



Trabajo de Fin de Grado

Grado en Ingeniería Informática

Fantasy Teller, mejorando la experiencia rol

Fantasy Teller, improving roleplaying experience

Sebastián José Díaz Rodríguez

La Laguna, a 4 de septiembre de 2018

D. **Rafael Arnay del Arco**, con N.I.F. 78.569.591-G, profesor Ayudante Doctor de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor.

C E R T I F I C A

Que la presente memoria titulada:

"Fantasy Teller, improving the roleplaying experience"

ha sido realizada bajo su dirección por D. **Sebastián José Díaz Rodríguez**, con N.I.F. 43.379.824-F.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en San Cristóbal de La Laguna a 3 de septiembre de 2018.

Agradecimientos

Este Trabajo de Fin de Grado está dedicado a mi familia y a mi pareja, que han estado junto a mí, apoyándome para seguir hacia adelante en los buenos y los malos momentos.

También quiero dedicarlo a todas las personas de la comunidad de rol, con las que comparto esta gran afición día a día. Espero poder mejorar su experiencia y la mía a través de este gran esfuerzo.

Licencia



© Esta obra está bajo una licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional.

Resumen

El objetivo de este Trabajo de Fin de Grado es el desarrollo de una aplicación híbrida[1], es decir, una aplicación multiplataforma basada en herramientas de desarrollo web y recursos nativos de los dispositivos en los que es utilizada para llevar a cabo su cometido.

La aplicación en cuestión, denominada Fantasy Teller, presenta al usuario una colección de herramientas y recursos útiles para la dirección y gestión de partidas de juegos de rol de mesa. Las funcionalidades de la aplicación se basan en tres pilares conceptuales ampliamente utilizados al describir las acciones dentro de los juegos de rol: Exploración, Interacción y Combate, conceptos capaces de abarcar a grosso modo cualquier aspecto dentro de los escenarios de un juego de rol de mesa. Otro de los puntos que perfilan este trabajo es la consecución de un servicio robusto pero adaptable por el usuario para permitirle al mismo utilizar con fiabilidad la aplicación en cualquier sistema de juego.

Antes de adentrarnos en profundidad en el trabajo, revisaremos conceptos esenciales para comprender el origen del proyecto, así como el ámbito en el que éste tiene cabida y las razones de ello. Posteriormente, entraremos en comparativa con productos ya existentes y expondremos las mejoras que presentamos con la aplicación respecto a ellos. Una vez expuesto lo necesario para comprender de forma directa y concisa lo aplicado en el desarrollo de la aplicación, procederemos a los aspectos técnicos del proyecto, así como su fase organizativa previa y la metodología de desarrollo. Finalmente, se hará una reflexión sobre lo obtenido en el trabajo, las posibles mejoras y los próximos pasos del proyecto.

Palabras clave: aplicación híbrida[1], aplicación nativa, Google API[2], Mean Stack[3], Ionic[4], narración, juego, rol de mesa, Roll20[5], EpicGM[6], Google Play[7].

Abstract

The objective of this End of Degree Project is the development of a hybrid application[1], an application based on web development tools and native resources of the devices in which it is used to accomplish its goal.

The application in question, called Fantasy Teller, presents to the user a collection of useful tools and resources for the management and management of sessions of role-playing games. The functionalities of the application are based on three widely used conceptual pillars when describing the actions within the role-playing games: Exploration, Interaction and Combat, concepts capable of roughly covering any aspect within the scenarios of a tabletop role-playing. Another point that outline this work is the achievement of a robust but adaptable service for the user to allow him to use the application with reliability in any game system.

Before going deeper into the work, we will review essential concepts to understand the origin of the project, as well as the scope in which it has a place and the reasons for it. Afterwards, we will enter into comparisons with existing products and we will present the improvements we achieve and its features respect to the other products. Once exposed what is necessary to understand directly and concisely what is applied in the development of the application, we will proceed to the technical aspects of the project, as well as its previous organizational phase and the development methodology. Finally, there will be a reflection on what was obtained in the work, its possible improvements and the next steps of the project.

Keywords: hybrid application[1], native application, Google API[2], Mean Stack[3], Ionic[4], storytelling, game, table role, Roll20[5], EpicGM[6], Google Play[7].

Índice general

1. Introducción	1
1.1. Juego de rol de mesa	2
1.1.1. Participantes	2
1.1.2. Elementos conceptuales de un juego de rol de mesa	3
1.1.3. Pilares de juego	3
1.2. Aplicación híbrida	4
1.2.1. Diferencias respecto a una aplicación nativa	5
1.2.2. Otras opciones	5
1.3. Objetivos del proyecto	6
1.3.1. Objetivos específicos	6
2. Antecedentes	8
2.1. Contexto Social	8
2.2. Contexto Económico	9
2.3. Contexto Legal	10
2.4. Necesidades socio-económicas abarcadas por el proyecto	10
3. Metodología de desarrollo	12
3.1. Metodología Aplicada	12
3.2. Decisiones técnicas previas al desarrollo	13
3.3. Tecnología aplicada	14
3.3.1. MEAN Stack	14
3.3.1.1. MongoDB	14
3.3.1.2. ExpressJS	14
3.3.1.3. Angular	14
3.3.1.4. Node.js	15
3.3.2. Ionic Framework	15
3.3.3. API REST	16
3.3.3.1. Google Drive API REST	16
3.3.4. Correlación de las tecnologías aplicadas	16
3.4. Riesgos	17
3.5. Fuentes de información	18
3.6. Diagrama de Elementos	18
3.6.1. Estructuras y elementos principales	19
3.6.2. ModalService: Servicio de señales	21

3.6.3. UserService: Servicio de Usuario Google	22
3.6.4. DriveService: Servicio de Peticiones Drive	22
3.6.5. HomePage: Página de inicio	23
3.6.6. CampaignMenuPage: Página de partida	26
4. Resultados	31
4.1. Aplicación desarrollada	31
4.2. Correlación Pilares-Componentes	32
4.3. Guía de uso	33
4.4. Repositorio del proyecto	38
5. Conclusiones	39
5.1. Problema planteado	39
5.2. Solución aportada	39
5.3. Aprendizaje en el proceso	40
5.4. Futuras mejoras	41
5.5. Conclusiones	41
5.6. Conclusions	42
6. Presupuesto	43
Bibliografía	44

Índice de figuras

1.1. Boceto de la aplicación	1
3.1. Componentes de la Aplicación	19
3.2. Estructura JSON de campaigns	20
3.3. Home Page	25
3.4. Campaign Menu Page	29
4.1. Página Inicial	33
4.2. Creación de campaña y Acceso a Google	34
4.3. Página de selección de campaña	35
4.4. Ejemplo JSON de personajes y criaturas	36
4.5. Página de partida	38

Capítulo 1

Introducción

En este trabajo se ha desarrollado la aplicación híbrida[1] **Fantasy Teller**, destinada a ayudar en la organización y ejecución de partidas de juegos de rol de mesa, supliendo la falta de herramientas enfocadas al servicio del director de la partida. Esta se sirve de bases de datos en línea y de los **servicios de Google Drive** para gestionar los recursos de la partida, ya sean documentos, ficheros pdfs, imágenes o sonido ambiente, para dejar en manos del director la tarea de crear y usar más recursos.

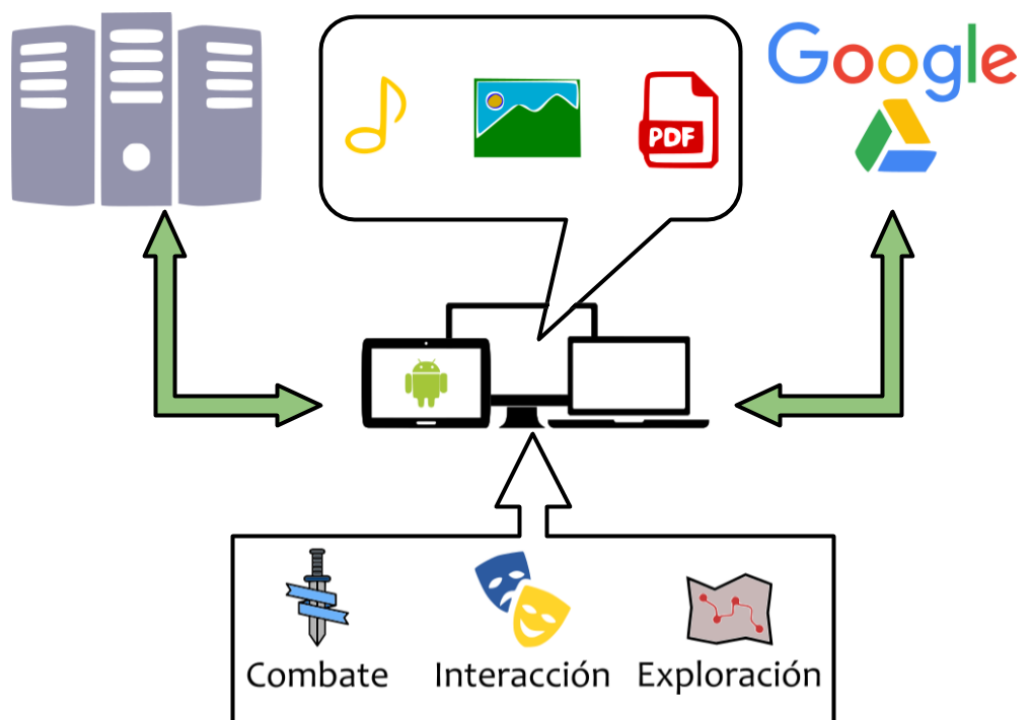


Figura 1.1: Boceto de la aplicación

Antes de entrar en materia, se expondrán una serie de conceptos ligados a la actividad, necesarios para comprender las tareas realizadas en el desarrollo de la aplicación y en sus casos de uso. Una vez se planteen estos conceptos, pasaremos a observar el entorno en el que se localiza la actividad actualmente, así como las razones por las que este proyecto se ha llevado a cabo. Los aspectos técnicos así como la metodología llevada a cabo continuará lo expuesto hasta el momento y, finalmente, realizaremos una reflexión sobre el producto logrado, así como de aspectos mejorables y los conocimientos aprendidos o fortalecidos en el proceso.

Un último punto a destacar, personalmente crucial para ubicar este proyecto en la realidad social, es que, partiendo de que la premisa de este proyecto es crear una aplicación, como tal, el objetivo final propio es la consecución de un producto publicable en tiendas oficiales, con la posibilidad y meta en mente de actualizar sus contenidos así como añadir aspectos no considerados durante el desarrollo del proyecto.

1.1. Juego de rol de mesa

Un juego de rol de mesa es un juego de interpretación en el que los jugadores toman el papel de unos personajes dentro de un hilo narrativo. Estos realizan diálogos entre ellos y otros personajes ficticios y describen sus acciones con el objetivo de avanzar en la narración, sujeta únicamente por las decisiones que los jugadores toman basándose en la personalidad, rasgos físicos y rasgos culturales con los que han definido al personaje que interpretan. Debido al peso de las decisiones de los jugadores, la imaginación, la narración oral, la originalidad y el ingenio son primordiales para el adecuado desarrollo de esta forma dramática.

Al contrario que en las películas, las obras de teatro o las series, en un juego de rol de mesa no existe un guión prefijado, sino que la historia, como citamos anteriormente, avanza según la expresión de los personajes dada a través de las decisiones de los jugadores. Estas decisiones y la toma de acciones se ven afectadas por el azar, lo que permite que la historia sea flexible y no caiga en la previsión pura de los jugadores, como suele ocurrir para los espectadores de una serie o de una película.

Sin embargo, normalmente, el poder del azar puede interferir en el disfrute de esta actividad, por lo que existe el papel del director de partida, a quién se le delega la tarea de definir hasta qué punto el azar toma papel dentro de una escena y las normas del juego aplicadas a las acciones que desean realizar los jugadores.

Un aspecto a destacar de estos juegos es que no existe una meta objetiva de victoria. No hay una condición en la que se gane el juego y, como en el resto de expresiones dramáticas, lo que busca es comunicar una historia.

1.1.1. Participantes

Para mayor énfasis en las diferencias que separan conceptualmente a los jugadores del director de partida, podemos definir más concisamente las tareas que realizan cada uno.

En el caso del jugador, su mayor labor es la de permitir el desarrollo de la historia sin bloquear o perjudicar el disfrute del resto de jugadores y del director. Se podría decir que, como tal, el jugador participa en el juego con la preparación necesaria para disfrutar del mismo.

Por su lado, el director de partida se encarga de tareas más allá de la participación en partida. Para lograr un avance significativo en la partida, debe de conocer las normas del sistema que aplican para crear personajes y definir sus acciones, y debe preparar el conflicto narrativo en el que se verán envueltos los jugadores y que llamará a que éstos actúen en el mundo que ha desarrollado, así como materiales que promuevan la diversión durante la partida.

1.1.2. Elementos conceptuales de un juego de rol de mesa

Los jugadores interpretan, dentro del juego, a unos personajes conocidos como Personajes Jugables o en inglés Playable Characters. Definiciones homónimas de este papel son los Personajes Jugadores, en inglés Player Character, abreviado como PC. Este último será el término aplicado durante la exposición del trabajo.

Los personajes interpretados por el director de partida son denominados como Personajes No Jugables (Non Playable Characters) o Personajes No Jugadores (Non-Player Character), abreviado como NPCs. Al igual que en el caso de los personajes de los jugadores, el término NPC será el más utilizado a la hora de exponer el trabajo.

En cuanto a las partidas, conocidas comúnmente como sesiones, pueden formar agrupaciones que dan lugar a otras estructuras narrativas. Así pues, una sesión autoconclusiva y desligada del entorno en el que se realiza es conocida como un one-shot adventure, término inglés que tomamos prestado para hablar de esta modalidad, así como una colección de sesiones no autoconclusivas con un desarrollo propio de la trama es conocido como una aventura.

Finalmente, una colección de aventuras tiene el nombre de campaña. Esta última es de mayor complejidad que las aventuras por separado ya que puede presentar conexiones narrativas entre aventuras no consecutivas y suele permitir un desarrollo más profundo de los personajes interpretados por los jugadores y el director de partida.

1.1.3. Pilares de juego

Entrando más en profundidad en el concepto de los juegos de rol de mesa, sus bases pueden interpretarse de muchas maneras, tomando como pilares del juego unos conceptos u otros. El propio sistema puede adoptar una definición propia de estas bases para explicar a sus usuarios qué significa cada elemento dentro de una escena o dentro de una partida. En el proyecto, los pilares del juego vienen determinados según las bases presentadas por el juego Dragones y Mazmorras[8], en el que se distinguen tres pilares generales de los que pueden partir cualquier elemento narrativo dentro de una partida. Estos se definen como los pilares de Exploración, Interacción y Combate.

El pilar de la **Exploración** tiene sus bases en el descubrimiento, visita y descripción de un lugar dentro de un mundo real o ficticio introducido a los jugadores mediante los personajes ficticios que interpretan. Dentro del espectro descriptivo no solo se usa la narración propiamente dicha, ya que pueden vincularse otros sentidos para complementar el lugar, como música ambiente, objetos físicos e incluso olores, dispuestos con el atrezzo, vídeos, imágenes y mapas dispuestos por el director de la partida. Con la exploración introducimos al usuario en el mundo en el que queremos narrar la historia.

El pilar de la **Interacción** abarca todo lo referido al intercambio de información con los personajes no jugables y la influencia de los jugadores en las sociedades del mundo en el que son introducidos. Este intercambio se realiza entre jugadores y director mediante los personajes que interpretan, dotándolos de rasgos físicos y mentales y de información vinculada a la posible vida que han tenido previa a su introducción en la escena.

Este pilar implica, por parte del director, la gestión de los personajes no usados por los jugadores y la historia en la que se desarrollan los hechos, así como la generación de vías argumentales para continuarla, de lo que también deriva la generación rápida de personajes e historia para mantener el ritmo de la narración en situaciones no preparadas previamente.

El pilar de **Combate** se centra en la gestión de un grupo de agentes que entran en conflicto con los personajes interpretados por los jugadores. Esta gestión conlleva el marcado de las estadísticas por separado de los agentes y personajes que se incluyen en la escena, así como sus posibles acciones y comportamientos en un entorno hostil. Aunque siendo el más fácil de explicar, es un pilar más técnico debido a la gestión de la información de la que debe disponer el director y las facilidades que se le entregan mediante las herramientas de la aplicación para que el combate no ralentice el avance de la partida.

1.2. Aplicación híbrida

Las aplicaciones híbridas [1] son aplicaciones móviles diseñadas en un lenguaje de programación web ya sea HTML5, CSS o JavaScript, junto con un framework que permite adaptar la vista web a cualquier vista de un dispositivo móvil. Son aplicaciones construidas para ser utilizadas o implementadas en distintos sistemas operativos móviles, lo que nos permite evadir la tarea de crear una aplicación para cada sistema operativo por separado y otra para disponer de sus funcionalidades en web. De esta manera, una aplicación híbrida puede ser adaptada a múltiples plataformas móviles sin crear nuevos códigos, sino adaptando ciertas partes del mismo para acceder a los recursos únicos de cada sistema operativo.

A pesar de que el desarrollo de aplicaciones híbridas y nativas requiere de una construcción totalmente distinta, en los dispositivos móviles su instalación y uso son el mismo. Accediendo a nuestro centro de aplicaciones podemos descargarlas e instalarlas de la misma forma. En el caso de las aplicaciones web, solo tendríamos que acceder a su portal para poder usarlas.

1.2.1. Diferencias respecto a una aplicación nativa

La diferencia entre aplicaciones nativas y aplicaciones híbridas es, visualmente, relativa. En principio las aplicaciones híbridas se atienen a las tecnologías web para presentar una interfaz sin los mismos recursos y control de vistas que ofrecen los sistemas nativos, sin embargo, estos comportamientos pueden ser alcanzados sino complemente, de manera muy similar. Esta diferencia de cara a la experiencia de usuario no es única. En el punto anterior ya se expresó la diferencia en cuanto a las tecnologías de desarrollo y las prestaciones a las que podemos acceder desde estos dos enfoques.

Debido al lenguaje empleado, una aplicación nativa emplea todos los recursos de los que dispone en ámbito del dispositivo, mientras que la híbrida aplica lenguajes web para poder funcionar mediante un buscador sobre la plataforma. Esta misma diferencia afecta al rendimiento percibido por el usuario, y es que al no poder acceder a los recursos directamente, las aplicaciones híbridas se ven limitadas por la propia potencia de los buscadores en los que son lanzadas.

Haciendo a un lado las especificaciones técnicas, se pueden observar diferencias claras en el proceso de desarrollo. Realizar una aplicación nativa conlleva un mayor tiempo de desarrollo, no solo por la necesidad de generar códigos diferentes para un misma aplicación si se quiere alcanzar un nicho de mercado mayor, sino también por la flexibilidad de los lenguajes web y la facilidad con la que se puede adentrar el programar en ellos, lo que permite un tiempo de desarrollo menor a favor de las aplicaciones híbridas.

Otro de los aspectos que pone en conflicto el usar un tipo de aplicación u otra es que las aplicaciones híbridas no presentan de base la interfaz de usuario propia del dispositivo, así como una posible respuesta lenta debido a la conexión con los servidores que sostienen la aplicación, y esto puede refrenar al usuario medio al no estar familiarizado con la misma.

1.2.2. Otras opciones

De las herramientas aplicadas en el proceso de creación de una aplicación multi-plataforma como la que hemos hecho, podemos deducir otras opciones viables, como son la creación de una página web o una aplicación a partir de la plataforma Unity. Estas opciones fueron descartadas del proyecto por diversos motivos. En el caso de la página web, la incapacidad de acceder a los recursos propios del dispositivo en el que se ejecuta y la puesta en marcha a través de una página web genérica respaldaron la mejor adaptación al medio por el lado de la aplicación híbrida.

La plataforma Unity[9] es bastante conocida por la facilidad que aporta a la hora de desarrollar videojuegos en 2D y 3D. Uno de los aspectos que ofrece como parte de este desarrollo es la creación de interfaces interactivas y entre sus recursos permite conexiones a través de la red, además de poder consumir las API[10] REST[11] de Google. Es una plataforma capaz de llevar a cabo todas las actividades previstas y mucho más, lo que deriva en el uso de una plataforma mucho más potente que la necesaria para generar una aplicación con vistas en web.

1.3. Objetivos del proyecto

1. Desarrollar un producto que suponga un nuevo enfoque para la comunidad existente.
2. Aprender el proceso de desarrollo de una aplicación híbrida desde la base organizativa hasta la producción y publicación.
3. Aprender sobre las herramientas de desarrollo web comunes utilizadas en la actualidad.
4. Aprender sobre el funcionamiento de REST APIs.

En resumen, el proyecto tiene por objetivo el desarrollo de una aplicación híbrida así como el aprendizaje de las herramientas establecidas como el estándar en esta comunidad. El desarrollo de la aplicación Fantasy Teller como Trabajo de Fin de Grado ha tomado por meta el ofrecer una aplicación sólida, funcional y libre para la dirección de partidas de rol de mesa, dotando así a la creciente comunidad establecida en las islas de una herramienta versátil y adaptable a sus necesidades que fomente la participación presencial en esta afición, y al mismo tiempo, expandir los conocimientos desempeñados en algunas asignaturas del propio grado.

1.3.1. Objetivos específicos

Dentro de esta clasificación de actividades a realizar para el desarrollo de la aplicación entendemos el manejo de información como la petición de la misma al servidor web y su consiguiente petición de información a la cuenta de Google Drive del usuario.

1. Implementación sección exploración
 - a) Implementación de la lógica
 - 1) Funcionamiento de galería visual
 - 2) Música ambiente
 - b) Implementación de la interfaz
 - 1) Visualización
 - 2) Edición de contenido

2. Implementación sección combate

a) Implementación de la lógica

- 1) Manejo de entradas de información de criaturas
- 2) Roster - Listado de jugadores con información
- 3) Iniciativa - Lista ordenada de jugadores y criaturas en combate

b) Implementación de la interfaz

- 1) Visualización
- 2) Edición de contenido

3. Implementación sección interacción

a) Implementación de la lógica

- 1) Manejo de entradas de información del módulo de rol
- 2) Manejo de entradas de información para personajes
- 3) Generadores de información adicional

b) Implementación de la interfaz

- 1) Visualización
- 2) Edición de contenido

Capítulo 2

Antecedentes

2.1. Contexto Social

El comienzo de los juegos de rol se remonta al siglo dieciséis, con la comedia dell'arte[12], grupos teatrales que representaban obras improvisadas y prosiguen hasta el siglo veinte en forma de juegos de mesa de carácter detectivesco o histórico. No es hasta los sesenta que se inicia una introducción de la ambientación medieval fantástica a estos juegos, y se realiza por primera vez en la Sociedad por el Anacronismo Creativo. Por su propia vertiente, los juegos de guerra (wargames[13]) evolucionan del padre del ajedrez, el Catur[14], a juegos como Little Wars[15] y Diplomacy[16]. Es de estos juegos de los que sale el concepto de miniatura, una figura representativa de un grupo de unidades dentro de un marco de batalla.

No es hasta 1974 que se da inicio a la venta de los juegos de rol de mesa icónicos de esta afición, como Dungeons & Dragons[8] por la parte de los juegos de fantasía medieval, o Metamorphosis Alpha[17] para la ambientación de ciencia ficción. Desde este punto, los juegos de rol se diversifican, siguiendo líneas narrativas de terror, como el caso del conocido Call of Cthulu[18], y de investigación y espionaje, como James Bond 007 RPG[19].

En el año 2014, con la salida de la quinta edición de Dungeons & Dragons[8], un sistema nuevo, simplificado y limpio, construido en parte con las críticas de la comunidad, los juegos de rol de mesa recibieron muchos más integrantes, en el momento ajenos a la actividad. La expansión de estos juegos dio un boom gracias a la emisión de las sesiones a través de canales multimedia como Youtube y, sobre todo, Twitch. Este fenómeno vigente ha logrado una difusión a nivel mundial nunca antes experimentada en el sector que no parece aflojar fuerzas en la sociedad.

Para entender el impacto del juego de rol de mesa en el entorno tinerfeño primero se debe exponer la evolución de estos juegos en el entorno nacional, el cual se desarrolló asemejándose a la experiencia extranjera relativamente lento. Y es que su asentamiento en la nación no se produce hasta el reflote de los juegos de rol de mesa, cuando ya en Estados Unidos habían pasado por una época transitoria de deterioro.

Por el dos mil, la situación de los juegos de rol de mesa era ambigua, mientras que las editoriales de juegos cerraban, la comunidad no hacía más que aumentar y los encuentros surgían con mayor naturalidad. En la actualidad, existen juegos editados en España que relatan parte de nuestra historia en la ambientación que poseen, como es el caso de 1808[20], un juego de rol basado en la época de la Guerra de la Independencia Española, o que son de conocimiento mundial, como es el caso de Aventuras en la Marca del Este[21].

Uno de los factores cruciales en la expansión de los juegos de rol de mesa en España fue la crisis económica que sufrió el país durante los últimos años, época en la que las personas empezaron a interesarse más por estas actividades relativamente asequibles en comparación a otros servicios, ya que su carácter relativamente novedoso le permitió no verse tan afectado por las expectativas pesimistas del estado económico familiar.

Estos últimos años se ha convertido en la comunidad española y más notablemente la comunidad tinerfeña en una actividad predilecta entre jóvenes y adultos hasta formarse en la isla comunidades dedicadas a la promoción de actividades para dar a conocer sistemas de juego diferentes y enseñar en teoría y práctica el desarrollo de la afición.

2.2. Contexto Económico

En el ámbito económico, la época de crisis comentada en el anterior punto también supuso la conversión de las tiendas dedicadas a la venta de estos juegos. Así fue como se formaron muchos de los movimientos actuales dedicados a los juegos de rol de mesa, así como los juegos de cartas intercambiables y los juegos de mesa en general. La dependencia parcial de estos eventos ha producido una economía estable para las tiendas ya asentadas, pero una limitación para la creación de nuevos negocios. La conducta que caracteriza a los usuarios de estos juegos y, por tanto, los legítimos usuarios de la aplicación desarrollada, es la facilidad para acostumbrarse y la petición continua de entretenimiento sencillo y obtenible sin mucho esfuerzo.

El formato de venta de las aplicaciones ya existentes marca otra pauta a seguir en cuanto al método en el que la aplicación debe ser promocionada y publicada. En el mercado actual se oferta normalmente una base gratuita a la que se le pueden añadir módulos externos mediante un micropago. Otra pauta es la del pago de cuotas para el acceso a una serie de características avanzadas, como ocurre en algunos de los productos de referencia del proyecto de los que se hablará más tarde. Gracias al carácter inicialmente gratuito del proyecto esto no limita el comportamiento previsto o la adición de mecanismos de publicidad que afecten al producto que presentamos.

En definitiva, el contexto económico favorece el desarrollo de este tipo de aplicaciones, aunque a su vez presenta una gran cantidad de competidores a los que nos enfrentamos favorablemente por el enfoque dado al proyecto, la destinación de los recursos al director de partida en vez de a la partida en sí o al jugador.

2.3. Contexto Legal

Los aspectos legales que afectan directamente al desarrollo y publicación de la aplicación en mercado son la ley orgánica de protección de datos y las normas vigentes sobre los derechos de autoría de las empresas distribuidoras de contenido.

Debido a la utilización de una cuenta de usuario propia para la aplicación, almacenada en la base de datos en línea, y el uso y acceso a datos personales del usuario para el disfrute de todas las capacidades de la aplicación mediante su propia API, la aplicación de la Ley Orgánica de Protección de Datos es obligatoria, un aspecto tratado una vez se proyecte la producción para el mercado del proyecto. Bajo esta ley no se pueden divulgar los datos personales dispuestos por el usuario, así como utilizar los mismos en su contra o como un medio de producción económico.

Sobre los derechos de autoría de las obras presentadas dentro de la aplicación, ninguno de los datos de base dentro de la misma estarán sujetos a esta legislación, bien sea por ser de carácter propio o por la regulación sobre los archivos almacenados por parte de Google. Si bien, se advierte al usuario de que todo contenido implementado dentro de la misma está sujeto a lo antes mencionado.

2.4. Necesidades socio-económicas abarcadas por el proyecto

Durante la explicación del contexto en el que nos ubicamos para suplir una necesidad social se ha resaltado el auge de la tecnología en esta afición con materiales tan simples como el papel, lápices y libros. Los juegos de rol de mesa ya han establecido su lugar en la producción de entretenimiento multimedia en canales como Youtube y Twitch, pero no solo se sirven de estos canales para aprovechar las características de automatización y comunicación que ofrece la tecnología de nuestro siglo. Ya es un elemento común en las partidas la utilización de música de ambientación ya sea mediante un ordenador o un equipo de sonido particular. La inclusión de proyectores para mostrar imágenes y posicionar mapas en paredes o mesas no es un factor extraño, y la utilización de realidad virtual parece acercarse a este ámbito cada vez más debido a los proyectos de realidad aumentada que la propia comunidad ha iniciado para la proyección de miniaturas digitales durante las partidas.

El mayor peso para el director de partida, nuestro usuario, es la gestión de grandes cantidades de información interrelacionada y por su parte unida a material multimedia, lo que hace de la tarea de desarrollar una buena narrativa y un juicio de las normas justo un desafío increíble. Es por ello que la propia comunidad empezó hace años proyectos software para unificar las fuentes de información así como organizar los recursos extraídos con simplicidad y rapidez. Esta meta es compartida entre nuestro proyecto y las aplicaciones que se han tomado como referencia para el proyecto: Roll20[5] y EpicGM[6].

El motivo principal por el que se ha elegido desarrollar una aplicación del estilo es por la falta de un programa de apoyo para dirección de partidas con enfoque único en este aspecto. La mayoría de aplicaciones en mercado que benefician el desarrollo de esta afición están destinadas a la generación de contenido narrativo pseudoaleatorio, la exposición de referencias de normas de juegos determinados o referencias a normas y contenido para los jugadores. Esta gran mayoría de aplicaciones, además, suele centrarse en un único juego, lo que limita el alcance potencial del software y la propia libertad del usuario sobre el contenido de sus programas. Por otra parte, una breve minoría tiene por objetivo la gestión en general de una partida. Estos casos reducidos han servido como inspiración para la creación conceptual de Fantasy Teller y de estos parten los errores que se pretenden solventar con la aplicación.

Si bien en el caso de Roll20[5] se ve incluida la dirección de partidas, su objetivo es sustentar la partida al completo, incluyendo la comunicación entre jugadores y director, la gestión de la información y el registro de las acciones llevadas a cabo por los participantes. Es una aplicación con una comunidad asentada con pagos para acceder a características específicas así como contenido oficial de los juegos de rol más conocidos. La mayor falla que presenta es la ralentización que sufre en ciertos servicios como la música ambiental y la apertura de contenidos como varias hojas de personaje debido a la gran cantidad de aspectos que pretende mantener en un mismo instante y la forma de conexión de la que disponen para incluir en una misma zona de juego a jugadores y a director.

EpicGM[6], por su parte, es una aplicación que comparte el objetivo con nuestro proyecto, aunque con el principal defecto de una interfaz estática que, aunque estilizada, no permite mucha libertad de movimiento para el director de la partida, su contenido estandarizado para la quinta edición del juego de rol de mesa Dragones y Mazmorras limita mucho su uso, y el ciclo de vida entre interfaces provoca fallos que bloquean la aplicación.

Por nuestra parte, como ya hemos comentado, ofrecemos una aplicación gratuita enfocada a una necesidad real dentro de la comunidad local y nacional formada entorno a los juegos de rol de mesa, con una mayor flexibilidad en el registro de contenido y la manipulación del mismo, así como libertad organizativa y visual de los contenidos antes, durante y después de las sesiones de cada juego. La solución que ofrecemos a través del producto creado para el estudio de las aplicaciones híbridas pretende suplir la inconsistencia del ciclo de interfaces y dejar en manos del usuario qué información utiliza y de qué forma en todo momento.

Capítulo 3

Metodología de desarrollo

Ya hemos definido los elementos que compondrán la aplicación con una aproximación más cercana al usuario que al propio programar. Es en este apartado en el que nos centraremos en las bases técnicas del proyecto, planteando primero la metodología de desarrollo llevada a cabo, seguida de las decisiones técnicas tomadas como base para la proyección del trabajo hacia una aplicación multiplataforma destinada a web y a dispositivos android.

Las condiciones técnicas establecidas precederán la descripción de la tecnología software utilizada para desarrollar la aplicación, unificada con la definición de sus utilidades dentro del proyecto. Finalmente, repasamos los posibles riesgos del proyecto, los problemas previstos y los problemas añadidos durante el desarrollo y las herramientas que utilizados como fuentes de información de la aplicación.

3.1. Metodología Aplicada

La metodología utilizada para organizar el avance del proyecto se ha basado en iteraciones continuas de desarrollo compaginadas con reuniones bisemanales con el tutor de proyecto. Esta metodología se complementa con una filosofía de ciclos, con los que se van turnando los recursos y esfuerzos de desarrollo pasando de un pilar conceptual a otro según se fueron obteniendo resultados. De esta forma, la aplicación ha obtenido un balance de complejidad entre los tres apartados descritos en la introducción (Exploración, Interacción y Combate), así como un contenido en herramientas aceptable para iniciar una campaña de juego.

La investigación fue destinada a la recolección de información sobre los recursos necesarios para la programación de una aplicación híbrida multiplataforma, aplicando parte de las herramientas de MEAN Stack y el entorno de desarrollo Ionic.

A partir de este punto, el desarrollo siguió con la codificación de cada pilar de juego, intercalado por sesiones de revisión del proyecto para establecer los puntos concluidos y los siguientes objetivos. La utilización de esta metodología ha permitido la agregación de opiniones sobre el proyecto, así como la redirección del mismo en casos en los que el desarrollo se ha enfocado equívocamente.

Además, ha permitido la revisión de los contenidos logrados y un progreso balanceado entre las distintas partes. Fue elegida por su flexibilidad y ha demostrado ser una metodología práctica para este tipo de trabajos de menor escala y reflexión continua.

3.2. Decisiones técnicas previas al desarrollo

Previo al estudio de las tecnologías asociadas al proyecto, hubo una fase aún más atrás en el tiempo en el que se evaluaron las posibilidades de hacer una aplicación nativa a través del entorno Android Studio[22]; sin embargo, la alta complejidad que suponía la inicialización y uso de ciertos aspectos básicos en la aplicación así como la próxima obsolescencia de otras funcionalidades aplicadas durante el desarrollo provocaron la redirección del proyecto hacia el entorno más flexible y simplista. Esta decisión fue tomada de forma tardía aunque no sin producir frutos en forma de una experiencia mayor a la hora de planificar un producto software.

A la hora de ubicar la aplicación en el mercado, se establecieron dudas respecto a los sistemas operativos para los que producir la aplicación. Esta duda surgió de la imposibilidad de reproducir una misma función en los sistemas operativos Android y Mac OS X a la vez, en específico la proyección de música y archivos pdf, debido a la necesidad de utilizar recursos nativos del dispositivo para ello. Por esta razón se tomó la decisión de producir para plataformas web y plataformas Android. Más específicamente, respecto a los dispositivos Android, si bien existen los smartphones y las tablets, la aplicación va destinada para estas últimas, debido al tamaño de la pantalla y de su manejo más intuitivo del menú que hemos planteado para la aplicación.

La última decisión técnica que ha perfilado el desarrollo del proyecto ha sido la necesidad de una cuenta de Google por parte del usuario de la aplicación, de la cual ya dispondrá en caso de hallarse usando un dispositivo Android. Esto viene acompañado de la utilización de la interfaz actualizada de autorización OAuth2, implementada para el acceso a la información de los datos del usuario a través de las API REST de Google.

Con esta medida, migramos el mayor peso de la información almacenable del servidor de acceso de usuarios al espacio personal del usuario, lo que nos permite manejar únicamente referencias desde el lado del servidor y aplicar éstas a las funciones integradas en la aplicación.

3.3. Tecnología aplicada

3.3.1. MEAN Stack

MEAN Stack es un conjunto de subsistemas que permite el desarrollo de aplicaciones web mediante lenguajes de programación conocidos y fáciles de comprender. Los componentes de la pila MEAN son MongoDB, una base de datos NoSQL, Express.js, un marco de aplicación web que se ejecuta en Node.js, Angular, framework JavaScript MVC que se ejecutan en los motores de JavaScript del navegador, y Node.js, un entorno de ejecución para aplicaciones de red y servidor controladas por eventos. Todos son proyectos de código abierto y uso gratuito, de los cuáles no se ha utilizado MongoDB debido al apoyo de Google Drive para la conservación de la información.

3.3.1.1. MongoDB

MongoDB es una base de datos documental flexible y escalable aunque no tan robusta como las bases de datos relacionales. Este tipo de bases de datos está constituida por un conjunto de programas que almacenan, recuperan y gestionan datos de documentos o datos de algún modo estructurados, lo que significa que los campos pueden variar de un documento a otro y la estructura de datos se puede cambiar con el tiempo. Por ello, este tipo de bases de datos normalmente no soportan operaciones JOIN, ni garantizan completamente ACID, y habitualmente escalan bien horizontalmente.

3.3.1.2. ExpressJS

ExpressJS es un framework de desarrollo de aplicaciones web para NodeJS inspirado en el Lenguaje de Dominio Específico Sinatra para Ruby. Comparte con éste su simplicidad y flexibilidad, además de la facilidad con la que integra a su comportamiento básico funcionalidades potentes a través de plugins. Está diseñado para construir aplicaciones web y APIs y se ha convertido en el entorno de desarrollo estándar para el lado del servidor en las aplicaciones basadas en NodeJS.

3.3.1.3. Angular

Angular es otro framework de desarrollo de aplicaciones web destinado al desarrollo del front-end. Está basado en el lenguaje Typescript, un superset de ECMAScript 6 con compatibilidad hacia atrás, lo que permite ser directamente compatible con JavaScript, permite el tipeado estático, programación orientada a objetos, el uso de lambdas y iteradores y facilita la reflexión.

Por su parte, Angular elimina el concepto de alcance (scope) y promueve la generación de una jerarquía de componentes como base de su arquitectura. Estos componentes, independientes los unos de los otros, permiten la modularización de la información, su tratado y su exposición. Este proyecto ha sido desarrollado desde 2016 por Angular Team de Google y por la comunidad, gracias a que se mantiene como un proyecto de código abierto, y ya ha alcanzado la versión 6 de su desarrollo.

3.3.1.4. Node.js

Node.js es un entorno multiplataforma, de código abierto, basado en JavaScript que permite ejecutar código JavaScript por el lado del servidor. Esto supone una filosofía de desarrollo en la que se puede lanzar código JavaScript en cualquier parte de la aplicación o página web, en vez de embeber fragmentos de este código en partes específicas en la que históricamente se requería. El mayor logro de Node.js ha sido el de unificar la construcción de una aplicación web en un único lenguaje de programación tanto en el lado del servidor como en el del cliente. A través de Node.js se conectan la mayoría de tecnologías aplicadas durante el desarrollo del proyecto gracias a su capacidad para manipular módulos funcionales que permiten expandir el comportamiento sencillo de base.

3.3.2. Ionic Framework

Ionic Framework, directamente conocido como Ionic, es una SDK de código abierto para el desarrollo de aplicaciones móviles basada en Angular y Apache Cordova, un entorno de desarrollo que permite acceder a los componentes nativos de los dispositivos móviles, como la cámara, el GPS o la linterna. Se basa, como ocurre en el resto de aplicaciones híbridas, en el uso de lenguajes de programación web (HTML, CSS, Sass, JavaScript...) para la imagen y la lógica del programa. Este framework viene acompañado de una herramienta de construcción de interfaces llamada Ionic Creator, que permite la rápida implementación del apartado gráfico.

Las aplicaciones creadas mediante las herramientas que ofrece este framework pueden ser distribuidas para dispositivos Android, iOS, Windows e incluso navegadores web. Esto lo logra envolviendo las funcionalidades de Apache Cordova para el acceso a los elementos nativos con una línea de comandos simplificada.

Al usar Angular, Ionic se sirve de la estructuración mediante componentes personalizables y métodos para la interacción entre estos, como señalábamos anteriormente. Gracias a esta estructura, con un solo componente podrían crearse colecciones del mismo en forma de lista sin repercusiones en el rendimiento. A esto, Ionic añade sus propias tipografías, paradigmas interactivos y una temática base extensible que facilita un desarrollo limpio, rápido y agradable a la vista.

En comparación con otras aplicaciones híbridas, unir Ionic al código de la aplicación móvil nativa en Apache Cordova permite un mayor rendimiento del producto resultante: la utilización de Angular permite a Ionic confiar en la aceleración de hardware nativa y aprovecha las transiciones CSS y las transformaciones y animaciones propias para reducir el consumo de tiempo de ejecución del procesador, delegando esta tarea en la GPU del dispositivo. Al aplicar Ionic al desarrollo de la aplicación del proyecto se solventa en gran medida uno de los mayores puntos de debates del desarrollo de aplicaciones nativas, híbridas y web: el rendimiento reducido por la utilización de un buscador.

3.3.3. API REST

Una API REST es una interfaz que permite la simplificación del uso de funcionalidades definidas en una librería software y que permite operar y manipular información relacionada con la librería y sus servicios con un lenguaje de peticiones web, lo que facilita su comprensión y permite su adición a las herramientas de desarrollo estructuradas en los puntos anteriores.

3.3.3.1. Google Drive API REST

Uno de los aspectos cruciales de la aplicación que hemos desarrollado para el estudio de este proyecto ha sido la utilización de la API que ofrece Google para la autenticación del usuario. Tras esta autenticación, se nos permite acceder a los servicios de Google vinculados a la cuenta del usuario, de forma que podemos adquirir acceso y permisos de manipulación de los datos encontrados en la cuenta de Google Drive del usuario.

Uno de los fundamentos por los que nos hemos guiado para la optimización de espacio y mejora organizativa de la base de datos documental es la delegación del almacenamiento de los archivos en esta cuenta, para así mantener un formato estructurado en los documentos de la base de datos, donde se encontraría la referencia al fichero.

Esta ventaja de peso y organización nos es brindada gracias al uso de la API REST vinculada a las funcionalidades de Google Drive, lo que nos permite dentro de la aplicación, modificar la información del usuario a través de órdenes con la misma estructura que las de un servidor web. Google no se queda en este punto, sino que ha desarrollado y ha expuesto para los programadores API REST para todos sus servicios, organizados en Google Cloud Platform, con la posibilidad de estudiar y probar el comportamiento de las peticiones de cada servicio a través de un laboratorio de desarrollo implementado en Google. En él se incluyen API como los servicios de Gmail, su API de traducción y su integración de Google Maps para JavaScript.

3.3.4. Correlación de las tecnologías aplicadas

Por lo general, las herramientas usadas en el proyecto presentan las características de flexibilidad, accesibilidad y complejidad modular. Estos aspectos definen correctamente a su vez el comportamiento de la aplicación que hemos desarrollado, pues ambos comparten un comportamiento sencillo, de fácil aprendizaje, con capacidad para adaptarse a lo que pretende realizar el usuario y con la posibilidad aún más importante de acceder y utilizar solo aquellos aspectos necesarios en el momento, independientes del resto de funcionalidades. Los cuatro elementos que componen MEAN Stack, como comentábamos antes, forman las bases del desarrollo web. Partimos con la base no jerárquica de MongoDB, la abstracción de enrutamientos con ExpressJS, el entorno de desarrollo aportado por Angular y la unificación de estas funcionalidades a través de Node.js.

Por su parte, el entorno Angular permite la estructuración de todo el código en componentes modulares autocontenidos, es decir independientes del resto. Esta estructuración nos permite la replicación de un tipo de marco dentro de la pantalla en la que se muestran contenidos similares, como por ejemplo la proyección de diferentes pdfs partiendo de una misma estructura técnica y visual.

Su mayor ventaja viene de la mano del lenguaje aplicado, TypeScript, el cual permite una mentalidad más jerarquizada y proyectada hacia lo esperado gracias a la adición de tipeado para las estructuras de datos y variables. Dentro de la aplicación, forma el mecanismo desde el inicio de sesión, hasta la creación, acceso y uso de campañas y documentos. Es a través del entorno ofrecido de Angular, apoyado por el entorno Ionic, que hemos logrado el desarrollo principal de la aplicación.

Si Angular es el cuerpo de la aplicación, podemos definir a Node.js dentro del proyecto como el sistema nervioso que conecta las distintas funcionalidades en pro de un avance seguro y ordenado sin dejar de lado el aumento de la complejidad a medida que el proyecto evoluciona. Mediante Node.js se administran los requerimientos prácticos para el desarrollo de la aplicación, desde la instalación y actualización de los paquetes necesarios para realizar ciertas funciones, como la utilización de una plantilla cuadrículada preparada para el drag-and-drop dentro de la aplicación, hasta la puesta en marcha del servidor y la conexión entre la aplicación desarrollada y el navegador web durante la etapa de construcción.

En cuanto a Ionic, su entorno más vinculado a la facilitación de los recursos nativos a través del plugin Apache Cordova y a la creación de un entorno visual familiar para el usuario ha rebajado la carga de diseño y estudio de la experiencia de usuario, permitiendo una mayor concentración en el apartado de programación sin castigar la dedicación pertinente de la visión de la aplicación.

3.4. Riesgos

Durante la producción de una aplicación para dispositivos móviles, uno de los aspectos más relevantes son las condiciones mínimas que deben de cumplirse para que el equipo de desarrollo pueda publicar el software creado. A lo largo de los años, la conducta de las tiendas on-line como Google Play de Android y la App Store de MacOS X ha variado en estos aspectos y los requerimientos han ido evolucionando junto al aumento de la producción en el sector y por las herramientas desarrolladas para la generación de aplicaciones.

Mientras que la plataforma Google Play se mantiene al margen, la plataforma App Store ha tomado unas directrices mucho más estrictas, llegando a analizar con qué entorno de desarrollo se ha generado una aplicación a publicar en la tienda y tomando esta condición como una característica más de filtrado de calidad. Si bien aún ninguna de las tiendas se han expresado para dar su opinión sobre el desarrollo de aplicaciones híbridas mediante el entorno de desarrollo Ionic, el riesgo de que la aplicación sea dada de baja, cancelada o bloqueada de las tiendas es un hecho a tener

en cuenta cuando nos embarcamos en su desarrollo.

Un riesgo latente provocado por una de las características principales de la aplicación es la baja aceptación del público debido al diseño de experiencia de usuario logrado en el menú principal de las campañas de juego. Puntualicemos en que la interfaz aplicada es una malla cuadrangular que permite exponer los contenidos que deseemos, con el tamaño que queramos y en las posiciones que queramos. Esta cierta flexibilidad es en el primer contacto muy poco intuitiva, lo que puede suponer el disgusto del usuario en los primeros minutos y que, aunque a la larga resulte práctico, no se profundice en su uso por no estar acostumbrado.

3.5. Fuentes de información

Como principales fuentes de información para el desarrollo de la aplicación se utilizarán Google Drive y la base de datos del servidor, utilizando MongoDB. Estas herramientas gestionan toda la información de la aplicación en cuanto al usuario se refiere. Para la presentación de la aplicación, se ha utilizado un servidor local personal y la información de la cuenta personal de Google Drive, mostrando un entorno de uso corriente y la estructura de pasos que seguiría el usuario medio para la utilización de la aplicación.

3.6. Diagrama de Elementos

Con las tecnologías presentadas anteriormente hemos desarrollado la aplicación Fantasy Teller, software para la organización y puesta en marcha de campañas en juegos de rol de mesa. La aplicación presenta, principalmente, dos interfaces visuales con comportamiento aislado, denominadas páginas dentro del proyecto.

Estas dos páginas, HomePage y CampaignMenuPage, utilizan componentes, fragmentos de una página web, que se comunican a través de servicios de señales para el intercambio de información. Los componentes actúan como contenedores de funcionalidad para generar una jerarquía dentro de la aplicación, lo que favorece un desarrollo más limpio y estructurado.

Será fácil diferenciar las páginas de estos, ya que el propio entorno de trabajo los separa por una nomenclatura específica. Ambos utilizan el estilo de escritura camelcase y, mientras que las páginas terminan en Page, los componentes terminan en Component. Además de esta diferencia en el nombre, páginas y componentes también se distancian en cuanto a funcionalidad, ya que los componentes no son tratados como capas dentro de la pila de pantallas interna que nos permite navegar por la aplicación. Este último aspecto subyuga a los componentes a pertenecer a una página para poder ser visualizados.

La página HomePage contiene las funcionalidades de creación y selección de campañas, mientras que la CampaignMenuPage contiene todas las acciones vinculadas al desarrollo de las partidas. Así pues, definiremos las funcionalidades de cada elemento dentro de la aplicación según nuestro diagrama de componentes, aunque primero presentaremos un listado de los componentes que serán mentados a continuación.

<p>HomePage</p> <ul style="list-style-type: none"> <u>MenuComponent</u> <ul style="list-style-type: none"> → <u>CustomTextComponent</u> → <u>PrincipalMenuComponent</u> → <u>CampaignSelectorComponent</u> <u>CampaignCreatorComponent</u> <ul style="list-style-type: none"> → <u>DriveFilePickerComponent</u> 	<p><u>CampaignMenuPage</u></p> <ul style="list-style-type: none"> <u>PreferencesComponent</u> <ul style="list-style-type: none"> → <u>PickerComponent</u> <u>LateralMenuComponent</u> <ul style="list-style-type: none"> → <u>FileTypeSearcherComponent</u> <ul style="list-style-type: none"> ◆ <u>PickerComponent</u> <u>CentralMenuComponent</u> <ul style="list-style-type: none"> → <u>CreatureViewerComponent</u> → <u>ModuleViewerComponent</u> <u>FootMenuComponent</u> <ul style="list-style-type: none"> → <u>AmbienceComponent</u> → <u>GalleryComponent</u>
<p>ModalService</p>	
<ul style="list-style-type: none"> <u>SelectedRefImg</u> <u>ViewChanged</u> <u>SelectedPrefFolder</u> <u>PreferencesUpdated</u> 	<ul style="list-style-type: none"> <u>SelectedFileToAdd</u> <u>TrackDeleted</u> <u>ImgDeleted</u> <u>SlideDeleted</u>

Figura 3.1: Componentes de la Aplicación

El código fuente está expuesto en un repositorio de GitHub perteneciente a la cuenta de usuario creada para el alumnado. Para información más detallada respecto al código, por favor, acceder a este y aportar las cuestiones que se le planteen. Durante el análisis de la aplicación se detallará su funcionamiento sin entrar en el código fuente a menos que el detalle de la explicación lo requiera.

3.6.1. Estructuras y elementos principales

Storage es un componente interno de Ionic que ofrece una forma fácil de almacenar pares clave-valor y objetos JSON mediante diferentes sistemas de almacenamiento, escogiendo el mejor disponible dependiendo de la plataforma. Es en este sistema donde salvaguardamos la información de las campañas mediante un único objeto JSON `campaigns`. Cuando Storage se ejecuta en el contexto de una aplicación nativa, priorizará el uso SQLite, mientras que en la web o como una aplicación web progresiva, Storage intentará usar IndexedDB, WebSQL y localStorage, en ese orden.

Como comentamos antes, **campaigns** es un objeto JSON donde se almacena toda la información de las partidas creadas por el usuario, así como el estado en el que se encontraban tras ser utilizadas y las preferencias establecidas por el usuario. Este objeto se estructura tal y como se refleja en la imagen a la derecha.

```

campaigns {2}
  ▶ <campaign-name-I> {2}
  ▼ <campaign-name-II> {4}
    desc : <description>
    ▼ imgRef {4}
      name : <name>
      id   : <id>
      mimeType : <mimeType>
      webContentLink : <link>
    ▼ preferences {4}
      ▶ ambience {4}
      ▶ modules {4}
      ▶ gallery {4}
      ▼ creatures {4}
        name : <name>
        id   : <id>
        mimeType : <mimeType>
        ▶ parents [1]
    ▼ saved {4}
      ▼ showed [1]
        ▼ 0 {4}
          name : <name>
          id   : <id>
          mimeType : <mimeType>
          ▶ parents [1]
        ▼ inputs [1]
          0 : {inputs}
        ▶ imgs [1]
        ▶ music [1]

```

Figura 3.2: Estructura JSON de campaigns

Se puede observar que dos campañas no pueden tener el mismo nombre, así como la jerarquía que se sigue en la aplicación para el guardado de información. Los paquetes de respuesta del servicio DriveService son objetos JSON de los que extraemos principalmente el nombre y el identificador, lo que nos permite señalar el contenido y buscarlo en la cuenta de Google Drive del usuario.

La figura 3.2 muestra un modelo de campaña una vez haya sido utilizada para guardar preferencias y se hayan seleccionado elementos. Una vez cerrado el menú de la campaña, esta información es almacenada en Storage para formar el árbol presentado. En el caso de los componentes pertenecientes a saved.showed, se trata de los componentes mostrados en CentralMenuComponent, cuya categoría ha sido almacenada en este vector, mientras que su identificador y nombre han sido almacenados en inputs.

Aunque más adelante haremos hincapié en los componentes de la aplicación, **PickerComponent** y **DriveFilePickerComponent** merecen una mención por separado debido al carácter repetitivo y semejante en sus tareas. Ambos componentes ofrecen una interfaz que permite explorar las carpetas y ficheros de la cuenta de Google Drive del usuario desde la carpeta raíz. Por su parte, PickerComponent permite definir la búsqueda que queremos realizar según el tipo de elemento que queramos traer a la aplicación, mientras que DriveFilePickerComponent busca imágenes y está adaptado al componente modal que se abre al buscar la imagen de referencia. Su distinción se produjo por la necesidad de dos estilos visuales diferentes para dos páginas diferentes, sin choque de estilos o peticiones al servidor.

3.6.2. ModalService: Servicio de señales

Antes de entrar en materia componente a componente, debemos de explicar el mecanismo por el que se realiza la mayor parte de la comunicación dentro de la aplicación, el ModalService, el servicio que utilizamos para el manejo de señales dentro de la aplicación. Este almacena las señales y es integrado como módulo dentro de los componentes para poder suscribirse a estas o emitirlas. En total, dentro de la aplicación utilizamos ocho señales:

- **selectedRefImg:** Señal aplicada para comunicar desde el DriveFilePickerComponent al CampaignCreatorComponent que una imagen ha sido seleccionada como imagen referencia de la campaña y así pasar la referencia al CampaignCreatorComponent.
- **viewChanged:** Señal aplicada para comunicar al CustomTextComponent dentro de MenuComponent que la vista ha sido cambiada, por lo que debe cambiar su texto.
- **selectedPrefFolder:** Señal aplicada para comunicar al PreferencesComponent dentro de CampaignMenuPage que un directorio ha sido seleccionado dentro de un determinado PickerComponent.

- **preferencesUpdated:** Señal aplicada para comunicar a CampaignMenuPage que las preferencias han sido modificadas y que por tanto debe actualizar las carpetas de LateralMenuComponent para reflejar el cambio.
- **selectedFileToAdd:** Señal aplicada para comunicar a CampaignMenuPage que un fichero ha sido seleccionado en uno de los apartados FileTypeSearcherComponent de LateralMenuComponent.
- **trackDeleted:** Señal aplicada para comunicar a CampaignMenuPage que una pista de música ha sido eliminada en AmbienceComponent, por lo que debe actualizar la información de las pistas.
- **imgDeleted:** Señal aplicada para comunicar a CampaignMenuPage que una imagen de GalleryComponent ha sido eliminada, por lo que debe actualizar la información de las imágenes.
- **slideDeleted:** Señal aplicada para comunicar a CampaignMenuPage que uno de los elementos del CentralMenuComponent ha sido eliminado, ya sea un CreatureViewerComponent o un ModuleViewerComponent, por lo que debe actualizar la información de los componentes mostrados listada dentro de CampaignMenuPage.

3.6.3. UserService: Servicio de Usuario Google

Este servicio permite el inicio de sesión y retención del token de acceso durante el uso de la aplicación. Para ello, utiliza las referencias a los espacios a los que se quiere acceder con la aplicación y el identificador del cliente de la aplicación para reconocerlo en los servidores de Google que reciben las peticiones de la aplicación. Dentro de la aplicación, los métodos aplicados son el de inicio y cierre de sesión, el primero lanzado según lo predicho, y el segundo cuando el menú de campaña en uso es cerrado, para evitar conflictos con el uso de tokens de seguridad dentro de la aplicación y las peticiones recurrentes. El resto de métodos del servicio son de carácter interno para el control de errores y la obtención del usuario para ser usado internamente.

3.6.4. DriveService: Servicio de Peticiones Drive

El servicio que nos permite aplicar la Google Drive API REST es lanzado antes de buscar la imagen de referencia de la campaña a crear en el menú de creación de campañas y antes de acceder a una campaña para obtener acceso completo a las funciones de la aplicación, así como en cada ocasión en la que pedimos referencias a archivos para mostrarlos en la aplicación o cuando se exploran las carpetas de la misma en busca de los archivos. Para realizar el filtrado de las búsquedas, se han definido como constantes los parámetros necesarios para cada uno de los tipos de búsqueda:

- **QNOTRASHEDROOT** para adquirir los ficheros no borrados.

- **QFOLDERITEMS** para obtener las carpetas.
- **QINPARENTS** para definir el padre de los archivos o carpetas de interés.
- **QLISTFIELDS** para indicar los campos que deseamos que se retornen cuando solicitamos listados de archivos y carpetas para mostrar en los componentes de búsqueda PickerComponent y DriveFilePickerComponent.
- **QGETFIELDS** para indicar los campos que deseamos que se retornen cuando solicitamos un archivo en específico.
- **QIMAGEITEMS** para filtrar las búsquedas por los tipos de archivo de imagen que la aplicación admite (png y jpeg).
- **QAUDIOITEMS** para filtrar las búsquedas por los tipos de archivo de audio que la aplicación admite (mpeg y wav).
- **QMODULEITEMS** para filtrar las búsquedas y solo mostrar los archivos PDF.
- **QPARSEITEMS** para filtrar las búsquedas por los tipos de archivo de texto parseables por la aplicación (documentos de Google y texto plano).
- **QALLSEARCH** para buscar todos los archivos de tipos aceptados por la aplicación.

Estas constantes se combinan para realizar las búsquedas pertinentes en la aplicación y así obtener los archivos que el usuario quiere usar y mostrar durante la sesión mediante peticiones HTTP organizadas en funciones dentro del servicio, tales como la obtención del identificador de la ruta del Drive del usuario y la obtención de las carpetas y ficheros hijo de una carpeta padre según un tipo de archivo. Dos de las funciones son remarcables: `getText(file_id:string, authtoken:string)` y `getTypeByName(type:string)`. La primera permite obtener la información en texto plano de un fichero de Google Drive y el segundo obtener el filtro de tipo de archivo según el apartado al que corresponda dentro del buscador. `getTypeByName(type:string)`, para ser más específico, es la función que nos permite mostrar en cada sección de `LateralMenuComponent` los ficheros con la extensión que deben buscar; por ejemplo, se correlaciona el tipo ambiente con `QAUDIOITEMS`, devolviendo este último para realizar la búsqueda.

3.6.5. HomePage: Página de inicio

HomePage es la primera página que se muestra a los usuarios de la aplicación. En ella se realizan las tareas externas a la administración de la partida; es decir, la creación, borrado y acceso a las campañas creadas dentro de la aplicación. Dentro de HomePage existe un único componente, `MenuComponent`, que se encarga de mostrar tanto el menú principal como el menú de acceso a las campañas y al menú de creación.

El componente más simple que conforma el `MenuComponent` es `CustomTextComponent`, un elemento que presenta el texto al usuario en función del menú en el que se encuentre. Se sirve del uso de un condicional interno y de la señal `viewChanged` perteneciente a `ModalService` para comunicar desde los otros dos componentes de `MenuComponent` el paso de un menú a otro sin un cambio de página real, sino sólo aparente para el usuario. Esta metodología permite tratar un cambio tan simple de forma directa, mientras que para el usuario, el comportamiento del menú es el esperado.

`PrincipalMenuComponent`, el primer componente menú que se presenta al usuario, permite el acceso al creador de campañas, `CampaignCreatorComponent`, y al selector de campañas, `CampaignSelectorComponent`. Al acceder al `CampaignCreatorComponent`, una ventana de tipo modal se superpone a la interfaz de `HomePage`, donde podemos introducir el nombre y descripción de la campaña y, tras realizar el acceso de usuario proporcionado por el servicio `UserService`, seleccionar una imagen localizada en nuestro Google Drive.

Con la selección de la imagen en el `DriveFilePickerComponent`, un menú específico para la petición de imágenes y carpetas dentro del Google Drive del usuario, se realiza la señal `selectedReflmg` para almacenar la referencia de drive de la imagen dentro del `CampaignCreatorComponent` y seguidamente apartar la ventana de del `DriveFilePickerComponent`. Al cerrar crear la campaña, los datos del nombre, la descripción y la referencia a la imagen son almacenados en la base de datos nativa de la aplicación, `Storage`, módulo perteneciente a las librerías de Ionic y que se adapta a las propiedades del dispositivo según el sistema operativo del dispositivo móvil o el navegador.

Por otra parte, al activar el `CampaignSelectorComponent`, lo que para el usuario se mostraría como un cambio de menú al selector de campaña, se muestra un menú deslizante en el que podemos ver la imagen de referencia de la campaña, con el nombre y la descripción bajo esta. `CampaignSelectorComponent` es uno de los componentes más básicos de la aplicación ya que su función es la de mostrar las campañas registradas dentro del almacenamiento local proporcionado por Ionic. Esto lo consigue extrayendo las campañas de la base de datos, separando las claves de las campañas de su contenido e iterando por las primeras para permitir el borrado de las campañas, así como la petición de la imagen de referencia para ser mostrada en el menú.

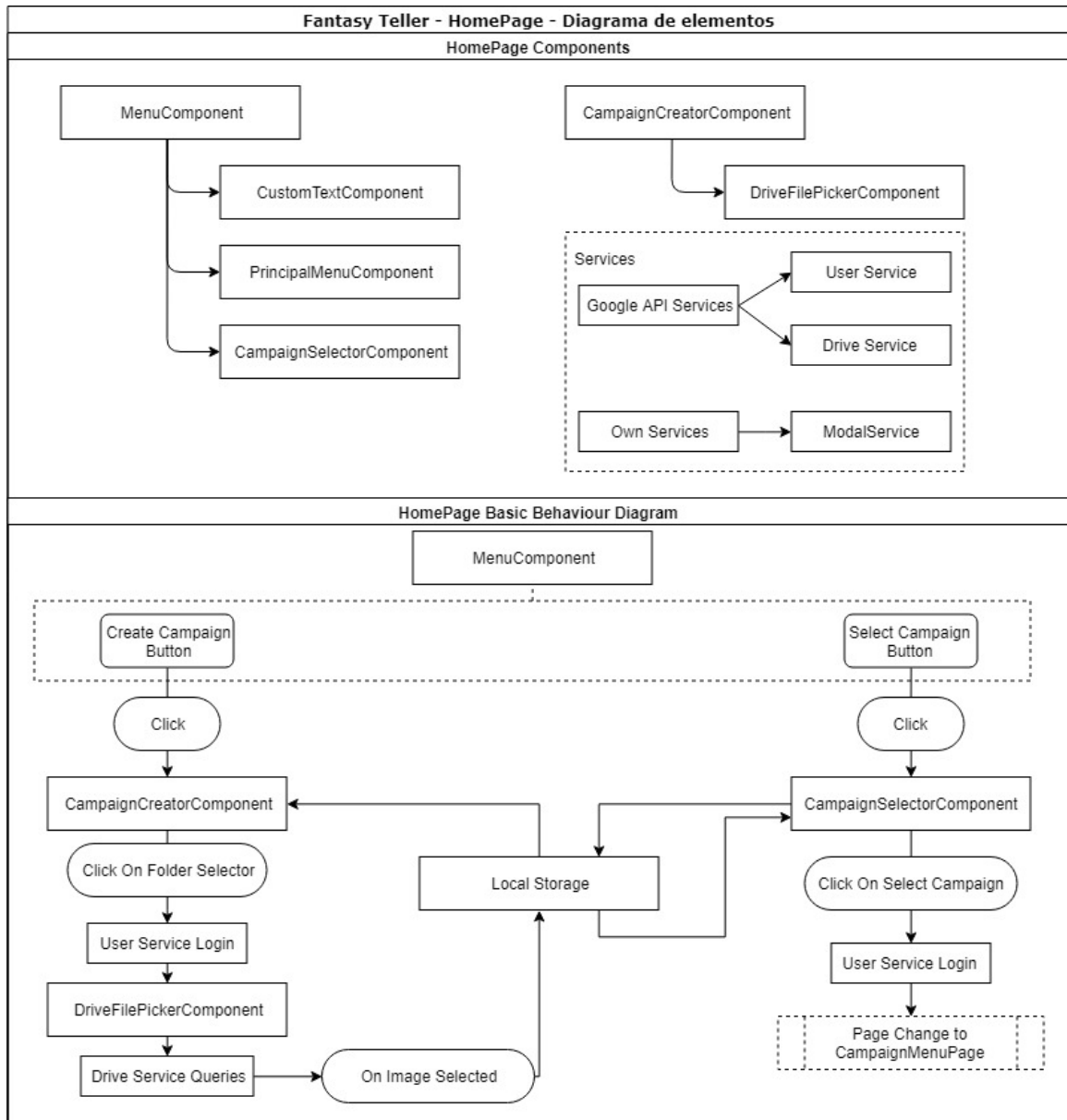


Figura 3.3: Home Page

3.6.6. CampaignMenuPage: Página de partida

Al pasar de HomePage a CampaignMenuPage, la campaña seleccionada es proporcionada a la nueva página para extraer de su estructura los componentes que han sido abiertos en el acceso anterior. En caso de no existir esta parte de la estructura, como es el caso de la campaña recién creada, esta se introduce sin datos para proceder al guardado una vez se cierre la sesión.

En comparación a HomePage, la mecánica de CampaignMenuPage es diferente en cuanto a su encapsulación. Mientras que HomePage solo contiene MenuComponent, CampaignMenuPage contiene una estructura de malla que permite la recolocación de los componentes dentro de la interfaz. Esta funcionalidad, así como la redimensión de los diferentes apartados está planteada para un futuro posterior a la entrega del proyecto debido a cuestiones de tiempo, aunque sus bases hayan sido aplicadas para mantener un orden en el desarrollo.

Dentro de la malla, hemos colocado tres componentes principales que encapsulan toda la funcionalidad de la página: LateralMenuComponent, CentralMenuComponent y FootMenuComponent, que unidos a las preferencias de la barra de navegación, nos permiten acceder a las principales funciones de la aplicación. CampaignMenuPage se sirve de los servicios de Google, UserService y DriveService, para poder acceder a la información de la cuenta de Google Drive del usuario, así como del servicio ModalService para comunicar los tres componentes principales de la página con las estructuras de datos de la misma. Más específicamente, las señales nos permiten saber qué elementos de la campaña añadir o quitar de la aplicación y cuándo hacerlo, manejando esa información mediante cuatro vectores para el almacenamiento de las referencias a imágenes, audios, archivos PDF y documentos.

Antes de definir paso a paso los componentes de CampaignMenuPage, expliquemos las funciones básicas de cada uno. El componente que encabeza las funciones dentro de la página es LateralMenuComponent, ya que dentro de él se dividen los buscadores de archivos dentro de la cuenta de Google Drive del usuario y es a través de estos buscadores que obtenemos la referencia a los archivos en línea. Cuando un archivo es seleccionado, se utiliza su referencia para mostrar la información en el componente adecuado, ya sea AmbienteComponent o GalleryComponent dentro de FootMenuComponent o ModuleViewerComponent y CreatureViewerComponent dentro de CentralMenuComponent. En el caso de FootMenuComponent, su espacio muestra el listado de pistas de audio seleccionadas o la galería de imágenes, según cuál activemos en su propio menú.

Por su parte, CentralMenuComponent recibe la información procesada desde CampaignMenuPage, señalando el tipo de archivo que se manda, la referencia y cualquier dato extra necesario para su visualización. Al tratarse de un deslizador con componentes diferentes, CentralMenuComponent debe permitir la carga dinámica de información, aspecto conseguido a través del módulo ng-dynamic-component, escrito en Angular. Este componente permite la carga dinámica de componentes con sus inputs particulares, en nuestro caso listas dentro de CampaignMenuPage.

Cabe destacar que para el guardado de la sesión por cada campaña se ha recurrido a una traducción de componentes a identificadores que permite la serialización de toda la información almacenable de la sesión. Esta información es almacenada dentro de la estructura de datos que representa a la campaña dentro de la base de datos y contiene las estructuras JSON de las referencias tanto a imágenes como a audios, mientras que las referencias a PDFs y documentos, además de ser guardadas como estructuras JSON, se ven apoyadas en un vector de información que contiene el tipo de componente que debe introducirse en el `CentralMenuComponent` para mostrar la información de la referencia. Esto es necesario debido a que el `CentralMenuComponent` se sirve de un módulo de Angular que permite la visualización dinámica de componentes en la página web, permitiendo así que en el menú deslizante que compone `CentralMenuComponent` hayan componentes de distintos tipos, como es el caso de los archivos PDF y los documentos de Google parseados.

Dentro de la interfaz de `CampaignMenuPage` tenemos acceso a las preferencias de la campaña mediante un botón en la barra de navegador de la aplicación. Al activarlo, se genera el `PreferencesComponent` de la campaña, donde tenemos una lista de los diferentes apartados existentes en `LateralMenuComponent`. Cada objeto de la lista puede vincularse a una carpeta de Google Drive, permitiendo así que en su buscador, dentro de `LateralMenuComponent`, podamos revisar archivos y carpetas internos al seleccionado. Esto lo conseguimos a través de un `PickerComponent` que nos permite navegar por carpetas hasta que seleccionemos uno de ellos, momento en el que utilizamos la señal `selectedPrefFolder` de `ModalService` para que `PreferencesComponent` obtenga la referencia a la carpeta y almacene todas esas referencias dentro de la estructura de datos de la campaña una vez cerremos este componente.

Como decíamos, `LateralMenuComponent` es uno de los componentes esenciales de la aplicación ya que nos aporta la capacidad para navegar por nuestros documentos de Google Drive filtrados por las extensiones admitidas por la aplicación. Más específicamente, `LateralMenuComponent` contiene una lista de elementos `AccordionComponent`, contenedores visuales que permiten la minimización y maximización del contenido de los buscadores no nombrado por ser un mero artículo de embellecimiento. Dentro de cada `AccordionComponent` se almacena un `FileTypeSearcherComponent` en el que se realiza la búsqueda y la visualización de las carpetas marcadas por las preferencias del usuario.

En la mecánica, `LateralMenuComponent` recibe la señal de actualización de preferencias que hace que su vector de referencias cambie. Los `FileTypeSearchComponents`, al depender de estas referencias, son actualizados automáticamente y, según la referencia que reciben, buscan en la cuenta de Google Drive del usuario carpetas y ficheros desde la carpeta marcada como raíz. Esto lo hacen encapsulando un `PickerComponent` para la búsqueda y utilizando una señal para avisar de la selección de un archivo. Además de esto dispone de las funciones necesarias para llamar a los procesos del Drive Service para explorar las carpetas, indicar cuál es su raíz y así saber cuando poder ir hacia la carpeta anterior y cuándo no y la llamada al login de Drive Service para evitar cualquier conflicto con el acceso.

Cuando CampaignMenuPage recibe la señal de que un elemento ha sido seleccionado, filtra esa información para enviarla a CentralMenuComponent, donde se muestran las criaturas, personajes y módulos PDF, o bien a FootMenuComponent, donde se accede a las pistas de audio y a las imágenes seleccionadas en el buscador. Este filtrado se realiza comparando el mimeType de la respuesta del servidor de Google con la cadena de caracteres con la que definimos cada tipo de archivo dentro de la aplicación, listadas durante la explicación de los resultados obtenidos en el desarrollo del proyecto. Una vez se detecta si pertenece a un componente u otro, añade el paquete de respuesta a la lista pertinente.

En el caso del CentralMenuComponent, los componentes añadidos a la lista de su menú pertenecen a la lista showedComponents, donde se señala qué componente debe añadirse al deslizador de componentes interno de CentralMenuComponent y que viene acompañado de showedComponentsInputs, donde se añade el paquete de respuesta del elemento producto de la petición al servidor de Google. Anteriormente citamos la propiedad dinámica de CentralMenuComponent, pero no especificamos del todo cómo se ha obtenido. Así pues, esta propiedad ha sido adquirida gracias al DynamicModule, perteneciente al paquete ng-dynamic-component[23] de Gund.

El paquete ng-dynamic-component permite añadir a las páginas web elementos encapsulados en su componente y que estos puedan ser cambiados por otros elementos, así como cambiados dinámicamente sin la necesidad de modelos. Además permite la introducción de información externa mediante inputs a través de una propiedad de su propio elemento, evitando así la limitación en funcionalidad de los componentes. Aparte de lo citado, solo queda decir que CentralMenuComponent es el que maneja la señal de borrado de los componentes enviada a CampaignMenuPage.

Los componentes introducidos en CentralMenuComponent, ModuleViewerComponent y CreatureViewerComponent, son encapsulaciones básicas de la información proporcionada por la respuesta del servidor de Google. ModuleViewerComponent construye el enlace de vista previa de los ficheros a través de un saneador de errores de seguridad en enlaces, DomSanitizer de Angular, y expone el resultado en su espacio dentro de CentralMenuComponent. La carga de trabajo que se produce en los CreatureViewerComponents es la contraria a la producida en CentralMenuComponent.

En el componente anteriormente explicado, la creación del enlace se hace a través de su propia clase, al crearse el componente, mientras que en el caso de CreatureViewerComponent, al ser la respuesta del servidor un objeto JSON, se ha tomado ese objeto como input del componente y sus resultados son mostrados directamente a través de referencias en el código HTML del componente, lo que nos ahorra la programación de estos pasos dentro de su clase.

Por su parte, FootMenuComponent ofrece un menú parecido a las pestañas, que oculta el contenido de todas menos la seleccionada. El comportamiento es controlado a través del índice de la pestaña, el cual cambia según la que pulsemos. Actualmente solo se encuentran en el AmbienceComponent, componente encargado del listado y reproducción de pistas de audio seleccionadas por el usuario, y GalleryComponent, encargado de la visualización de las imágenes seleccionadas por el usuario.

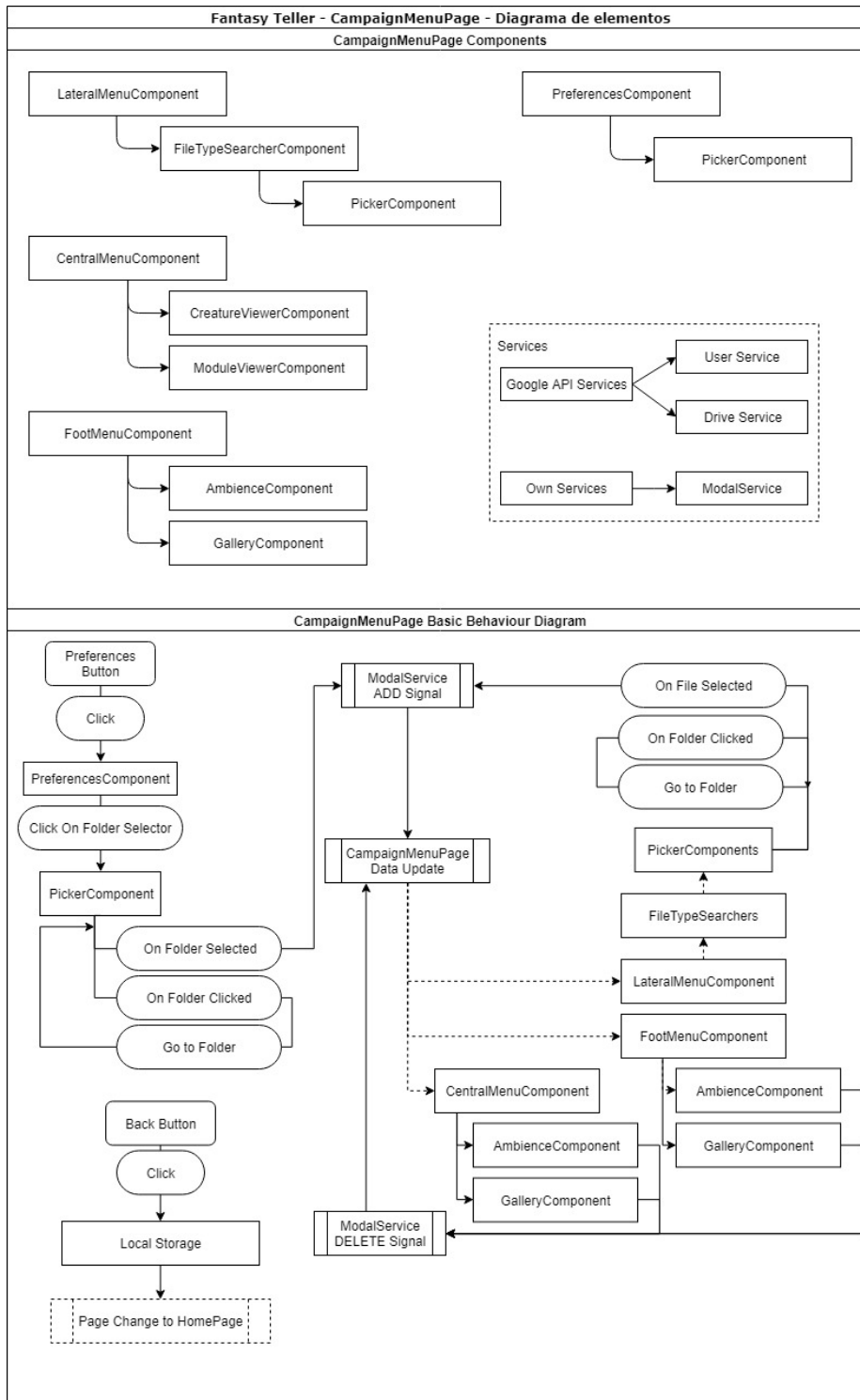


Figura 3.4: Campaign Menu Page

Las listas de elementos a mostrar por ambos componentes son transferidas desde CampaignMenuPage, pasando por FootMenu, hasta tomarlas como inputs de su componente específico, lo que facilita el comportamiento de eliminación visto ya en anteriores secciones como el CentralMenuComponent, donde una señal marcaba el elemento a eliminar. Las señales enviadas desde AmbienceComponent y GalleryComponent son trackDeleted y imgDeleted respectivamente y siguen la misma mecánica de eliminar el elemento de las listas de CampaignMenuPage para que sean actualizados el resto de componentes en cascada.

Tanto AmbienceComponent como GalleryComponent son elementos poco complejos dentro de la jerarquía de la aplicación, pero que ofrecen funcionalidades muy útiles dentro de la gestión de una partida de rol. AmbienceComponent se encarga de tomar los identificadores de los archivos de audio y mostrarlos al usuario junto a los comandos básicos de reproducción, acompañada cada pista de audio de su botón de eliminación en caso de no ser necesaria en el momento. Con la misma filosofía se comporta GalleryComponent que, a través de un deslizador horizontal muestra miniaturas de las imágenes seleccionadas por el usuario, cada una de ellas con su botón de eliminación para su borrado en caso de no quererla más en la galería. Además, las imágenes de la galería pueden ser ampliadas a pantalla completa con un click o toque en la pantalla, lo que permite dar información visual a los jugadores sin desvelar información de lo organizado por el director de partida.

Capítulo 4

Resultados

4.1. Aplicación desarrollada

Este Trabajo de Fin de Grado, como hemos explicado al principio de la memoria, ha tenido por objetivo la creación de una aplicación híbrida Fantasy Teller, destinada a la organización de partidas de juegos de rol de mesa en general, dividiendo las diferentes partes que componen al juego de rol y separándolas en Exploración, Interacción y Combate. Estos tres pilares del juego, se ven reflejados a través de los elementos reproducibles dentro de la aplicación, distribuidos en el menú de juego mediante la sección central e inferior.

Dentro del almacenamiento que ofrece Google Drive, muchos tipos de archivos son permitidos para el guardado e intercambio de información. Por cuestiones de simplicidad, al uso más que suficiente para lo que se quiere ofrecer dentro de la aplicación, los tipos de archivos con los que trabaja Fantasy Teller son documentos de texto plano, documentos de Google, archivos PDF, pistas de audio mp3 y wav e imágenes jpeg y png. Estos tipos de archivo nos permiten disfrutar de la aplicación desarrollada ofrece, pero secundariamente se ampliarán para ofrecer una mayor compatibilidad para los usuarios.

Fantasy Teller ha sido culminada como una aplicación híbrida reproducible en web, con la capacidad de registrar diferentes campañas de juego, con sus propios archivos. En su estado actual, el usuario puede crear y acceder a sus campañas, traer sus documentos desde su cuenta de Google Drive a la aplicación y con ellos registrar a sus jugadores, a las criaturas que se encontrarán en la aventura, leer archivos PDF, reproducir pistas de audio y mostrar imágenes a pantalla completa para disfrutar de sus sesiones de juego con el contenido centralizado.

Respecto a las tareas planteadas para el desarrollo de la aplicación, se ha completado la sección de exploración, permitiendo al usuario el uso de música e imágenes para sumergir a los jugadores dentro de la partida. La sección de Interacción cumple su objetivo de centralizar la información de la historia y los personajes a través de la visualización de archivos PDF y el análisis de archivos de texto plano para extraer información de la historia y de los personajes.

Por último, la sección de combate cumple con el objetivo de manejar la información de combate mediante el análisis de archivos de texto plano para el registro de los oponentes de los jugadores. La capacidad de registrar un Roster o equipo de aventureros dentro de la aplicación, de la forma en la que se ha manejado la información de personajes y criaturas, se considera un suplemento redundante, aunque en proceso de desarrollo para próximas actualizaciones del proyecto. Al igual ocurre con el registro de la Iniciativa, un sistema de turnos, aplicable si el sistema de juego que utilicen el director de juego y los jugadores del grupo se sirve de esto.

La edición de contenido se realiza a través de los archivos de Google Drive, los cuales se actualizan dentro de la aplicación una vez hayamos registrado nuestras carpetas de preferencia dentro de la campaña, aunque se plantea para siguientes actualizaciones del software permitir la modificación del contenido de los ficheros analizados para actualizar la información desde la aplicación hacia la cuenta de Google Drive del usuario.

4.2. Correlación Pilares-Componentes

Si bien hemos hablado de los pilares de juego y de los componentes que forman la aplicación, no se ha mostrado la relación directa entre los componentes de la aplicación y estos conceptos. Aunque la correlación se pueda intuir, no sería redundante una corta explicación de ello.

Así pues, el pilar de la Interacción, basado en la conversación con los elementos del entorno se ve reflejado mediante el componente `ModuleViewerComponent`, el cual permite la reproducción de ficheros PDF, los normalmente utilizados para la escritura de aventuras, historia de trasfondo y herramientas literarias para aportar argumento al juego.

En el mismo contenedor, `CreatureViewerComponent`, componente interno de `CentralMenuComponent`, nos permite tener a nuestro alcance la información histórica y estadística de personajes y criaturas, dando vida al pilar de Combate dentro de la aplicación, así como apoyando la Interacción con el entorno a través de información más directamente vinculada con el grupo de juego.

En una sección separada del resto encontramos el pilar de la Exploración, fundamentado en las aportaciones del `FootMenuComponent` para vincular los sentidos de la vista y el oído a través de música e imágenes para que los jugadores puedan imaginar con mayor fidelidad la idea que el director del juego quiere aportarles en la aventura.

4.3. Guía de uso

Previo al uso de la aplicación es recomendable organizar la información que queremos utilizar en partida en una jerarquía de carpetas para separar los archivos PDF, archivos de audio, imágenes y archivos de texto para mejorar nuestra experiencia de uso. En esta jerarquía directa podremos basar más tarde nuestras preferencias dentro de una campaña, lo que agilizará la búsqueda de información de nuestra cuenta Google Drive y, por ello, reducir el tiempo que perdemos al buscar elementos sin centralizarlos.

La primera pantalla que Fantasy Teller presenta al usuario, el director de juego, es un menú en el que puede leer un breve texto introductorio, acceder a la creación de campañas de juego y seleccionar la campaña en la que quiere trabajar. Una tercera opción aparece en el menú, el acceso a un manual de ayuda, temporalmente deshabilitada a falta de redactar e introducir este manual en la aplicación.

De serie, la aplicación no trae consigo material por defecto, por lo que no podremos acceder a ninguna campaña a menos que creamos una primero. Al abrir el creador de campañas, una segunda pantalla se superpondrá al menú anterior para mostrar un recuadro en el que podremos introducir el nombre de la campaña y una breve descripción de la misma como subtítulo dentro del menú de selección de campañas.

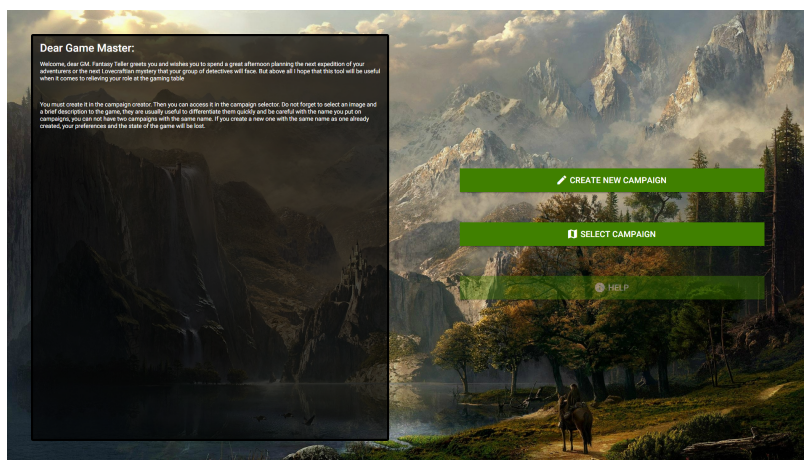


Figura 4.1: Página Inicial

Además podremos seleccionar una imagen de referencia de la campaña como elemento embellecedor. Este elemento es opcional y en caso de no seleccionar una imagen de referencia, se colocará una imagen por defecto en el menú de selección de campañas. Sin embargo, si deseamos incluir una imagen propia como referencia de la campaña, debemos pulsar sobre el botón derecho al lado del espacio de texto de la imagen. Una vez lo pulsemos, si no hemos aceptado los permisos de acceso a la aplicación o no hemos iniciado sesión con nuestra cuenta de usuario Google, se nos mostrará un menú en el que podremos cumplimentar estos requisitos.

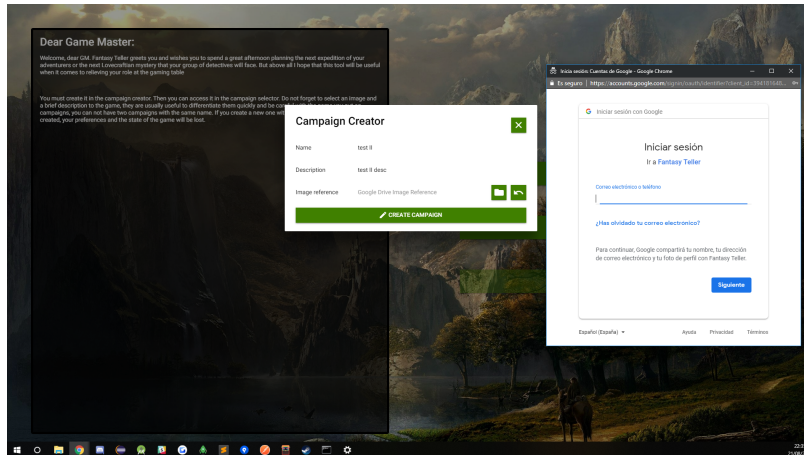


Figura 4.2: Creación de campaña y Acceso a Google

Una vez accedemos a nuestra cuenta de Google, podemos pulsar el mismo botón, ahora con el símbolo de una carpeta, para acceder a un menú en el que podemos recorrer el árbol de carpetas de nuestra cuenta de Google Drive en busca de imágenes. Una vez localicemos la imagen que queremos tener como referencia de la campaña, debemos pulsar la marca de verificación verde que se encuentra a la derecha de la imagen. Esto vinculará la referencia de su objeto dentro de Google Drive a la campaña que estamos creando, lo que nos permite acceder a ella por su identificador.

Una vez rellenemos la información, con pulsar el botón de crear, nuestra campaña se añadirá al almacenamiento local de la aplicación y un mensaje de “campaña creada” nos aparecerá en pantalla. En caso de que no añadamos un nombre a la campaña, el mensaje que nos saldrá será de error, ya que el nombre de la campaña actúa como su identificador dentro de la base de datos, lo que lo hace único e irreplicable. Por esta condición, la descripción y la imagen de referencia son opcionales, mientras que el nombre no puede repetirse entre campañas ya que al crear una campaña con el mismo identificador que una anterior, ésta última sobrescribirá la información de la anterior, perdiendo así el estado en el que quedó nuestra aplicación y las preferencias que hayamos definido.

Una vez creada la campaña, podemos acceder a ella a través del selector. Para ello, salimos del creador de campaña y pulsamos en el botón de selección de campaña. Tras esto, el texto auxiliar a nuestra izquierda habrá cambiado y a nuestra derecha un menú deslizante horizontal habrá ocupado el espacio del menú principal.

En este menú podemos ver la imagen de referencia o la imagen por defecto, en caso de no haber seleccionado una, así como su nombre y su descripción en un apartado por debajo de ésta. El botón a su lado tiene la función de eliminar la campaña, opción irreversible por lo que se debe tener cuidado de borrar una campaña que no queramos borrar. Para acceder a la campaña solo tenemos que pulsar el botón de entrar en campaña y para volver al menú principal debemos pulsar el botón de volver al menú principal, por debajo del menú deslizante.

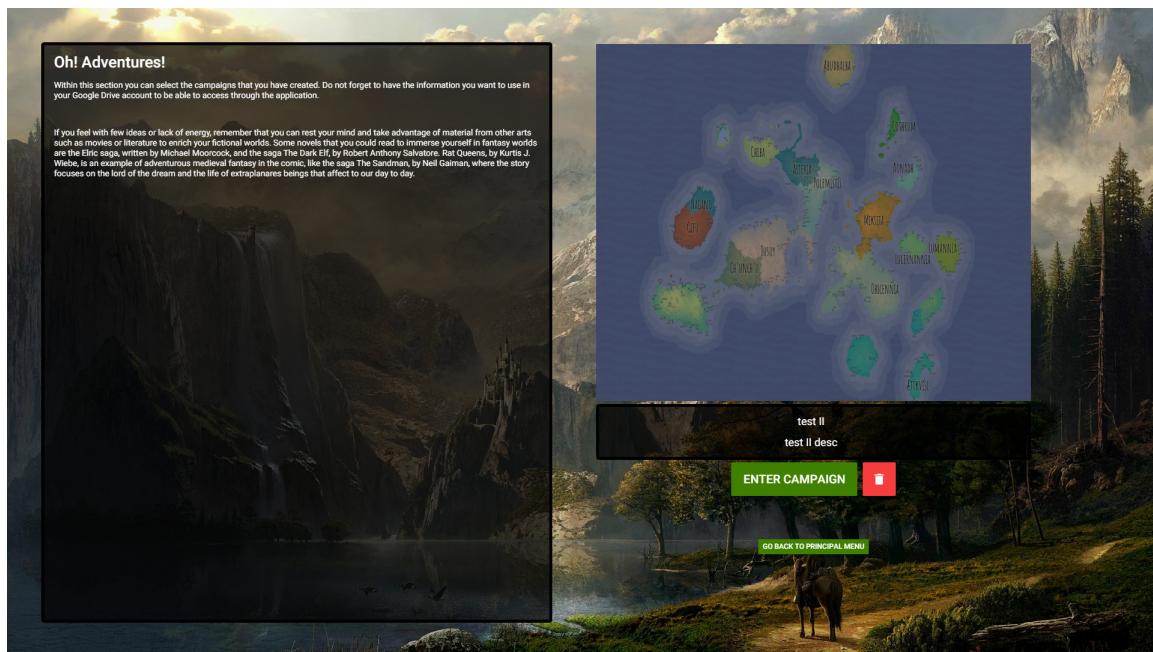


Figura 4.3: Página de selección de campaña

Al acceder a una campaña por primera vez, el menú se nos presenta tal y como vemos en la imagen inferior. Por defecto no tendrá ningún contenido y las preferencias se definirán como vacías. En la parte superior del menú se puede observar el título de la campaña y en el extremo opuesto un botón que nos permite acceder a las preferencias. En la zona central de la pantalla podemos ver a la izquierda el menú desplegable que nos permitirá navegar por nuestros archivos en nuestra cuenta de Google Drive y a la derecha se aprecia el espacio en blanco en el que se introducirán los apartados de criaturas, personajes y archivos PDF. En la zona inferior están las pestañas de música ambiente y galería, en las que podremos ver nuestras pistas de audio e imágenes justo cuando las seleccionemos en el buscador.

Al acceder al apartado de preferencias de la aplicación a través del botón de llave inglesa que hay a la derecha en la barra de navegación de la aplicación, una pantalla se superpondrá, del mismo modo que lo hace el creador de campañas, para mostrar el listado de subapartados que plantea la aplicación: ambiente, characters, legends, gallery y modules. Estos apartados corresponden a la información que podemos extraer de Google Drive y mostrar en la aplicación carpetas que se encuentra en la raíz de nuestro espacio de almacenamiento de Google Drive. En este listado podemos acceder al contenido de una carpeta pulsando sobre su nombre o podremos seleccionarla con el símbolo de verificación que se encuentra a su derecha, lo que quitaría el menú de carpetas y vincularía la referencia a la carpeta con el subapartado con el que se relaciona. Una vez hayamos definido las preferencias, debemos salir del menú para que estas se confirmen y guarden dentro de nuestra campaña.

Las preferencias no son obligatorias. Es decir, no tenemos por qué vincular todos los apartados a alguna carpeta para poder utilizar la aplicación. Por defecto, en el menú lateral tenemos acceso a todas las carpetas y archivos de nuestra cuenta de Google Drive, por lo que siempre podremos buscar información a través de ese menú en vez de en uno de los buscadores especializados. Sin embargo, al aplicar las preferencias y organizar nuestros archivos dentro de Google Drive, podremos tener la información ágilmente, lo que nos ahorrará tiempo de trabajo que podremos dedicar a dirigir la partida y crear contenido para las aventuras.

Hayamos definido o no nuestras preferencias dentro del menú de la campaña, podemos acceder al buscador en el lateral derecho de la interfaz. Este menú está dividido en apartados que separan la información por el tipo de documento que busquemos, además de ofrecer un buscador general que nos permite navegar por todos los archivos de nuestra cuenta de Google Drive admitidos por la aplicación.

En estos menús se nos ofrecerá el listado en acordeón de carpetas y archivos por separado. En esta ocasión las carpetas no son elementos seleccionables, pero se sigue pudiendo acceder a sus subcarpetas y documentos pulsando sobre el nombre de la carpeta. Para añadir un archivo del buscador a la aplicación solo tenemos que pulsar el símbolo de verificación a su derecha.

El apartado Legends filtra los documentos analizables, es decir, texto plano y documentos de Google, en los que almacenamos la información de personajes y criaturas con objetos de formato JSON. Esta información es analizada y visualizada dentro del área central derecha del menú.

```
[[
  {
    "name" : "name-test",
    "category" : "NPC",
    "health" : 12,
    "defense" : 12,
    "quickness" : 12,
    "lore" : "actions-test",
    "actions" : "actions-test",
    "number": 3
  },
  {
    "name" : "name-test-II",
    "category" : "PC",
    "health" : 13,
    "defense" : 13,
    "quickness" : 13,
    "lore" : "actions-test",
    "actions" : "actions-test",
    "number": 2
  }
]]
```

Figura 4.4: Ejemplo JSON de personajes y criaturas

Dentro del archivo de texto plano que queramos importar, debemos meter los objetos JSON dentro de un vector ([]), lo que nos permitirá importar varias criaturas y personajes a la vez. Para diferenciar entre personajes y criaturas, debemos indicar si se trata de un PC, personaje jugable, o un NPC, personaje no jugable.

Dentro de cada objeto podremos indicar nombre, categoría, vida, defensa, velocidad y número, lo que nos permitirá organizar en un futuro personajes y criaturas para el añadido de orden de combate, Initiative, y además los apartados de lore y actions, la información verdaderamente importante para el combate y la interacción, ya que podremos saber qué hace cada jugador y criatura en un momento, así como su historia previa a los encuentros.

Una vez se realiza la importación, los componentes CreatureViewerComponent con la información de cada criatura o personaje son añadidos al menú deslizante. La información del nombre, la cantidad y las estadísticas son mostradas directamente, mientras que las acciones y la historia del mismo son mostradas a través de un elemento acordeón que permite minimizar y maximizar el espacio que ocupa, de manera similar a los menús de búsqueda del lateral izquierdo. Para deslizarnos por el menú con un leve movimiento de arrastre será suficiente para desplazarnos de un elemento a otro.

El apartado ambiente se dedica a filtrar los ficheros de audio wav y mp3 y se añaden a la aplicación al igual que los componentes de criaturas y personajes. Una vez añadidos, podremos reproducirlos, pausarlos y regular su volumen. Además podemos reproducir varias pistas al mismo tiempo, lo que nos permite generar un ambiente personalizado con varias pistas de audio con sonidos simples. Además del apartado de música ambiente en el menú inferior de la aplicación, se encuentra la pestaña Gallery, donde podemos ver una galería de imágenes que se desliza horizontalmente. La galería, añadido a la visualización de una miniatura de la imagen, nos permite aumentar su tamaño a pantalla completa para mostrarla a los jugadores sin desvelar información de la partida.

Por último, los módulos, archivos PDF visualizables en el menú central y que se añaden de manera tan simple como el resto de elementos. Estos archivos pueden ser agrandados, minimizados, se puede aplicar una búsqueda dentro de ellos e incluso se pueden poner a pantalla completa, separando el archivo de la aplicación y abriéndolo a través del navegador con un solo click.

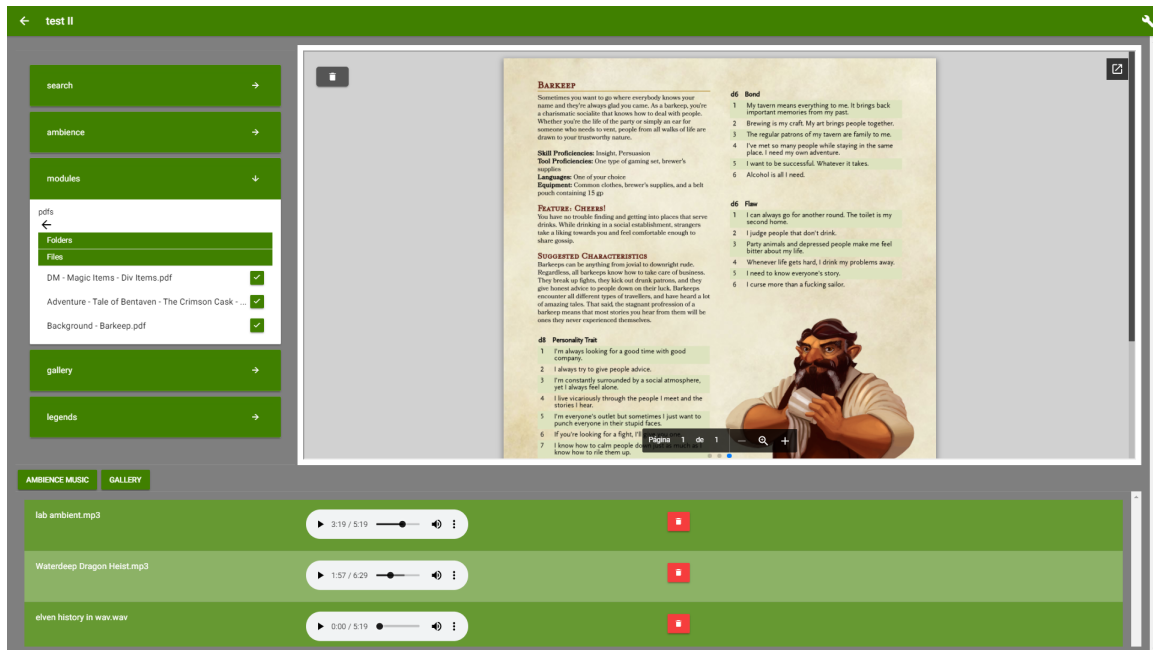


Figura 4.5: Página de partida

4.4. Repositorio del proyecto

Durante el desarrollo del trabajo, se ha llevado registro de la evolución del código mediante un repositorio git en GitHub. Este repositorio es público y se puede acceder a él a través de [alu0100893649/tfg-app](https://github.com/alu0100893649/tfg-app) [24]

Capítulo 5

Conclusiones

5.1. Problema planteado

En la actualidad existe un nicho de mercado vinculado al uso de las tecnologías en los juegos de rol de mesa para mejorar la experiencia de juego. En lo que respecta al desarrollo de herramientas digitales, la gran mayoría contiene material normalmente dedicado a los jugadores que toman el papel de personajes, ya sea mediante aplicaciones de referencia de normas, creación de personajes, lanzamiento de dados y demás. Esto demuestra cierto desamparo a quienes organizan estos juegos, los directores de partida, quienes deben crear contenido jugable para los jugadores y hacer evolucionar este contenido a medida que se va desarrollando la historia.

El problema específico tratado en este Trabajo de Fin de Grado es la ausencia de una aplicación dedicada únicamente a los directores de partida con el propósito de facilitarles la centralización, uso y visualización del contenido que hayan creado u obtenido para sus partidas. Esta deficiencia en el mercado se intenta suplir mediante el desarrollo de una aplicación híbrida que permita a los posibles usuarios acceder a través de la web o de otros dispositivos a su contenido propio, centralizado y de fácil visualización.

5.2. Solución aportada

Haciendo uso de la Google Drive API REST y del entorno de desarrollo de Ionic, el proyecto ha culminado en una aplicación dedicada a la centralización de contenido residente en el sistema de almacenamiento Google Drive del usuario, donde éste puede reproducir pistas de sonido, visualizar imágenes, procesar texto de personajes e historia y leer archivos PDF.

Respecto al sistema de puntuación propuesto para el proyecto del trabajo, se ha cumplido con todas las funciones necesarias para la reducción del trabajo por parte del director de partida, como nos habíamos planteado en un principio. A través de la aplicación desarrollada se ha marcado una posible nueva tendencia en cuanto a la creación de contenido para los aficionados a los juegos de rol de mesa.

Como indicamos anteriormente, no existe una aplicación dedicada a los directores de partida, sino hacia los jugadores y, aunque es cierto que existen aplicaciones que intentan abarcar ambos lados del mismo juego, no ofrecen una opción para cualquier sistema de juego, sino metodología restringida a una serie de estos. Aunque este producto no vaya a salir a mercado por lo pronto, será expuesto continuamente a posibles usuarios y jugadores, hasta poder definir un método de financiación claro que permita sostener sus mejoras, pues el objetivo, más allá del Trabajo de Fin de Grado, es dar una solución a una comunidad creciente a nivel internacional.

5.3. Aprendizaje en el proceso

Partiendo de un conocimiento nulo en el desarrollo web, y teniendo en cuenta los primeros planteamientos de desarrollar una aplicación nativa para el sistema operativo Android, se han adquirido amplios conocimientos sobre el mundo del desarrollo comercial y de las herramientas que en la actualidad se utilizan para satisfacer sus necesidades.

Primero que nada se han reforzado los conocimientos básicos de desarrollo web mediante lenguajes de marcado y de estilo, seguidos por la búsqueda de información de las aplicaciones híbridas, sus diferencias respecto al resto de categorías de aplicaciones, así como sus ventajas e inconvenientes y prácticas aconsejables. En este punto del aprendizaje rodado durante el Trabajo de Fin de Grado ya se habían cimentado el uso de los lenguajes de marcado, el conjunto de sistemas MEAN stack para desarrollo web y el entorno Angular para la programación mediante el lenguaje TypeScript, una variación del lenguaje JavaScript, que añade tipado estático y programación orientada a objetos.

Todos estos conocimientos fueron aplicados a través del entorno de desarrollo Ionic, entorno que también tuvo que estudiarse a fondo para definir las funcionalidades que aporta en el proyecto, así como las posibilidades que tiene para ofrecer una alternativa multiplataforma.

Una vez finalizado el desarrollo de la aplicación se ha indagado, aunque no profundamente en la instalación de la aplicación en PaaS (Platform as a Service), sistemas que permiten el almacenamiento, publicación y desarrollo de aplicación web progresivas, como es el caso de Heroku. También en su estudio se han estudiado los requisitos por los que un desarrollador debe pasar para poder publicar una aplicación en tiendas en línea, así como los mecanismos de los que debemos servirnos para la generación de la aplicación para dispositivos Android, así como su firma digital y su distribución. En este último aspecto han sido gratos los datos obtenidos debido a las bajas tasas a pagar para poder exponer una aplicación para el público de Android en la tienda Google Play, un aspecto que favorece la próxima publicación del producto.

5.4. Futuras mejoras

Sin embargo, antes de poder ofrecer a mercado esta solución para el mundo de los juegos de rol de mesa, debe pasar por una serie de mejoras ya planteadas como tareas externas al desarrollo del Trabajo de Fin de Grado, como es la adición de elementos de menor importancia, como un sistema de turnos o un contenedor de jugadores con el que registrarlos en un apartado separado del resto de la mecánica del menú del que dispone el usuario. Su integración ya está planteada en el código actual y será perfilada próximamente, aunque los métodos necesarios para completarla se han planteado. Tanto para el registro de turnos como en el de usuarios, bastará con crear dos componentes añadibles al menú inferior que sean actualizados con la integración de criaturas y personajes desde el menú de búsqueda. Esta actualización del contenido interno se verá reflejada en vectores de referencia semejantes a los encontrados en la estructura de datos del menú de la campaña, lo que promete una integración conocida y sin errores.

Además de la adición de estos dos elementos, una actualización a la forma en la que se distribuyen los componentes dentro del menú de campaña se ha planteado urgentemente, ya no aprovecha del todo el espacio disponible de la pantalla, pudiendo delegar algunas funciones a botones en el navegador o redistribuyendo los menús en la pantalla. La solución a este cambio visual consiste en la recolocación de los componentes de el menú inferior en el menú lateral y la integración de los componentes de búsqueda en un menú proporcionado a través de la barra de navegación, semejante al aplicado para las preferencias, lo que proporciona mucho más espacio para los componentes centrales.

5.5. Conclusiones

El proyecto de Fin de Grado planteado, definido como el desarrollo de una aplicación híbrida para solucionar un hueco en el mercado del desarrollo tecnológico de los juegos de rol de mesa, ha resultado en un producto a futuro de características notables. Su planteamiento supuso abarcar una falta dentro de la comunidad dedicada a esta afición, en la actualidad una comunidad creciente a nivel internacional, dotando a este conjunto de usuarios de una herramienta sólida y ligera que les permite aunar sus esfuerzo en la creación de contenido dentro de un único programa.

Durante su desarrollo se han fomentado en el alumno conocimientos sobre las tecnologías web y la política de mercado actual, preparándolo para el lanzamiento de productos propios con el paso de los años y el crecimiento de su experiencia. Y es que el mayor beneficio de este desarrollo es la motivación al desarrollo particular y la búsqueda de soluciones eficientes en un mercado evolutivo.

5.6. Conclusions

The proposed End of Degree project, defined as the development of a hybrid application to solve a gap in the technological development market for tabletop role-playing games, has resulted in a product with notable characteristics in the future. His approach involved covering a fault within the community dedicated to this hobby, currently a growing international community, providing this set of users with a solid and light tool that allows them to combine their efforts in the creation of content within a single program.

During its development, knowledge about web technologies and current market policy has been fostered in the student, preparing it for the launch of its own products over the years and the growth of its experience. And the biggest benefit of this development is the motivation to the particular development and the search for efficient solutions in an evolutionary market.

Capítulo 6

Presupuesto

Para el presupuesto del proyecto se ha acotado la hora de trabajo como pagada a 18 euros y tomando un total de 250 horas para su desarrollo, incluyendo en este la investigación previa al desarrollo así como el desarrollo propio de la aplicación. El presupuesto total de la aplicación, incluyendo licencias de desarrollador para Android y manutención del servidor quedaría en un total de 4500 euros.

Bibliografía

- [1] raona, “¿app nativa, web o híbrida?” 2017. [Online]. Available: <https://www.raona.com/aplicacion-nativa-web-hibrida/>
- [2] Google, “Api de google,” 2018. [Online]. Available: https://es.wikipedia.org/wiki/API_de_Google
- [3] Linnovate, “Página web oficial,” 2014. [Online]. Available: <http://mean.io/>
- [4] Ionic, “Página web oficial,” 2018. [Online]. Available: <https://ionicframework.com/>
- [5] The Orr Group, “Página web oficial,” 2018. [Online]. Available: <https://roll20.net/>
- [6] Stanley Sison, “Página de la appstore de ios de la aplicación,” 2018. [Online]. Available: <https://itunes.apple.com/us/app/epicgm/id1292118031?ls=1&mt=8>
- [7] Google, “Página web oficial,” 2018. [Online]. Available: <https://play.google.com/store>
- [8] Wizards of the Coast, “Página web oficial,” 2018. [Online]. Available: <http://dnd.wizards.com/>
- [9] Unity Technologies, “Página web oficial,” 2018. [Online]. Available: <https://unity3d.com/es>
- [10] Wikipedia, “Interfaz de programación de aplicaciones,” 2018. [Online]. Available: https://es.wikipedia.org/wiki/Web_API
- [11] Wikipedia, “Transferencia de estado representacional,” 2018. [Online]. Available: https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional
- [12] Wikipedia, “Comedia del arte,” 2018. [Online]. Available: https://es.wikipedia.org/wiki/Comedia_del_arte
- [13] Wikipedia, “Juego de guerra,” 2018. [Online]. Available: https://es.wikipedia.org/wiki/Juego_de_guerra
- [14] Wikipedia, “Cathuranga,” 2018. [Online]. Available: <https://en.wikipedia.org/wiki/Chaturanga>

- [15] Wikipedia, “Little wars,” 2018. [Online]. Available: https://es.wikipedia.org/wiki/Little_Wars
- [16] Wikipedia, “Diplomacy,” 2018. [Online]. Available: [https://en.wikipedia.org/wiki/Diplomacy_\(game\)](https://en.wikipedia.org/wiki/Diplomacy_(game))
- [17] Wikipedia, “Metamorphosis alpha,” 2018. [Online]. Available: https://en.wikipedia.org/wiki/Metamorphosis_Alpha
- [18] Wikipedia, “La llamada de cthulu,” 2018. [Online]. Available: [https://es.wikipedia.org/wiki/La_llamada_de_Cthulhu_\(juego_de_rol\)](https://es.wikipedia.org/wiki/La_llamada_de_Cthulhu_(juego_de_rol))
- [19] Wikipedia, “James bond 007: Role playing in her majesty’s secret service,” 2018. [Online]. Available: [https://es.wikipedia.org/wiki/James_Bond_007_\(juego_de_rol\)](https://es.wikipedia.org/wiki/James_Bond_007_(juego_de_rol))
- [20] Rol con Dados, “1808, el juego de rol ambientado en la guerra de la independencia,” 2018. [Online]. Available: <https://rolcondados.com/j/1808/>
- [21] LMDE Grupo Creativo, “Págin web oficial,” 2018. [Online]. Available: <http://lamarcadeleste.com/>
- [22] Google, “Página web oficial,” 2018. [Online]. Available: <https://developer.android.com/studio/>
- [23] Gund, “ng-dynamic-component - npm,” 2018. [Online]. Available: <https://www.npmjs.com/package/ng-dynamic-component>
- [24] S. J. D. Rodríguez, “Fantasy teller repository,” 2018. [Online]. Available: <https://github.com/alu0100893649/tfg-app>