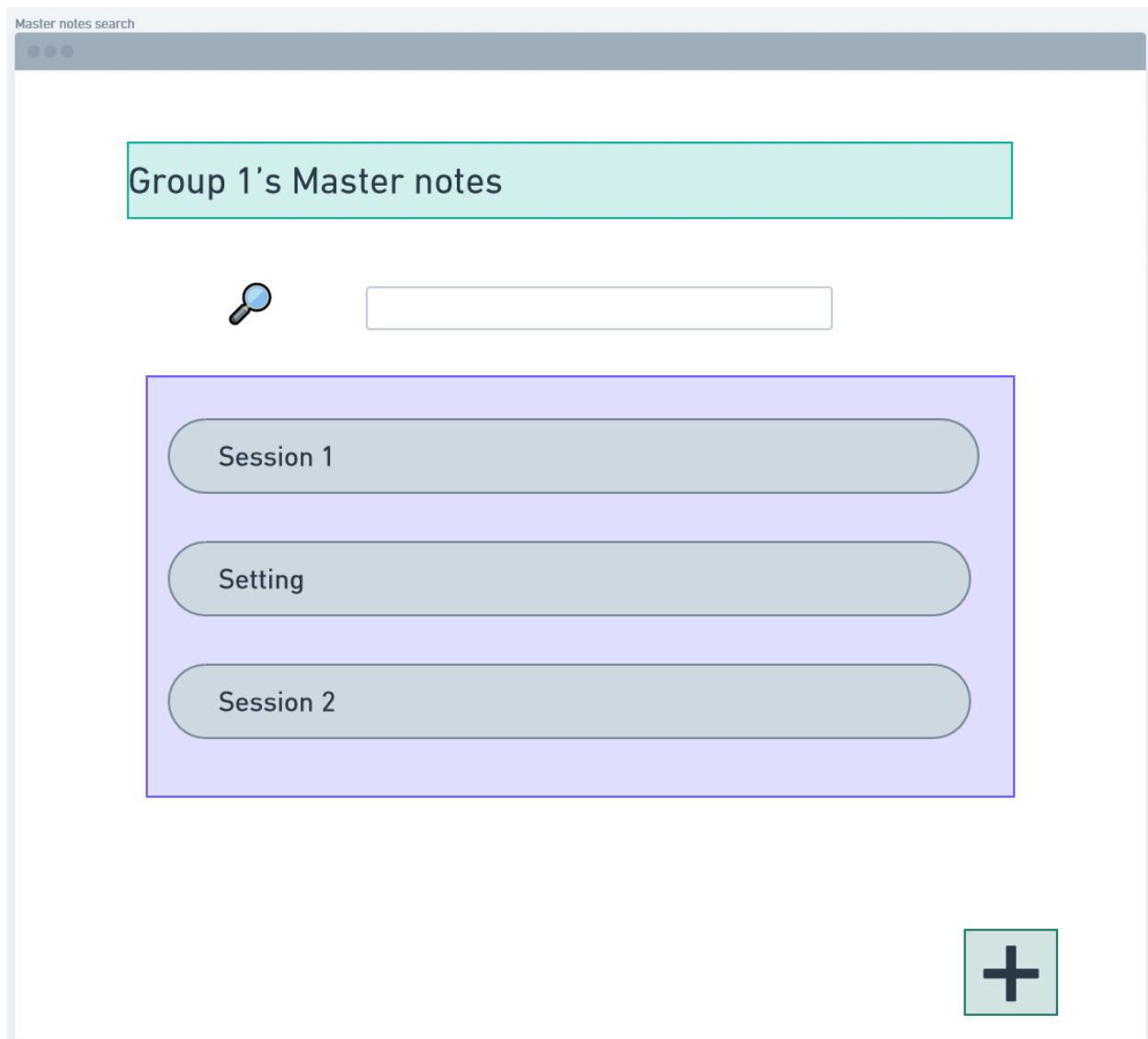


Figura B.13: Boceto Pantalla de Búsqueda de Notas de Master

Descripción:

- Permite al *Master* buscar y filtrar las notas previamente creadas.

Requisitos relacionados:

- **5.a:** El máster de un grupo tendrá acceso a una pantalla de diario, similar a la wiki pero que el resto de miembros del grupo no podrán ver, para planificar sesiones de juego futuras.

Pantalla de nota de *Master*

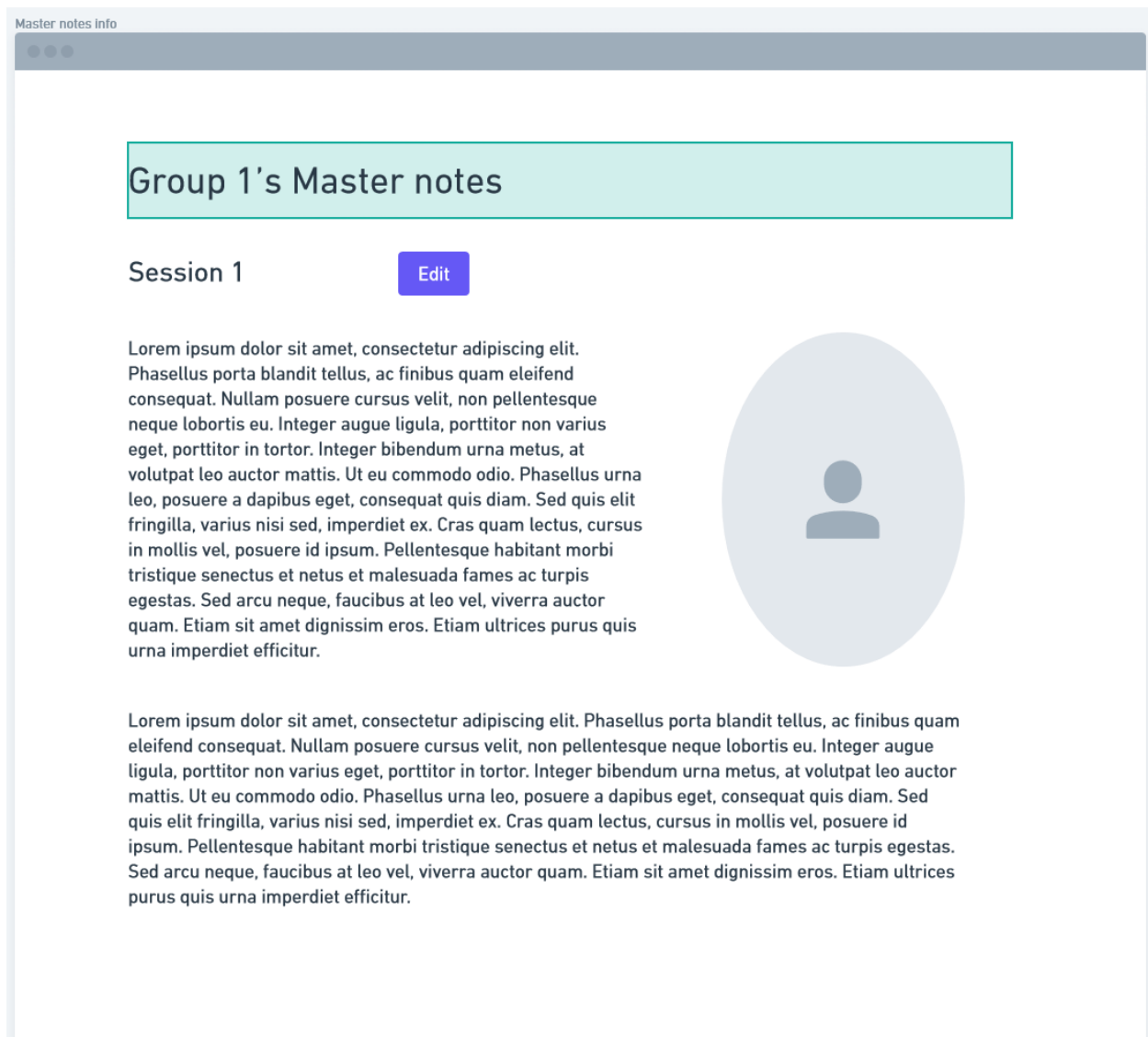


Figura B.14: Boceto Pantalla de Nota de *Master*

Descripción:

- Muestra el contenido de una nota del *Master*.

Requisitos relacionados:

- **5.a:** El máster de un grupo tendrá acceso a una pantalla de diario, similar a la wiki pero que el resto de miembros del grupo no podrán ver, para planificar sesiones de juego futuras.

Pantalla de editar/crear nota de *Master*

Master note create

Group 1's Master notes

Body:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus porta blandit tellus, ac finibus quam eleifend consequat. Nullam posuere cursus velit, non pellentesque neque lobortis eu. Integer augue ligula, porttitor non varius eget, porttitor in tortor. Integer bibendum urna metus, at volutpat leo auctor mattis. Ut eu commodo odio. Phasellus urna leo, posuere a dapibus eget, consequat quis diam. Sed quis elit fringilla, varius nisi sed, imperdiet ex. Cras quam lectus, cursus in mollis vel, posuere id ipsum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed arcu neque, faucibus at leo vel, viverra auctor quam. Etiam sit amet dignissim eros. Etiam ultrices purus quis urna imperdiet efficitur.

Discard changes

Accept

Figura B.15: Boceto Pantalla de Creación de Nota de Master

Descripción:

- Permite la creación de una nueva nota de *Master* o la edición de una ya existente.

Requisitos relacionados:

- **5.b:** El máster de un grupo podrá añadir nuevas notas a su diario o editar las que ya existen.

Anexo III - Manual de Usuario

Este anexo contiene un manual para el usuario final, describiendo todas las pantallas de la aplicación y las funcionalidades que ofrecen.

Pantalla de login

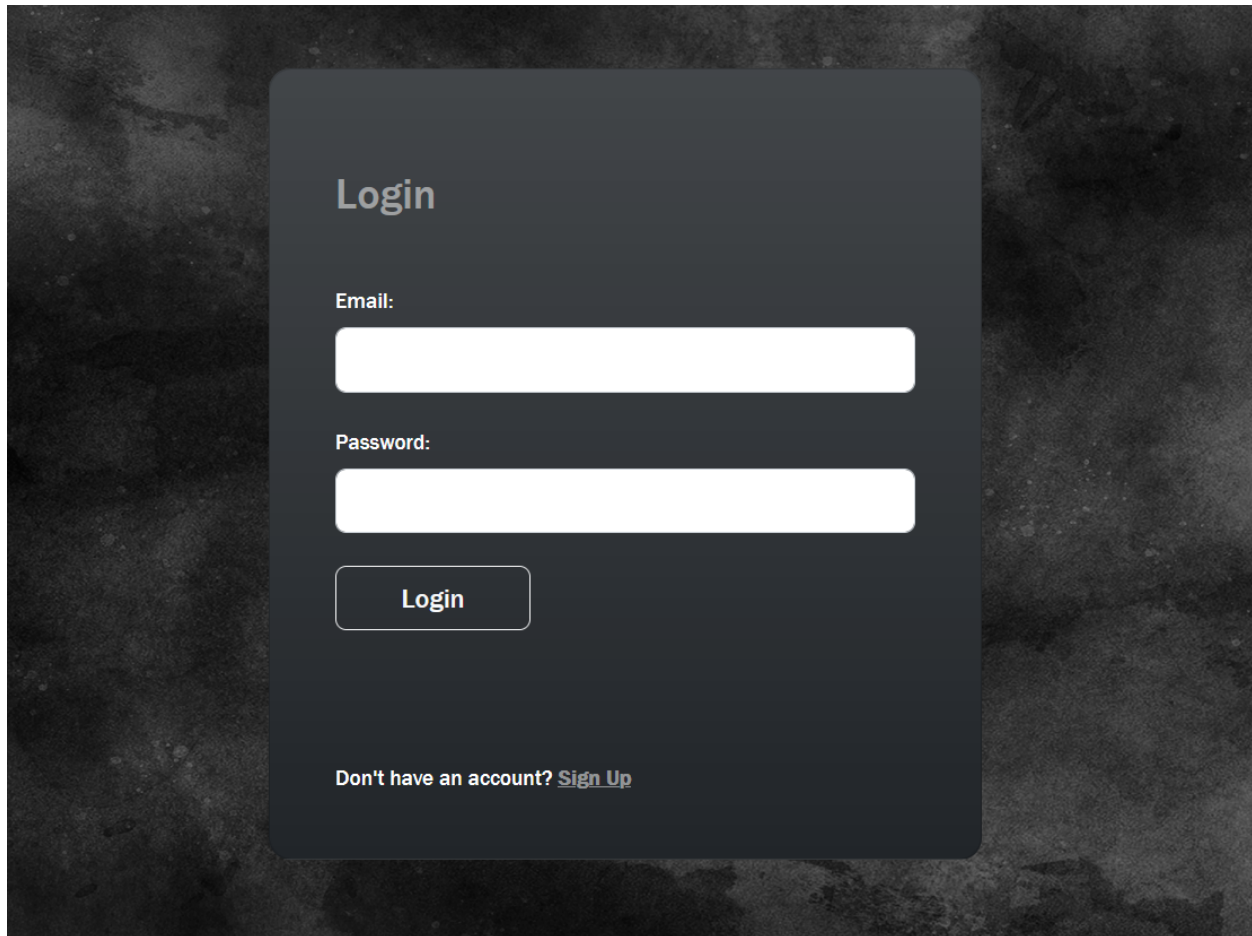
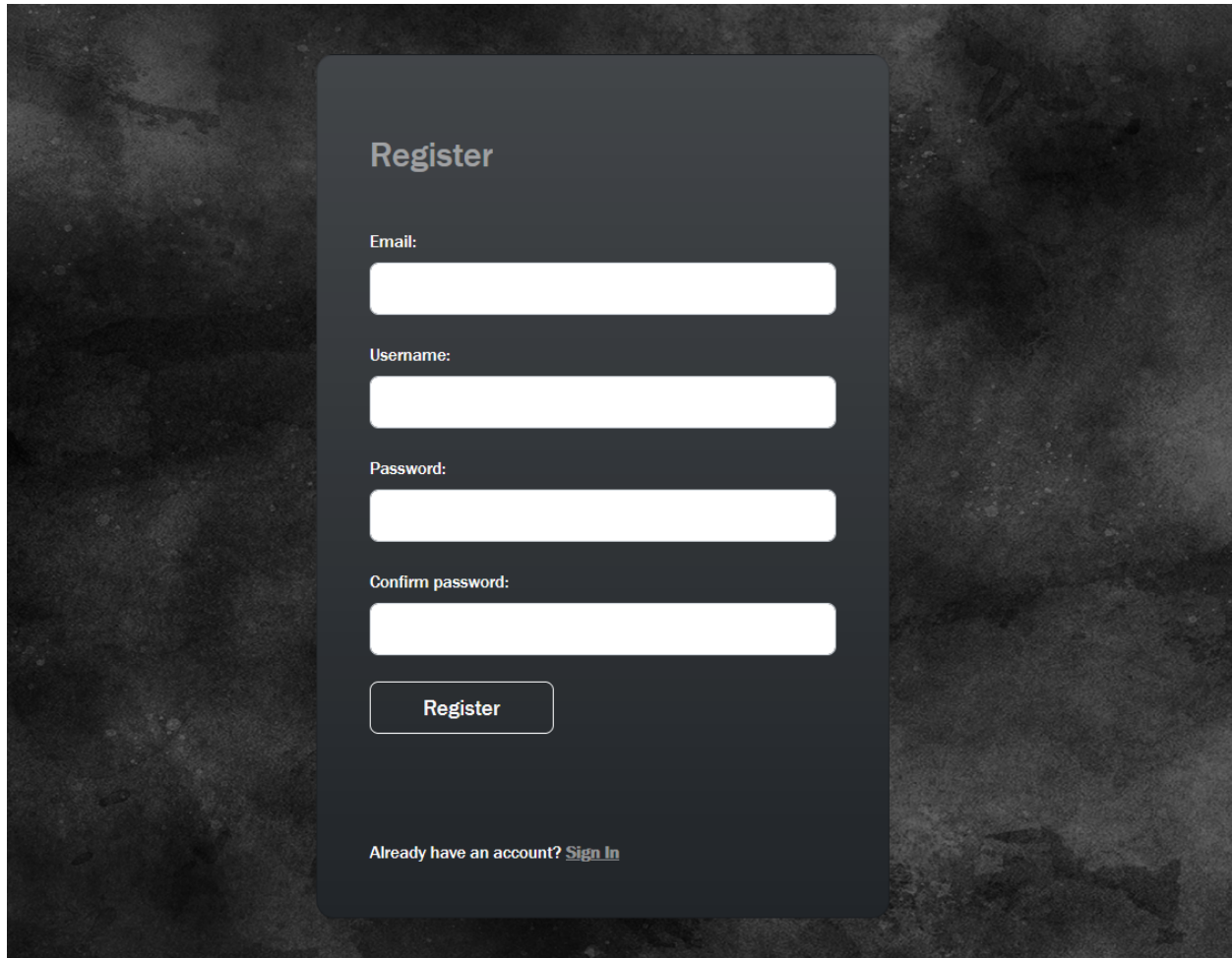


Figura C.1: Pantalla Final de Login

La pantalla de login contiene dos campos de texto: uno para el correo electrónico y otro para la contraseña. Cuando el usuario se identifica de forma correcta, se le redirigirá a la pantalla inicial de la aplicación y se guardará su clave de usuario. Esta clave permitirá al usuario acceder al resto de pantallas de la aplicación sin necesidad de volver a identificarse en los siguientes 30 días o hasta que cierre la sesión.

Pantalla de registro



The image shows a registration form titled "Register" on a dark background. The form contains four text input fields: "Email:", "Username:", "Password:", and "Confirm password:". Below the fields is a "Register" button. At the bottom of the form, there is a link: "Already have an account? [Sign In](#)".

Figura C.2: Pantalla Final de Registro

Esta pantalla contiene cuatro campos de texto: correo electrónico, nombre de usuario, contraseña y confirmación de contraseña. Si se introducen correctamente los datos y se presiona el botón *Register*, se creará una nueva cuenta de usuario que permitirá el acceso a la aplicación.

El correo tendrá que ser una dirección válida, el nombre contendrá menos de 12 caracteres y la contraseña tendrá que tener entre 6 y 16 caracteres y al menos una mayúscula y un número. Si el usuario introduce datos no válidos, el sistema informará de los errores.

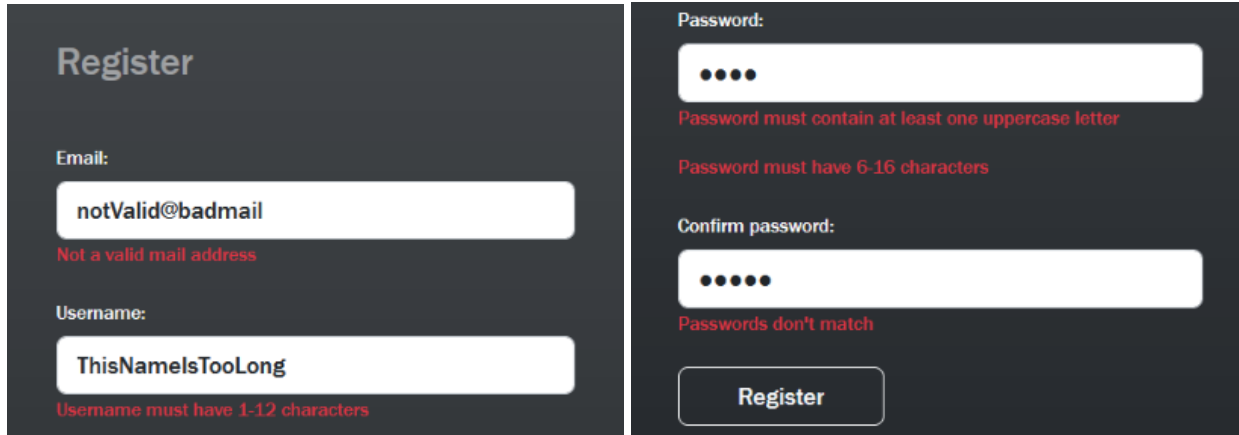


Figura C.3: Notificación Errores de Registro

Pantalla inicial

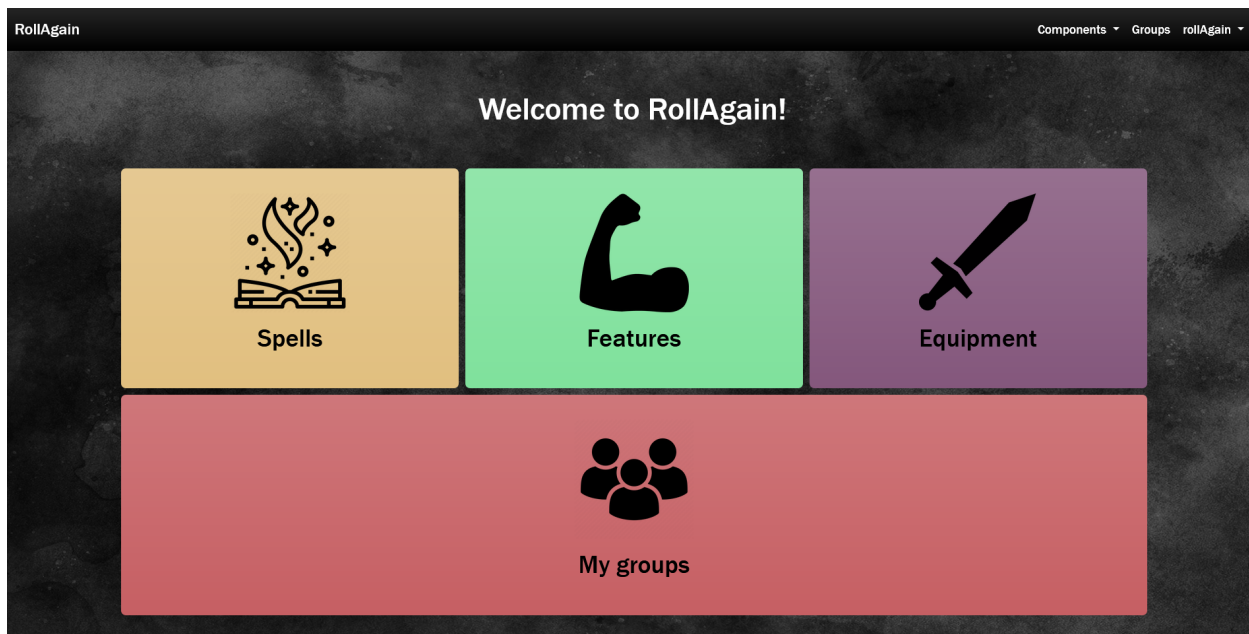


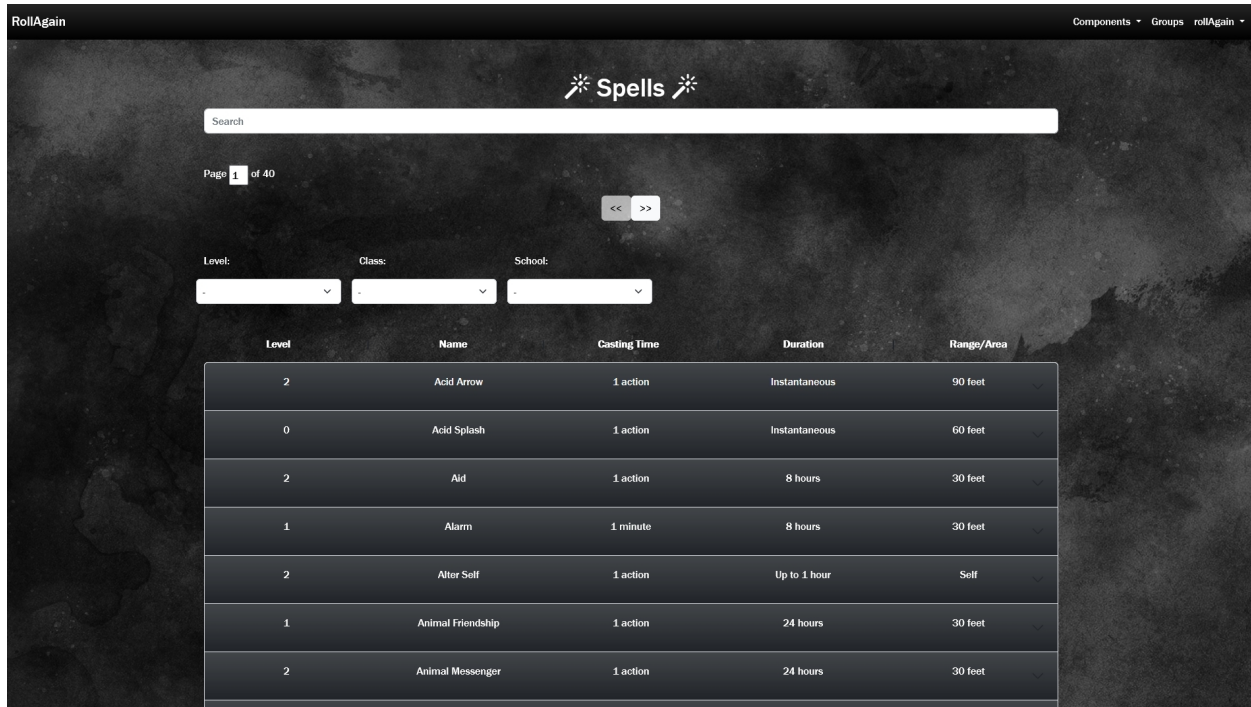
Figura C.4: Pantalla Final de Inicio

La pantalla inicial muestra un acceso rápido a las principales funcionalidades de la aplicación. Se pueden observar tres botones que llevarán a las pantallas de *Hechizos*, *Rasgos* o *Equipamiento* respectivamente. Debajo habrá un botón *Mis grupos* que llevará al listado de grupos del usuario.

Además, se puede observar la barra de navegación, que permite acceder rápidamente a esta pantalla (haciendo click en el botón *RollAgain* del extremo izquierdo). A la derecha hay botones que permiten atajar a las pantallas de componentes mediante un desplegable, acceder rápidamente a la pantalla de grupos y cerrar sesión haciendo clic en *LogOut*, tras mostrar el desplegable bajo el nombre de usuario.

Pantalla de componentes de juego

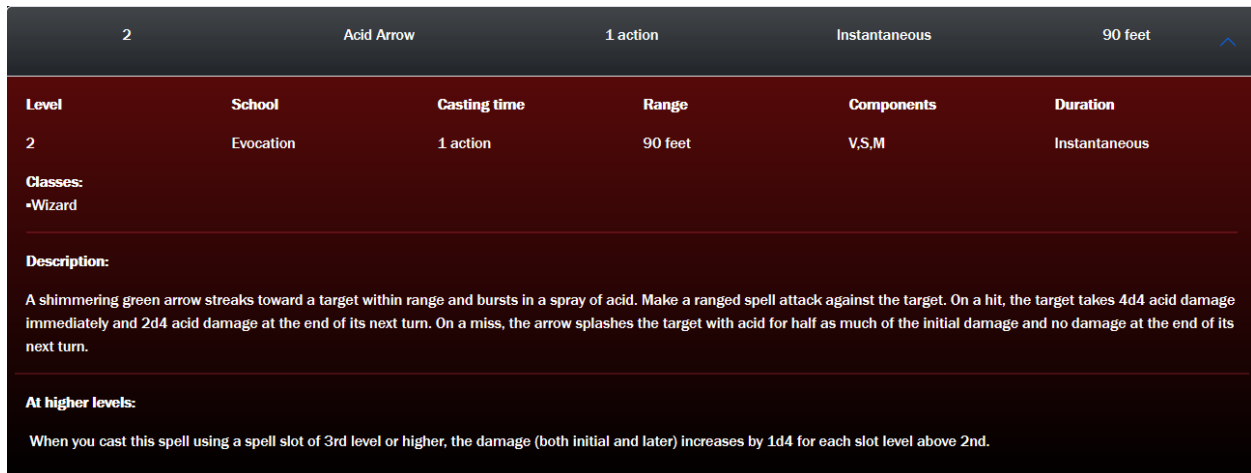
Como se ha mencionado, hay tres tipos de componentes de juego: *Hechizos*, *Rasgos* y *Objetos*, y cada uno tiene su propia pantalla. Las tres pantallas son muy similares, pues sirven para listar los elementos y poder filtrarlos.



Level	Name	Casting Time	Duration	Range/Area
2	Acid Arrow	1 action	Instantaneous	90 feet
0	Acid Splash	1 action	Instantaneous	60 feet
2	Aid	1 action	8 hours	30 feet
1	Alarm	1 minute	8 hours	30 feet
2	Alter Self	1 action	Up to 1 hour	Self
1	Animal Friendship	1 action	24 hours	30 feet
2	Animal Messenger	1 action	24 hours	30 feet

Figura C.5: Pantalla Final de Búsqueda de Componentes

Al seleccionar un elemento de la lista, se mostrará un desplegable con información detallada:



Level	School	Casting time	Range	Components	Duration
2	Evocation	1 action	90 feet	V,S,M	Instantaneous

Classes:
-Wizard

Description:
A shimmering green arrow streaks toward a target within range and bursts in a spray of acid. Make a ranged spell attack against the target. On a hit, the target takes 4d4 acid damage immediately and 2d4 acid damage at the end of its next turn. On a miss, the arrow splashes the target with acid for half as much of the initial damage and no damage at the end of its next turn.

At higher levels:
When you cast this spell using a spell slot of 3rd level or higher, the damage (both initial and later) increases by 1d4 for each slot level above 2nd.

Figura C.6: Desplegable Información de Componente

Mediante la barra de búsqueda se podrán filtrar los distintos componentes por su nombre, y mediante las cajas desplegadas se podrán establecer otros parámetros (como el nivel y clase para los que están disponibles los componentes).

Además, haciendo click en las cabeceras de la lista, se podrán ordenar según el parámetro seleccionado (por ejemplo, nivel ascendente/descendente).

Pantalla de grupos

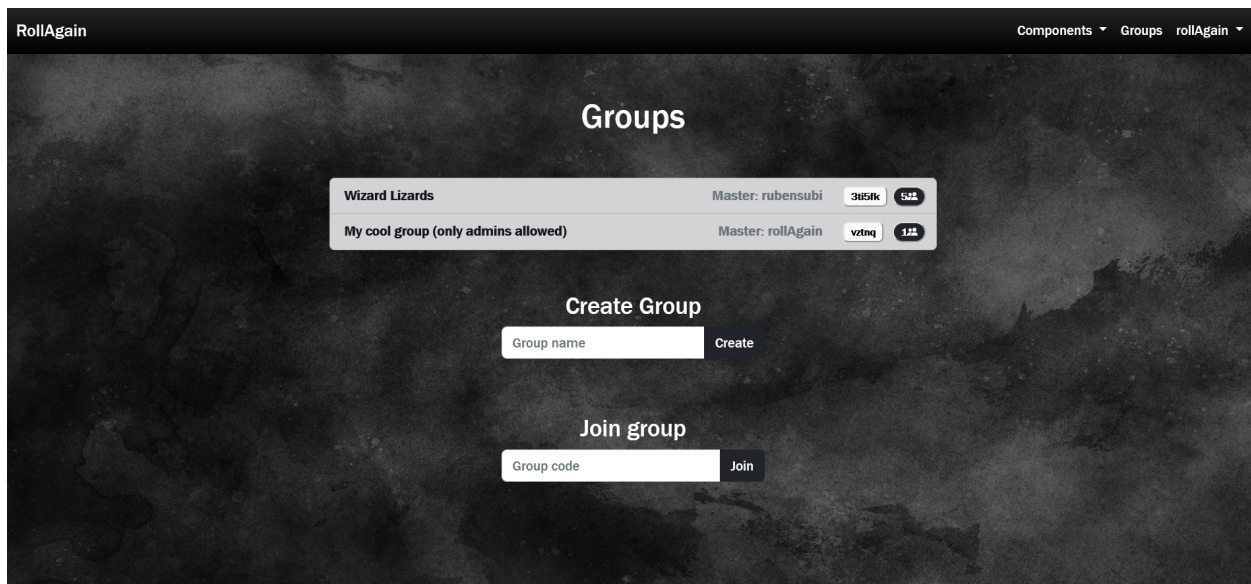


Figura C.7: Pantalla Final de Grupos

En esta pantalla se mostrarán todos los grupos a los que pertenece el usuario. Además, habrá dos campos de texto con sus respectivos botones; uno servirá para crear un nuevo grupo con el nombre indicado, mientras que el otro permitirá, mediante un código único, unirse a un grupo existente.

Al lado de cada grupo se mostrará el *Master* de este, un botón que copiará el código de invitación al portapapeles y otro botón que desplegará un listado con los integrantes del grupo.

Pantalla de información de grupo

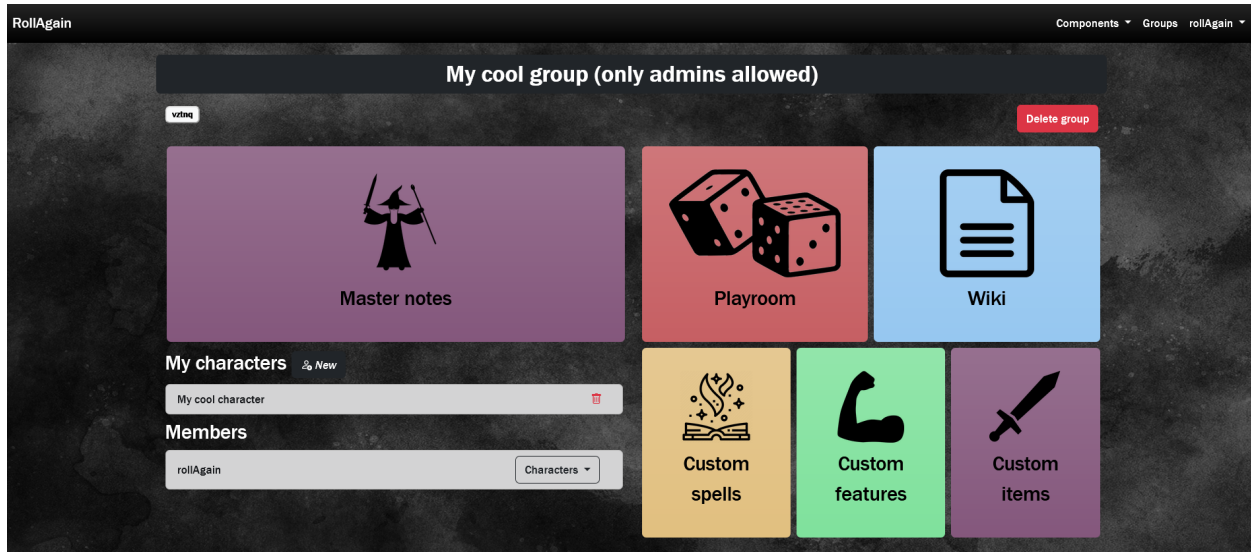


Figura C.8: Pantalla Final de Información de Grupo

Al entrar en la pantalla de un grupo, se podrán ver, a la derecha, los botones de acceso a la sala de juego, a la wiki y a los distintos componentes personalizados creados por ese grupo. También, a la izquierda, estarán presentes una lista de los personajes del usuario en ese grupo y un listado de los miembros del grupo.

El usuario podrá crear nuevos personajes haciendo click en el botón junto a *My characters*, que mostrará una ventana emergente para introducir el nombre del personaje. Además, se podrán consultar las fichas de personaje de otros miembros del grupo mediante un desplegable al lado de su nombre.

El *Master* del grupo podrá ver un botón correspondiente a las notas del *Master*, así como un botón en la esquina superior derecha para borrar el grupo. El resto de miembros del grupo no podrá ver el botón de *Master notes*, y el botón de borrar grupo se verá sustituido por un botón para abandonar el grupo.

Pantalla de personaje

The screenshot shows a character sheet for a Dwarf Wizard. The character's name is "My cool character". The class is Wizard, level 1, and the race is Dwarf. The background is Noble and the alignment is True neutral. The experience points (Xp) are 100. The casting ability is Intelligence, with a spell save DC of 2 and a spell bonus of 3. The character's stats are: Strength 12 (+1), Dexterity 10 (0), Constitution 9 (-1), Intelligence 15 (+2), Wisdom 9 (-1), and Charisma 12 (+1). The proficiency bonus is 2. The saving throws are: Strength 1, Dexterity 0, Constitution -1, Intelligence +4, Wisdom -1, and Charisma 1. The skills are: Acrobatics 0, Animal Handling -1, Arcana 2, Athletics 1, Deception 1, History +4, Insight -1, Investigation 1, Investigation 2, Medicine -1, Medicine +4, Perception -1, Performance 1, Persuasion 1, Religion 2, Secret of Hand 0, Stealth 0, and Survival -4. The character's AC is 8, Initiative is 2, and Speed is 25. The HP is 9/9 and Temporary HP is 0. The hit dice is 1d10 with a maximum of 10. The death saves are 0/3. The inventory is empty.

Figura C.9: Pantalla Final de Ficha de Personaje

La pantalla de personaje mostrará la ficha de un personaje creado por un usuario. Esta pantalla contendrá la información que contendría una ficha oficial de *Dragones y Mazmorras 5e*, como el nombre, retrato, clase y raza del personaje, sus atributos y habilidades, tiradas de salvación, etc.

Además, si el usuario es el dueño del personaje, podrá acceder al modo edición mediante un botón al lado del nombre.

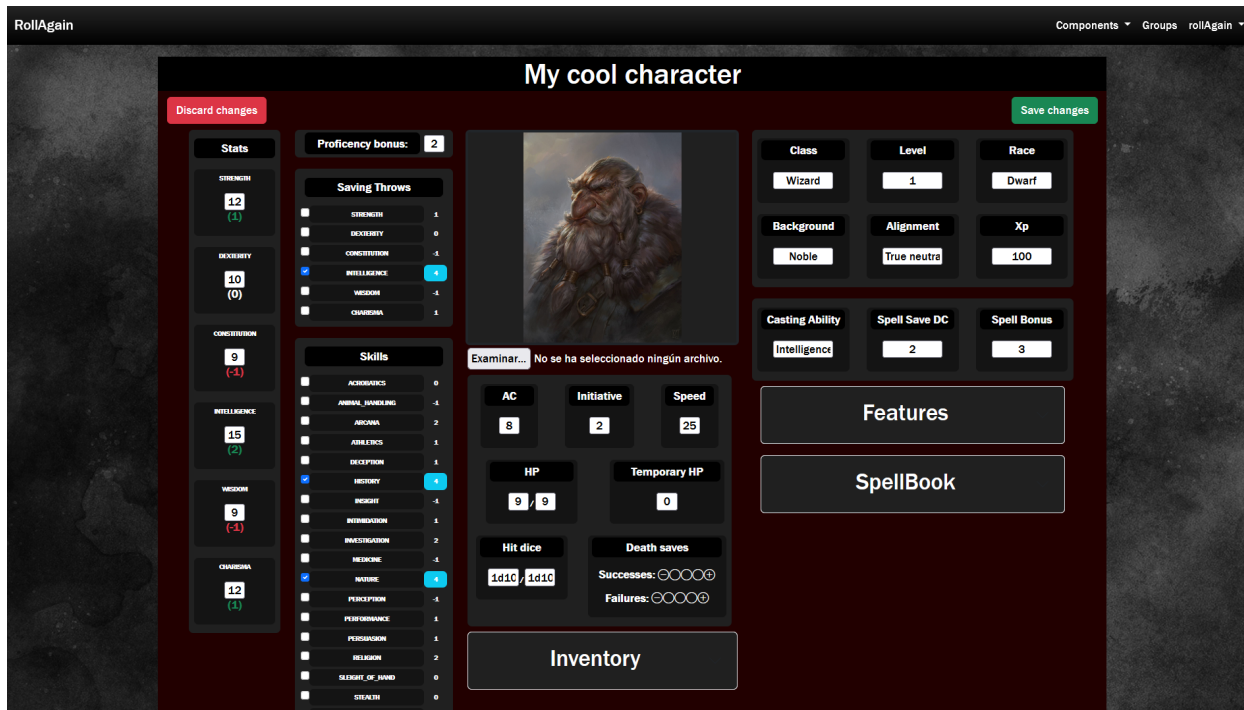


Figura C.10: Pantalla Final de Edición de Ficha de Personaje

En el modo edición, se podrán editar todos los campos de información del personaje, a excepción de las tiradas de salvación y habilidades. Estos funcionarán de forma distinta, “escalando” cada uno de ellos con un atributo distinto. De esta forma, podremos observar que al aumentar un atributo aumentarán también todas las habilidades que escalen con dicho atributo. Además, a las habilidades y salvaciones marcadas con un check se les añadirá el bonus de proficiencia (también especificado por el usuario).

En la pantalla de personaje se podrán observar también tres desplegables (Inventario, Rasgos y Libro de hechizos), que permitirán añadir componentes de juego (tanto oficiales como personalizados) a la ficha del personaje para facilitar un acceso rápido.

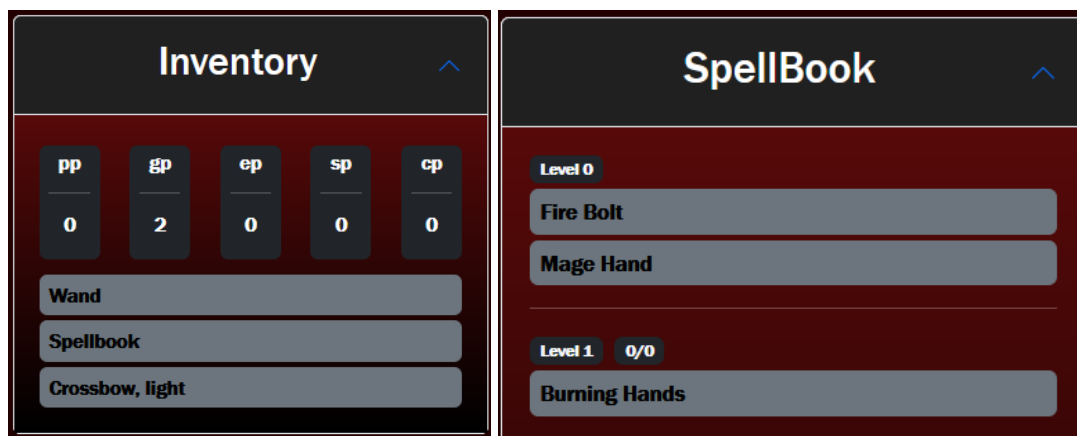


Figura C.11: Desplegables Inventario y Libro de Hechizos de Personaje



Figura C.12: Desplegable Rasgos de Personaje

En el modo edición, se podrán añadir y eliminar componentes al personaje mediante un desplegable similar al buscador de componentes. En este desplegable se podrán buscar tanto componentes oficiales como personalizados.

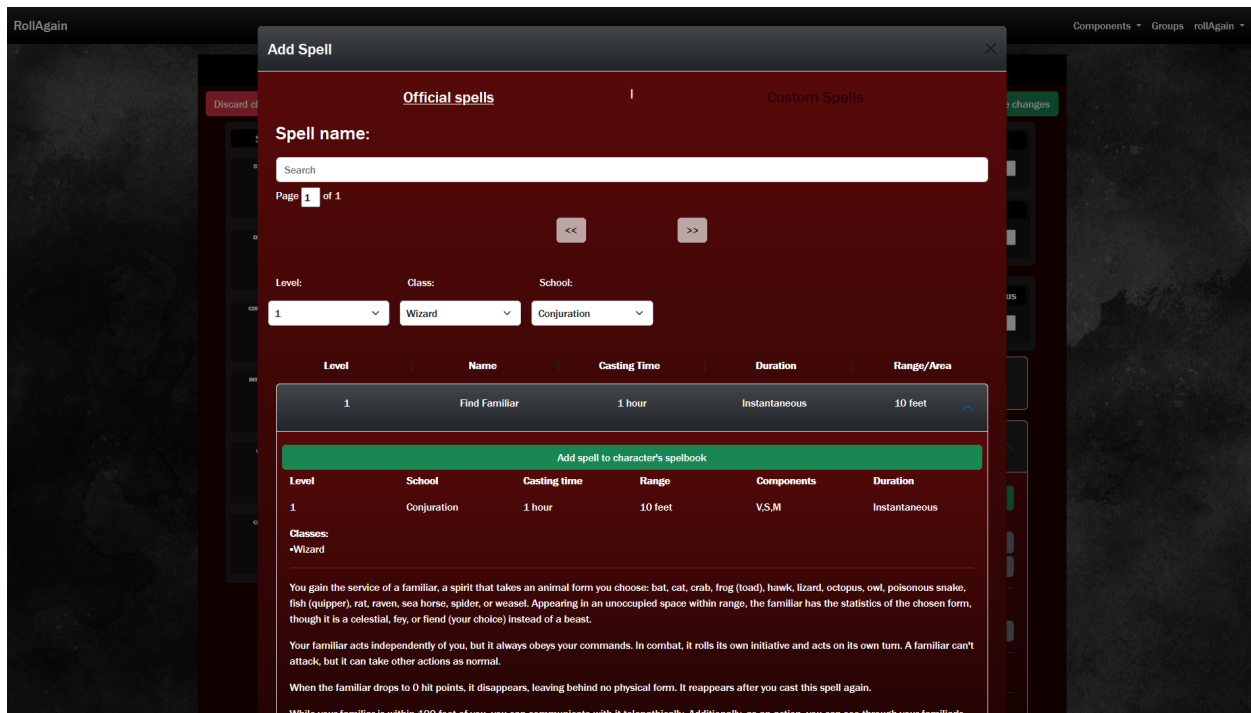


Figura C.13: Desplegable Añadir Componente a Personaje

También se podrá descargar la ficha de personaje en formato PDF mediante el botón en la esquina superior derecha.

Pantalla de componentes de grupo

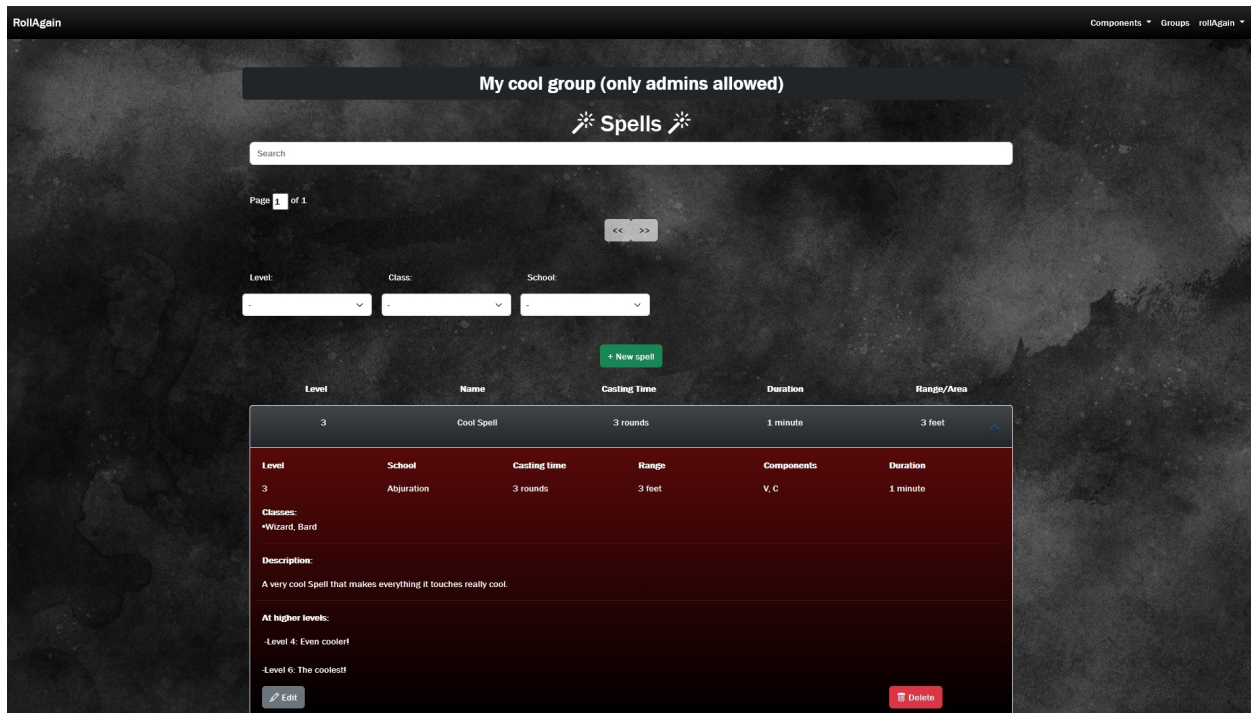


Figura C.14: Pantalla Final de Búsqueda de Componentes Personalizados

Similar a la pantalla de componentes oficiales, mostrará los componentes personalizados del grupo. Además, permitirá abrir un desplegable para crear nuevos componentes.

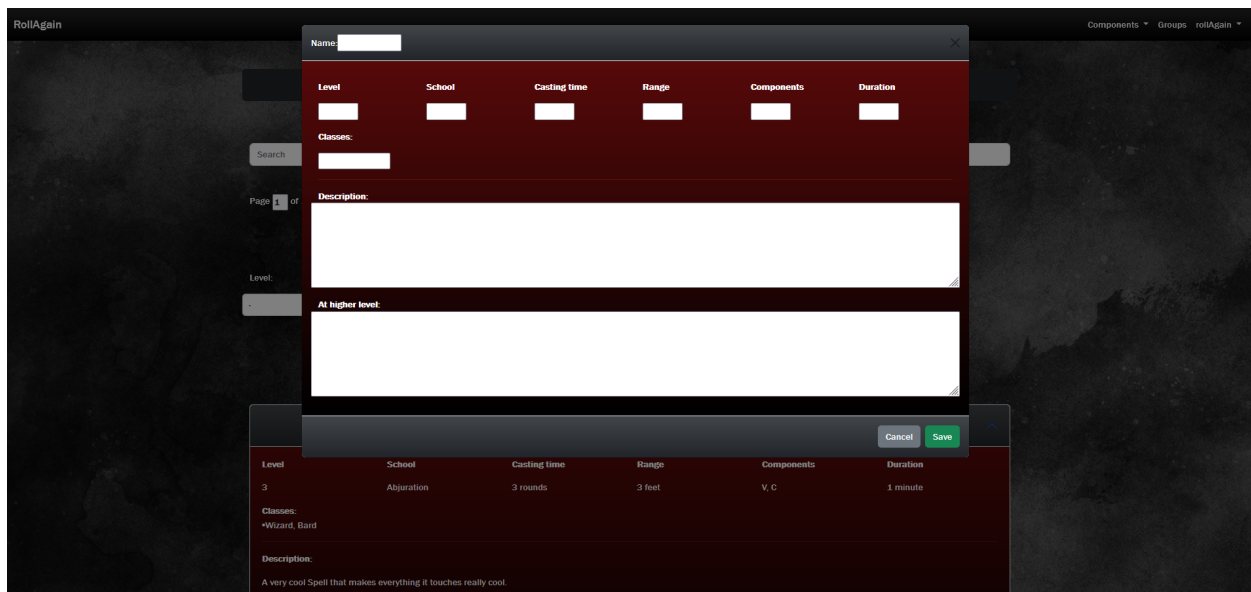


Figura C.15: Pantalla Final de Creación/Edición de Componentes

Los componentes ya existentes se podrán editar y eliminar con los botones al final de la descripción del componente.

Pantalla de sala de juego

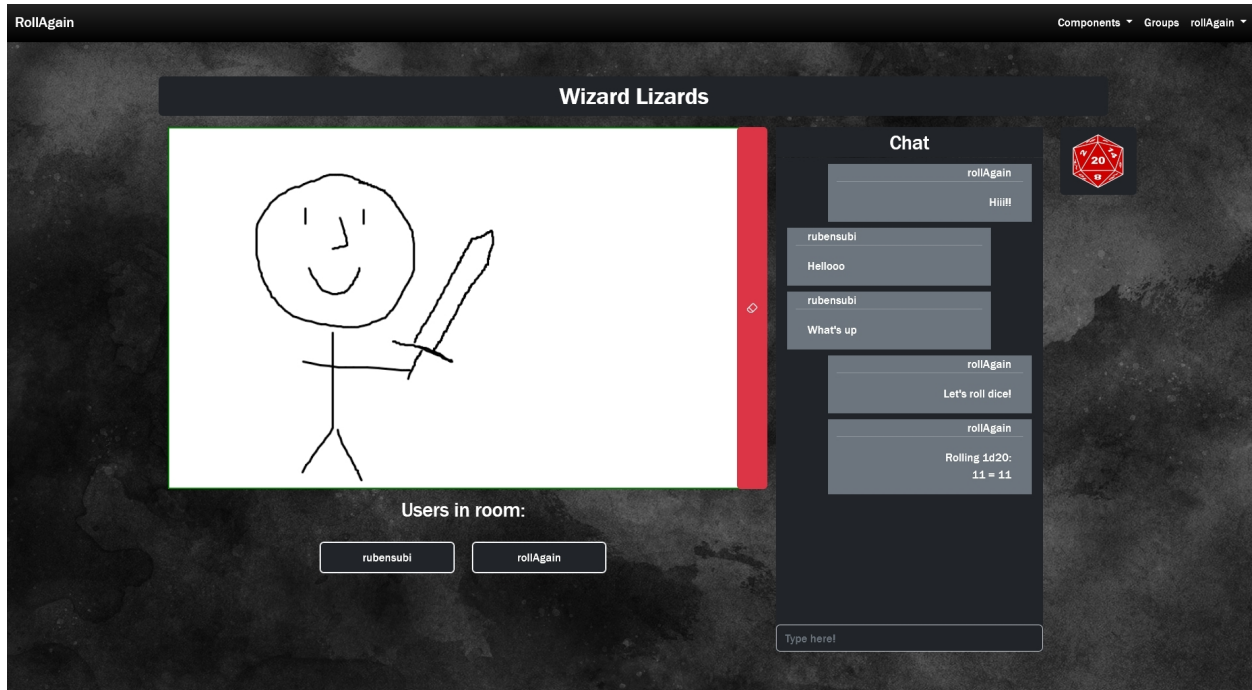


Figura C.16: Pantalla Final de Sala de Juego

Esta pantalla contendrá dos elementos principales: un tablero sobre el que los usuarios podrán dibujar arrastrando el click del ratón y un chat en el que podrán mandar mensajes.

Tanto los dibujos como los mensajes de chat serán visibles por todos los usuarios que estén conectados a la sala. Esta sincronización se ha realizado mediante el uso de WebSockets, de tal forma que cuando un cliente envía un evento, el servidor Web lo procesa y lo reenvía a los clientes Web del resto de usuarios conectados.

También se podrán hacer tiradas de dados mediante el desplegable en la esquina superior derecha. Los resultados de las tiradas se mostrarán en el chat.



Figura C.17: Desplegable Tiradas de Dados

Pantalla de wiki de grupo

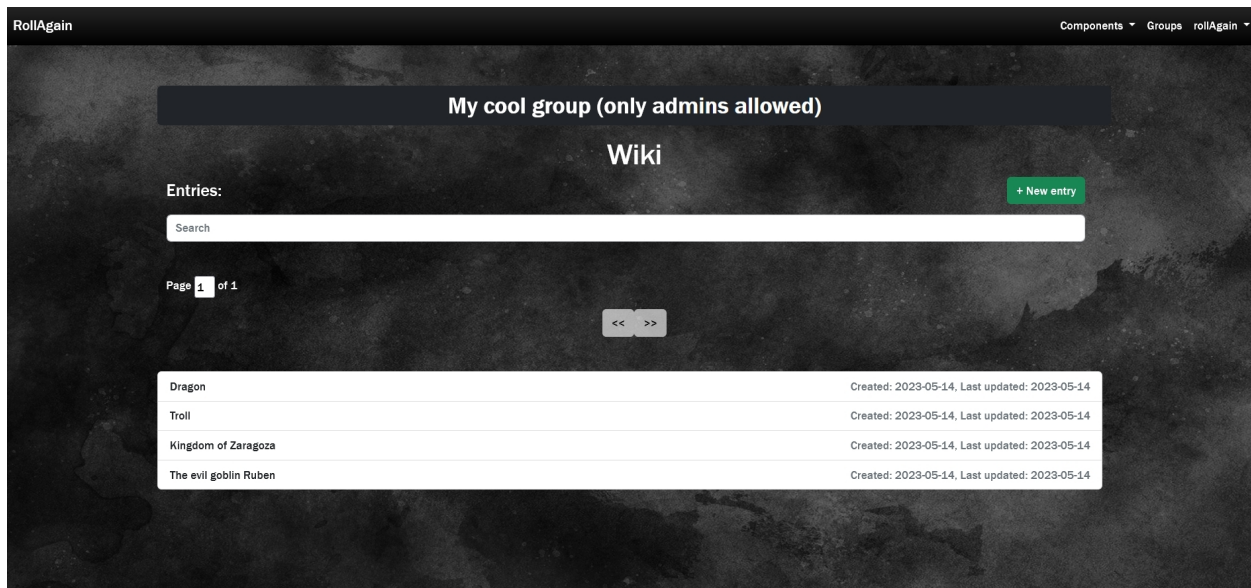


Figura C.18: Pantalla Final de Búsqueda de Entradas de Wiki

En esta pantalla se mostrará un listado de todas las entradas de la wiki de un grupo. El usuario podrá filtrarlas por sus nombres y ver la información de creación y edición de cada una de ellas.

Además, podrá crear una nueva entrada mediante el botón en la parte superior derecha. Al presionarlo, aparecerá una ventana emergente que solicitará al usuario un nombre para la nueva entrada.

Pantalla de entrada de wiki

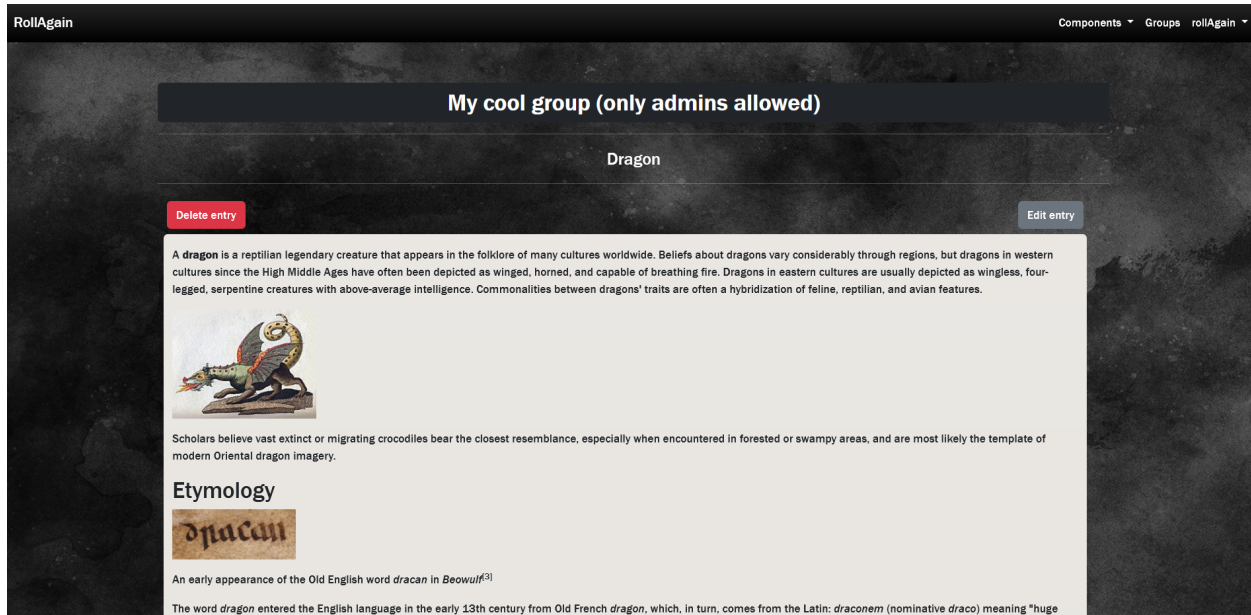


Figura C.19: Pantalla Final de Entrada de Wiki

La pantalla mostrará la información relativa a la entrada de la wiki creada por los integrantes del grupo, en forma de texto en formato HTML. Además, permitirá al usuario eliminar o editar la entrada con los botones situados en la parte superior.

En el modo edición, se mostrará un editor de hipertexto (se ha usado el editor *Quill*) que permitirá al usuario añadir la información que desee y guardar o descartar los cambios realizados.

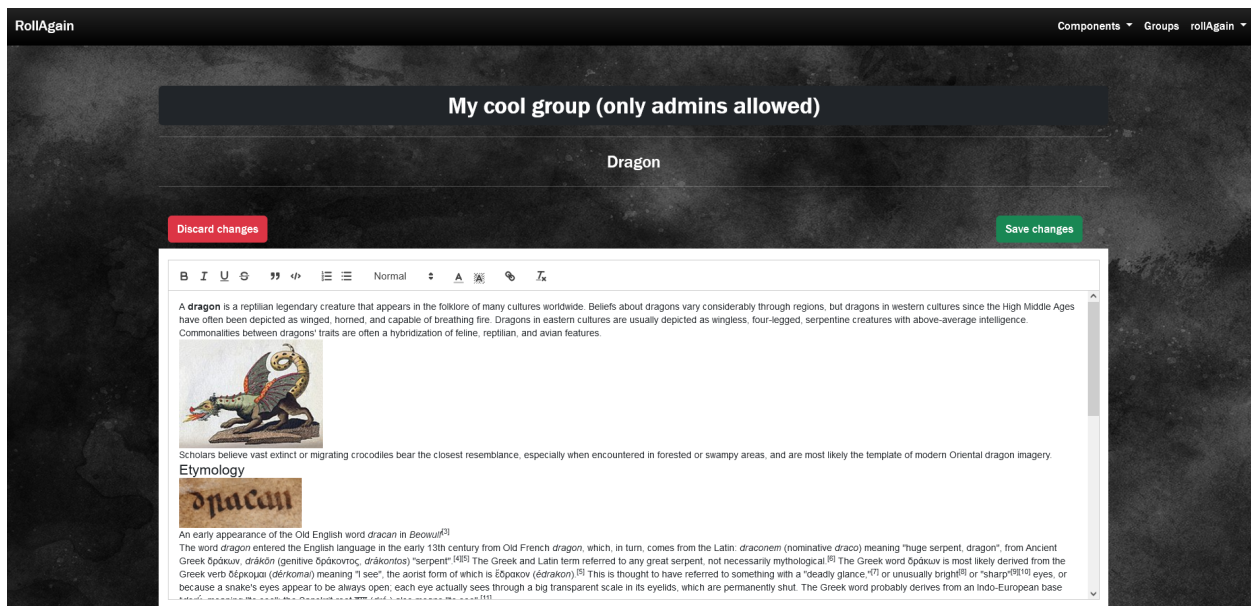


Figura C.20: Pantalla Final de Edición de Entrada de Wiki

Pantalla de lista de notas de *Master*

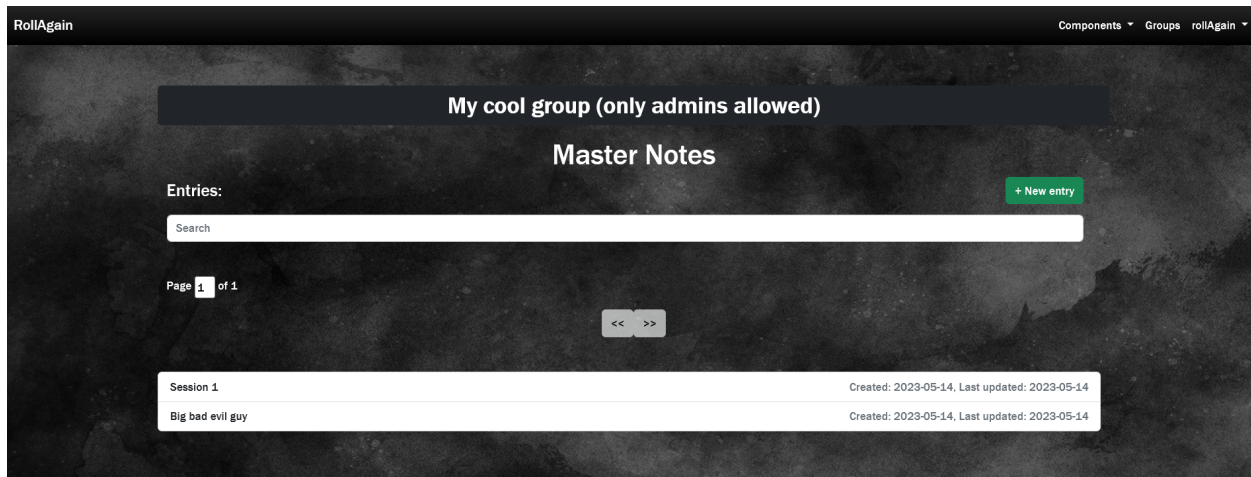


Figura C.21: Pantalla Final de Búsqueda de Notas de Master

Similar a la pantalla de entradas de wiki, contendrá un listado con todas las notas creadas por el *Master* del grupo y permitirá crear nuevas. Será accesible únicamente por el *Master* del grupo, y su utilidad es permitir almacenar información que el resto de jugadores no debe saber sobre las sesiones de juego. La pantalla de información de cada nota de *Master* actuará de manera similar a la de una entrada de wiki (de hecho, en el modelo de datos, una nota de *Master* es una entrada de wiki con un flag activado).