

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

**ARPEGOS: AUTOMATIZED ROLEPLAYING-GAME  
PROFILE EXTENSIBLE GENERATOR ONTOLOGY  
BASED SYSTEM**

AUTOR: ALEJANDRO MUÑOZ DEL ÁLAMO

Puerto Real, mes \_ año



TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

**ARPEGOS: AUTOMATIZED ROLEPLAYING-GAME  
PROFILE EXTENSIBLE GENERATOR ONTOLOGY  
BASED SYSTEM**

DIRECTOR: D. ALBERTO GABRIEL SALGUERO HIDALGO  
CODIRECTOR: D. PABLO GARCÍA SÁNCHEZ  
AUTOR: ALEJANDRO MUÑOZ DEL ÁLAMO

Puerto Real, mes \_ año



## DECLARACIÓN PERSONAL DE AUTORÍA

Alejandro Muñoz Del Álamo con DNI 77396804-X, estudiante del Grado en Ingeniería Informática en la Escuela Superior de Ingeniería de la Universidad de Cádiz, como autor de este documento académico, titulado ARPEGOS: Automated Roleplaying-game Profile Extensible Generator Ontology based System y presentado como trabajo final de Grado.

### DECLARO QUE

Es un trabajo original, que no copio ni utilizo parte de obra alguna sin mencionar de forma clara su origen, tanto en el cuerpo del texto, como en su bibliografía, y que no empleo datos de terceros sin la debida autorización, de acuerdo con la legislación vigente. Así mismo, declaro que soy plenamente consciente de que no respetar esta obligación podrá implicar la aplicación de sanciones académicas, sin perjuicio de otras actuaciones que pudieran iniciarse.

En Puerto Real, a día \_ de mes \_ de año

Fdo: Alejandro Muñoz Del Álamo



Historial de Cambios		
Proyecto: ARPEGOS		
Fecha: mes _ año		
Cambio No.	Tipo de cambio	Descripción del cambio





## *Dedicatoria*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam maximus facilisis odio vel dignissim. Integer sed tincidunt neque. Cras vitae nisi odio. Morbi in pellentesque nulla. Nam sagittis leo nec ex ultricies, eget hendrerit velit tincidunt. Phasellus fringilla varius tellus, in aliquam purus convallis eget. Donec mattis nisi turpis, vitae ultrices lectus ornare at. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in rutrum eros. Duis ut lacinia justo. Donec fringilla velit eu sem congue cursus. Etiam porttitor, justo gravida porttitor ultrices, eros sem lacinia ex, eget bibendum enim dolor non ligula. Lorem ipsum dolor sit amet, consectetur adipiscing elit.



## *Agradecimientos*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam maximus facilisis odio vel dignissim. Integer sed tincidunt neque. Cras vitae nisi odio. Morbi in pellentesque nulla. Nam sagittis leo nec ex ultricies, eget hendrerit velit tincidunt. Phasellus fringilla varius tellus, in aliquam purus convallis eget. Donec mattis nisi turpis, vitae ultrices lectus ornare at. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in rutrum eros. Duis ut lacinia justo. Donec fringilla velit eu sem congue cursus. Etiam porttitor, justo gravida porttitor ultrices, eros sem lacinia ex, eget bibendum enim dolor non ligula. Lorem ipsum dolor sit amet, consectetur adipiscing elit.



## *Resumen*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam maximus facilisis odio vel dignissim. Integer sed tincidunt neque. Cras vitae nisi odio. Morbi in pellentesque nulla. Nam sagittis leo nec ex ultricies, eget hendrerit velit tincidunt. Phasellus fringilla varius tellus, in aliquam purus convallis eget. Donec mattis nisi turpis, vitae ultrices lectus ornare at. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in rutrum eros. Duis ut lacinia justo. Donec fringilla velit eu sem congue cursus. Etiam porttitor, justo gravida porttitor ultrices, eros sem lacinia ex, eget bibendum enim dolor non ligula. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

## *Abstract*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam maximus facilisis odio vel dignissim. Integer sed tincidunt neque. Cras vitae nisi odio. Morbi in pellentesque nulla. Nam sagittis leo nec ex ultricies, eget hendrerit velit tincidunt. Phasellus fringilla varius tellus, in aliquam purus convallis eget. Donec mattis nisi turpis, vitae ultrices lectus ornare at. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in rutrum eros. Duis ut lacinia justo. Donec fringilla velit eu sem congue cursus. Etiam porttitor, justo gravida porttitor ultrices, eros sem lacinia ex, eget bibendum enim dolor non ligula. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

## *Palabras clave*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam maximus facilisis odio vel dignissim. Integer sed tincidunt neque. Cras vitae nisi odio. Morbi in pellentesque nulla. Nam sagittis leo nec ex ultricies, eget hendrerit velit tincidunt. Phasellus fringilla varius tellus, in aliquam purus convallis eget. Donec mattis nisi turpis, vitae ultrices lectus ornare at. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in rutrum eros. Duis ut lacinia justo. Donec fringilla velit eu sem congue cursus. Etiam porttitor, justo gravida porttitor ultrices, eros sem lacinia ex, eget bibendum enim dolor non ligula. Lorem ipsum dolor sit amet, consectetur adipiscing elit.



## Notación y formato

En la siguiente tabla se presenta un conjunto de convenios de notación de sintaxis.

Notación establecida	
<b>negrita</b>	Título o texto destacado
<i>cursiva</i>	Texto en otro idioma, destacado, citas o nombres de aplicaciones
<code>monoespaciado</code>	Referencias a código fuente
<u>subrayado</u>	Advertencia para el lector
color	Enlace interno ( <b>rojo</b> )





<b>I</b>	<b>Prolegómeno</b>	<b>1</b>
<b>1.</b>	<b>Introducción</b>	<b>3</b>
1.1.	Motivación . . . . .	3
1.2.	Alcance . . . . .	4
1.3.	Objetivos . . . . .	4
1.4.	Glosario de Términos . . . . .	5
1.5.	Metodología . . . . .	7
1.6.	Estructuración del documento . . . . .	8
<b>2.</b>	<b>Antecedentes</b>	<b>9</b>
2.1.	Estado del Arte . . . . .	9
2.1.1.	Situación actual de la tecnología . . . . .	9
2.1.2.	Crítica al estado del arte . . . . .	12
2.1.3.	Propuesta . . . . .	12
<b>3.</b>	<b>Análisis de requisitos del sistema</b>	<b>13</b>
3.1.	Catálogo de actores . . . . .	13
3.2.	Requisitos funcionales . . . . .	13
3.3.	Requisitos no funcionales . . . . .	14
3.4.	Reglas de negocio . . . . .	15
3.5.	Análisis de las soluciones . . . . .	15
3.5.1.	Android Studio . . . . .	15
3.5.2.	React Native . . . . .	16
3.5.3.	Xamarin . . . . .	16
3.6.	Solución propuesta . . . . .	16



---

## Índice de figuras

---

2.1.	<i>5e Character</i> : Pantalla de selección de clases . . . . .	9
2.2.	<i>5e Character</i> : Información de la clase <i>Druida</i> . . . . .	9
2.3.	<i>RPG Simple Dice</i> : Pantalla de selección de dados . . . . .	10
2.4.	<i>RPG Simple Dice</i> : Ejemplo de lanzamiento de dados . . . . .	10
2.5.	<i>BattleTrack</i> : Pantalla de selección de grupo . . . . .	10
2.6.	<i>BattleTrack</i> : Ejemplo de combate . . . . .	10
2.7.	<i>RPGSound</i> : Menú principal . . . . .	11
2.8.	<i>RPGSound</i> : Menú de <i>Ambiente Sostenido</i> . . . . .	11
2.9.	<i>RPG Character Sheet</i> : Pantalla de <i>Estado</i> . . . . .	11
2.10.	<i>RPG Character Sheet</i> : Pantalla de <i>Features</i> . . . . .	11



---

## Índice de cuadros

---

3.1. Comparativa entre <i>JavaScript</i> y <i>C#</i> . . . . .	16
--	----



Parte I

Prolegómeno





### 1.1. Motivación

El sector de los juegos de rol está creciendo a pasos agigantados. Hoy en día existe un extensísimo catálogo de juegos de rol, que varían en función de la ambientación del “universo” en el que nos sumergimos. Por ejemplo, *Dragones y Mazmorras* utiliza una ambientación medieval con elementos fantásticos, tales como magia y dragones, mientras que *Warhammer 40.000* está basado en un futuro distópico en el que se mezclan elementos de ciencia ficción con elementos de fantasía heroica.

Debido a la expansión de este sector, y gracias al avance de la tecnología, podemos encontrar un sinnúmero de aplicaciones que tratan de mejorar la experiencia de los jugadores, desde juegos de mesa que hacen uso de sistemas de reproducción (*CD*, *DVD*, *Blu-Ray*) para hacer del juego una experiencia interactiva (*Atmosfear*, *Piratas del Caribe: El Cofre del Hombre Muerto*, *Trivial Pursuit*, *Cluedo*), hasta juegos de rol completamente virtualizados que nos permiten sumergirnos en ellos, contemplar sus parajes, disfrutar de su ambientación, y en algunos casos, nos permiten participar con jugadores de cualquier parte del mundo a través de la red (*World of Warcraft*, *Diablo III*, *Black Desert Online*).

Dentro de este amplio espectro de posibilidades, los juegos de rol tradicionales también disponen de aplicaciones que enriquecen el transcurso de las partidas, y que proveen prestaciones útiles para las mismas, como lanzadores de dados virtuales, generadores de mazmorras, contadores de iniciativa, compendios de habilidades, etc.

Entre todas las utilidades que encontramos, cabe destacar las aplicaciones conocidas como *generadores de personaje*. Estas facilitan a los jugadores todo lo necesario para desarrollar rápidamente la información indispensable de un personaje para poder interpretarlo en una partida.

Por otro lado, puesto que la tecnología tiene un ritmo de avance trepidante, y más aún la informática, hoy en día podemos echar cuenta de servicios que simplifican tareas cuya ejecución resultaba impensable en el pasado.

Uno de estos servicios es la Web Semántica, que se basa en la idea de describir el contenido, el significado y la relación de los datos, de manera que sea posible evaluarlos automáticamente por máquinas de procesamiento, como pueden ser los ordenadores o los *smartphones*.

El planteamiento del presente proyecto surge de la concepción de una aplicación informática destinada a ampliar las funcionalidades de los generadores de personaje, innovando en el desarrollo de la misma al aplicar tecnologías de Web Semántica.

## 1.2. Alcance

El alcance de este proyecto comprende el desarrollo de una aplicación móvil para generar perfiles de personajes que puedan formar parte de una partida o campaña de un juego de rol, cuya base de datos esté incluida en la aplicación.

No forma parte del alcance de este proyecto el desarrollo de la base de datos del juego de rol, aunque ha sido necesario para poder realizar la sección de pruebas del proyecto.

## 1.3. Objetivos

El sistema va a tener tres finalidades claramente diferenciadas:

- **Selección de juego:** La aplicación podrá dar acceso a varios juegos, de manera que se pueda alternar entre éstos, permitiendo utilizar el mismo mecanismo para el catálogo de juegos disponible.
- **Creación/Modificación de personaje:** La aplicación permitirá al usuario seleccionar la información necesaria para la creación de un personaje a su gusto mediante un proceso guiado paso a paso. En caso de que el personaje ya exista, permitirá al usuario visualizar y realizar modificaciones a la información mostrada.
- **Automatización del proceso de cálculo de habilidades:** La aplicación facilitará al usuario una interfaz en la que, al indicar la habilidad que desea utilizar, y el resultado de su lanzamiento de dados, se devuelva el resultado total que se debe aplicar en el combate. También podrá calcular el resultado de tiradas enfrentadas.

Además, la aplicación deberá hacer muestra de las siguientes cualidades:

- **Generalidad:** La aplicación no estará directamente vinculada a la información específica de un juego, de forma que sea posible procesar diferentes bancos de datos, y por tanto, se pueda utilizar la misma aplicación para varias versiones distintas del mismo juego, o incluso para juegos completamente diferentes.

- **Sencillez:** La aplicación dispondrá de una interfaz simple y agradable, que permita al usuario hacer uso de sus funciones de forma asequible, sea cual fuere la complejidad del juego seleccionado.

## 1.4. Glosario de Términos

- **Android:** Sistema operativo que se emplea en dispositivos móviles, por lo general con pantalla táctil. De este modo, es posible encontrar tabletas, teléfonos móviles y relojes equipados con Android, aunque el software también se usa en otros dispositivos.
- **Blazegraph:** Base de datos de grafos de código abierto, escalable y de alto rendimiento basada en estándares. Escrito completamente en *Java*, la plataforma soporta las familias de especificaciones *Blueprint* y **RDF/SPARQL 1.1** incluyendo consultas, actualizaciones, consultas federadas básicas y descripción de servicios.
- **C#:** Lenguaje de programación multiparadigma desarrollado y estandarizado por Microsoft como parte de su plataforma **.NET**, que después fue aprobado como un estándar por la ECMA (*ECMA-334*) e ISO (*ISO/IEC 23270*). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común. Su sintaxis básica deriva de *C/C++* y utiliza el modelo de objetos de la plataforma net, similar al de *Java*, aunque incluye mejoras derivadas de otros lenguajes.
- **RDFSharp:** *Framework* de código abierto **C#** diseñado para facilitar la creación de aplicaciones *.NET* basadas en el modelo **RDF**, que representa una solución didáctica directa para comenzar a trabajar con conceptos de *Semántica Web*.
- **Git:** Software de control de versiones diseñado por *Linus Torvalds*, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.
- **GitHub:** GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones **Git**.
- **IDE:** (*Integrated Development Environment*) Aplicación con numerosas características que se pueden usar para muchos aspectos del desarrollo de software.
- **iOS:** Sistema operativo móvil de la multinacional *Apple Inc.* Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. No permite su instalación en hardware de terceros.
- **Metodología Ágil:** Las metodologías ágiles son métodos de desarrollo de software en los que las necesidades y soluciones evolucionan a través de una colaboración estrecha entre equipos multidisciplinarios. Se caracterizan por enfatizar la comunicación frente a la documentación, por el desarrollo evolutivo y por su flexibilidad.

- **Modelo:** Las clases de modelo son clases no visuales que encapsulan los datos de la aplicación. Por lo tanto, se puede considerar que el modelo representa el modelo de dominio de la aplicación, que normalmente incluye un modelo de datos junto con la lógica de validación y negocios.
- **Modelo de Vista:** El modelo de vista implementa las propiedades y los comandos a los que la vista puede enlazarse y notifica a la vista de cualquier cambio de estado a través de los eventos de notificación de cambios. Las propiedades y los comandos que proporciona el modelo de vista definen la funcionalidad que ofrece la interfaz de usuario, pero la vista determina cómo se mostrará esa funcionalidad.
- **MVVM:** Patrón de arquitectura de software que ayuda a separar la lógica de negocios y presentación de una aplicación de su interfaz de usuario.
- **Ontología:** Definición formal de tipos, propiedades, y relaciones entre entidades que realmente o fundamentalmente existen para un dominio de discusión en particular. Es una aplicación práctica de la ontología filosófica, con una taxonomía.
- **OWL:** (*Ontology Web Language*) es un lenguaje de marcado semántico para publicar y compartir ontologías en la World Wide Web. OWL se desarrolla como una extensión de vocabulario de **RDF** y es derivado del lenguaje DAML + OIL así.
- **Protégé:** Framework editor de ontologías de código abierto y gratuito para construir sistemas inteligentes.
- **RDF:** (*Resource Description Framework*) Modelo estándar para el intercambio de datos en la Web. RDF tiene características que facilitan la fusión de datos incluso si los esquemas subyacentes difieren, y admite específicamente la evolución de los esquemas a lo largo del tiempo sin requerir que se cambien todos los consumidores de datos.
- **RDFS:** *RDF Schema* es una extensión del vocabulario básico de *RDF* que proporciona un vocabulario de modelado de datos para los datos relativos a este modelo.
- **Scrum:** Marco de trabajo para la gestión y desarrollo del software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil del software.
- **SonarQube:** Herramienta de revisión automática de código para detectar errores, vulnerabilidades y olores de código en su código. Se puede integrar con su flujo de trabajo existente para permitir la inspección continua de código en todas las ramas de su proyecto y solicitudes de extracción.
- **SPARQL:** (*SPARQL Protocol and RDF Query Language*) es un lenguaje y protocolo de consulta para *RDF*.
- **Sprint:** Período en el cual se lleva el desarrollo de una tarea.

- **Tarsier**: Herramienta para la visualización interactiva en 3D de grafos RDF.
- **UML**: (*Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.
- **Vista**: La vista es responsable de definir la estructura, el diseño y la apariencia de lo que el usuario ve en la pantalla. Idealmente, cada vista se define en XAML, con un código subyacente limitado que no contiene la lógica de negocios. Sin embargo, en algunos casos, el código subyacente podría contener lógica de la interfaz de usuario que implementa el comportamiento visual que es difícil de expresar en XAML, como animaciones.
- **Visual Studio**: El *IDE* de Visual Studio es un panel de inicio creativo que se puede usar para editar, depurar y compilar código y, después, publicar una aplicación. Más allá del editor estándar y el depurador que proporcionan la mayoría de *IDE*, Visual Studio incluye compiladores, herramientas de finalización de código, diseñadores gráficos y muchas más características para facilitar el proceso de desarrollo de software.
- **W3C**: (*World Wide Web Consortium*) Consorcio internacional que produce recomendaciones para la *WWW*.
- **WWW**: (*World Wide Web*) Sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet.
- **Xamarin**: Xamarin es una plataforma de código abierto para compilar aplicaciones modernas y de rendimiento para iOS, Android y Windows con .NET.
- **Xamarin.Forms**: Xamarin.Forms es un marco de interfaz de usuario de código abierto. Xamarin.Forms permite a los desarrolladores compilar aplicaciones de Android, iOS y Windows desde un único código base compartido.
- **XAML**: (*eXtensible Application Markup Language*) Lenguaje basado en XML creado por *Microsoft* como una alternativa a código de programación para la creación de instancias e inicialización de objetos, y la organización de esos objetos en jerarquías de elementos primarios y secundarios.
- **Web Semántica**: “*Extensión de la actual web en la que a la información disponible se le otorga un significado bien definido que permita a los ordenadores y las personas trabajar en cooperación. Se basa en la idea de tener datos en la web definidos y vinculados de modo que puedan usarse para un descubrimiento, automatización y reutilización entre varias aplicaciones.*”

## 1.5. Metodología

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam maximus facilisis odio vel dignissim. Integer sed tincidunt neque. Cras vitae nisi odio. Morbi in pellentesque

nulla. Nam sagittis leo nec ex ultricies, eget hendrerit velit tincidunt. Phasellus fringilla varius tellus, in aliquam purus convallis eget. Donec mattis nisi turpis, vitae ultrices lectus ornare at. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in rutrum eros. Duis ut lacinia justo. Donec fringilla velit eu sem congue cursus. Etiam porttitor, justo gravida porttitor ultrices, eros sem lacinia ex, eget bibendum enim dolor non ligula. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

## 1.6. Estructuración del documento

El documento que se presenta está estructurado en una colección de capítulos, en los que se describen de manera precisa y detallada todas y cada una de las etapas por las que ha pasado el proyecto, desde su inicio hasta su conclusión. A continuación se muestra un breve resumen de los contenidos de cada capítulo:

- **Capítulo 1. Introducción.** El capítulo inicial consiste en una introducción al proyecto, explicando los antecedentes a su desarrollo, así como la motivación para ponerlo en práctica, los objetivos que debe cumplir y un glosario de términos para facilitar la comprensión del presente documento.
- **Capítulo 2. Conceptos básicos.** Tras la introducción, se abordan los fundamentos necesarios para simplificar la lectura de este documento. A su vez, se exponen las diferentes tecnologías de las que se han aplicado para la consecución y puesta en marcha de este proyecto.
- **Capítulo 3. Planificación del proyecto.** Este capítulo engloba toda la información relevante a aspectos de suma importancia tales como la evaluación de riesgos o la planificación temporal del proyecto.
- **Capítulo 4-8. Análisis, Diseño, Codificación y Pruebas.** Este conjunto de capítulos profundizan en los diversos aspectos y fases que comprenden el desarrollo de un producto haciendo uso de una *metodología ágil*. Esto posibilita analizar y enriquecer el producto en su desarrollo, mejorando así el resultado final.
- **Capítulo 9. Manual de Instalación.** Aquí se ha desarrollado un documento con las instrucciones necesarias para realizar la instalación de la aplicación.
- **Capítulo 10. Manual de Usuario.** Se ha redactado un manual de usuario para la aplicación, cuyo objetivo es garantizar un uso eficiente y responsable de la misma por parte de los usuarios.
- **Capítulo 11. Conclusiones.** El último capítulo es un breve repaso sobre el desarrollo del proyecto, que incluye una opinión personal y un apartado de posibles mejoras que se podrían realizar al proyecto en un futuro.

## CAPÍTULO 2

### Antecedentes

## 2.1. Estado del Arte

### 2.1.1. Situación actual de la tecnología

Con el objetivo de simplificar el proceso de creación de personajes en los juegos de rol tradicionales, se han originado multitud de aplicaciones con diferentes funciones y finalidades.

Algunas de las aplicaciones son referencias completas de los juegos, que sirven como elementos de consulta accesibles, rápidos y precisos. Un ejemplo de ello es **5e Character**, que es una referencia completa de personajes para *Dragones y Mazmorras*, 5ª Edición.

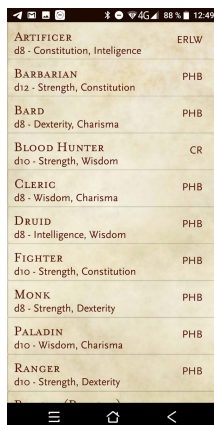


Figura 2.1: **5e Character**: Pantalla de selección de clases

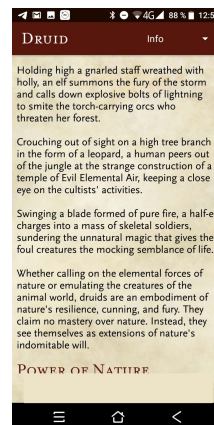


Figura 2.2: **5e Character**: Información de la clase *Druida*

También podemos encontrar aplicaciones como *RPG Simple Dice* que realizan simulaciones de lanzamiento de dados, en caso de que no dispongamos de dados físicos.

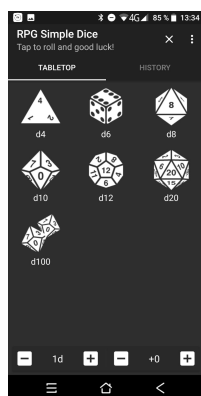


Figura 2.3: *RPG Simple Dice*: Pantalla de selección de dados

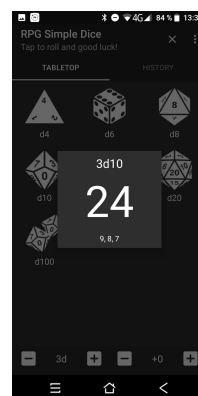


Figura 2.4: *RPG Simple Dice*: Ejemplo de lanzamiento de dados

Otras en cambio, proporcionan algunas herramientas que simplifican algunos cálculos que resultan tediosos durante el transcurso de la partida, como es el caso de *BattleTrack*.

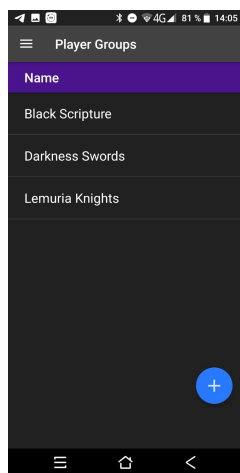


Figura 2.5: *BattleTrack*: Pantalla de selección de grupo

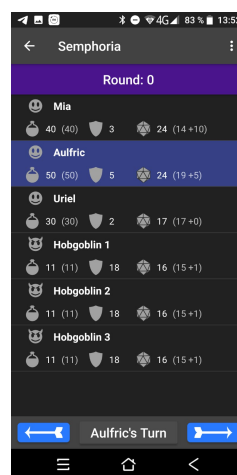


Figura 2.6: *BattleTrack*: Ejemplo de combate



Con el fin de ayudar en la ambientación, aplicaciones como *RPGSound* aportan bibliotecas de sonido que se pueden utilizar durante la representación de la partida para sumirse en ella.



Figura 2.7: *RPGSound*: Menú principal

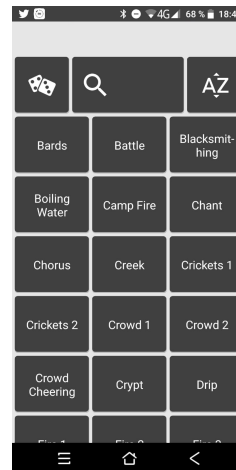


Figura 2.8: *RPGSound*: Menú de *Ambiente Sostenido*

Finalmente, quedan las aplicaciones conocidas como *generadores de personaje*, que permiten al usuario crear personajes para formar parte de una partida de rol, y acceder a esa información de forma rápida. Un buen ejemplo de esto es *RPG Character Sheet*.

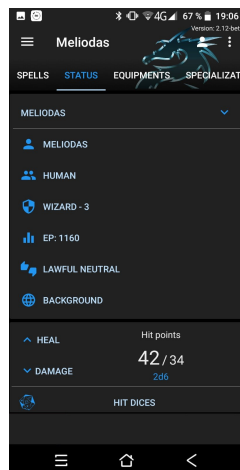


Figura 2.9: *RPG Character Sheet*: Pantalla de *Estado*

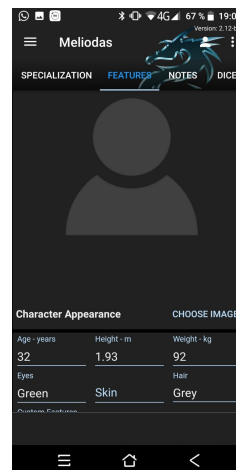


Figura 2.10: *RPG Character Sheet*: Pantalla de *Features*

### 2.1.2. Crítica al estado del arte

Tal y como se ha comentado previamente, existe un holgado abanico de aplicaciones cuya meta es mejorar y/o simplificar aspectos en lo referente a los juegos de rol, y aunque cumplen con su propósito, a veces no resultan tan efectivas como debieran.

Esto puede deberse a que tras dedicar el tiempo y esfuerzo necesarios para desarrollar la aplicación, el estudio del juego ha aprovechado ese tiempo de producción para revisar el juego y editarlo, realizando modificaciones que provocan que *la aplicación quede obsoleta en poco tiempo*.

Otro inconveniente es que las aplicaciones que requieren mucha información específica, como los generadores de personaje, pueden llegar a *resultar muy complejas*, provocando que el usuario considere que el esfuerzo que tiene que dedicar para aprender cómo utilizarla es mayor que el de realizar el proceso manualmente.

También existen aplicaciones que no contemplan la reutilización de la información que han producido para efectuar operaciones que mejoren la jugabilidad, por lo que el usuario no le ve provecho a emplear dichas aplicaciones.

### 2.1.3. Propuesta

Tomando en consideración los aspectos comentados en el apartado anterior, se ha formulado la siguiente solución:

- La aplicación permitirá al usuario seleccionar en qué juego y versión desea que trabaje la aplicación.
- La aplicación distribuirá el proceso de creación de personajes del juego seleccionado en un conjunto modular secuenciado de pasos a seguir, que guíe al usuario en el proceso y resulte sencillo de utilizar.
- La aplicación tendrá la capacidad de reutilizar los datos producidos para operar con ellos, ofreciendo al usuario información adicional práctica para algunos aspectos del juego escogido.

## CAPÍTULO 3

---

### Análisis de requisitos del sistema

---

A la vista de lo expuesto previamente en el apartado *Crítica al estado del arte*, se hace necesario el planteamiento de un análisis de la problemática razonada para poder esbozar una propuesta que permita resolver los aspectos descritos en el apartado anteriormente mencionado.

#### 3.1. Catálogo de actores

A causa de que el sistema se define como una aplicación móvil, se alcanza la conclusión de sólo considerar un único actor en este proyecto, que es el usuario de la aplicación.

En la presente sección vamos a proceder a realizar un análisis de requisitos del sistema, que recoge y describe el conjunto de requisitos específicos del sistema que se va a desarrollar.

En primera instancia, se presentarán los requisitos agrupados en conjuntos funcionales del sistema. Posteriormente, se describirán los casos de uso en el próximo capítulo.

Para ello, se hará una diferenciación entre *requisitos funcionales*, que son aquellos que detallan la funcionalidad del sistema, y *requisitos no funcionales*, que refieren a otros aspectos del software que deben ser satisfechos.

#### 3.2. Requisitos funcionales

Un requisito funcional especifica una función concreta del sistema o de alguno de sus componentes. A continuación se muestran los requisitos funcionales, estructurados según el módulo del sistema al que refieren.

El sistema precisa de una lógica estructurada y compleja que permita procesar información de diferentes fuentes, de manera que los procesos del sistema se adecuen a su contenido.

- **OBJ-001:** El usuario podrá seleccionar un juego concreto (juego activo) para poder acceder a la información relacionada con el mismo.
- **OBJ-002:** El usuario podrá crear un personaje para el juego activo, mediante un proceso guiado paso a paso
- **OBJ-003:** El usuario podrá seleccionar un personaje (personaje activo) de todos los existentes para el juego activo
- **OBJ-004:** El usuario podrá visualizar la información del personaje seleccionado del juego activo.
- **OBJ-005:** El usuario podrá eliminar un personaje ya creado del juego activo.
- **OBJ-006:** El usuario podrá realizar cálculos con los valores de las habilidades del personaje activo.

### 3.3. Requisitos no funcionales

Un requisito no funcional es una propiedad o cualidad que no forma parte de los fundamentos del sistema, pero es necesario para que el producto cumpla con su cometido apropiadamente.

Para la declaración de requisitos no funcionales, se establecerán como base los requisitos indicados en las normas *IEEE Std. 830* e *ISO/IEC 25010 (SQuaRE)*:

- **Adecuación funcional:** La aplicación debe cumplir con todos los requisitos necesarios, de manera que sea completo y correcto funcionalmente.
- **Seguridad:** El sistema no requiere asegurar la información que procesa, debido a que no contiene información sensible del usuario en ningún momento, ni realiza conexión externa alguna para obtener información.
- **Compatibilidad:** La aplicación deberá ser compatible con los ficheros que contienen la información de los juegos que formarán parte del sistema.
- **Usabilidad:** El sistema debe disponer de una interfaz de usuario intuitiva y fácil de manejar, de manera que pueda ser utilizado por usuarios sin conocimientos técnicos ni avanzados de informática. La curva de aprendizaje deberá ser lo más reducida posible, de manera que personas de cualquier ámbito puedan hacer uso del mismo.

- **Fiabilidad:** La aplicación deberá estar libre de errores que influyan negativamente en su uso normal. Debido a que la aplicación depende de información incluida por terceros, será necesario comprobar que dicha información es compatible con la aplicación.
- **Eficiencia:** El sistema debe evitar en la medida de lo posible el uso de información redundante para poder asegurar su funcionamiento cuando se introduzcan juegos de alta complejidad que requieran un elevado uso de recursos.
- **Mantenibilidad:** Este apartado representa la capacidad del producto software para ser modificado efectiva y eficientemente. Esto será posible debido al desarrollo de código limpio y bien documentado, al diseño y la implementación modular del mismo. Se plantea el uso de patrones de arquitectura de software, tales como MVVM.
- **Portabilidad:** El sistema está diseñado para su uso en dispositivos móviles, aunque no se descarta una futura ampliación para introducirlo en otro tipo de dispositivos. La implementación está realizada únicamente para sistemas *Android*, ya que no se dispone de las herramientas necesarias para el despliegue en *Mac OS*.

### 3.4. Reglas de negocio

Toda funcionalidad está relacionada con el único actor del proyecto, el usuario de la aplicación.

### 3.5. Análisis de las soluciones

En esta sección, se ofrece un estudio del arte de las diferentes alternativas tecnológicas que permitan satisfacer los requerimientos del sistema, para optar por una de las opciones planteadas, que será dispuesta como base para el software a desarrollar.

Con este motivo, hemos optado por recoger algunas de las tecnologías existentes para realizar desarrollo de aplicaciones móviles.

#### 3.5.1. Android Studio

Android Studio es el entorno de desarrollo integrado (*IDE*) oficial para el desarrollo de apps para Android, basado en *IntelliJ IDEA* de *JetBrains* y ha sido publicado gratuitamente a través de la Licencia Apache 2.0. Disponible para las plataformas *Windows*, *macOS* y *GNU/Linux*. Basado en el lenguaje *Java*, no tiene herramientas nativas para trabajar directamente con *RDF* y *OWL*. Para suplir este obstáculo, haríamos uso del framework libre *Apache Jena*, cuya API permite trabajar con *RDF*, consiguiendo vincular el desarrollo en aplicaciones móviles con el uso de ontologías.

htb

Cuadro 3.1: Comparativa entre *JavaScript* y *C#*

### 3.5.2. React Native

React Native es un *framework* para el desarrollo de aplicaciones móviles de código abierto desarrollado por *Facebook*. Se utiliza para desarrollar aplicaciones para *Android*, *iOS*, *Web* y *UWP* permitiendo a los desarrolladores usar *React* con funcionalidades nativas de las plataformas. Al igual que *Android Studio*, React Native no tiene herramientas nativas para el desarrollo de ontologías, de manera que haríamos uso de bibliotecas tales como *rdflib.js* para poder proceder al tratamiento de las ontologías.

### 3.5.3. Xamarin

Xamarin es una plataforma de código abierto para compilar aplicaciones modernas y con mejor rendimiento para *iOS*, *Android* y *Windows* con **.NET**. Xamarin es una capa de abstracción que administra la comunicación de código compartido con el código de plataforma subyacente. Xamarin dispone de una biblioteca conocida como *RDFSharp*, que permite generar y procesar ontologías en formatos *RDF* y *OWL*.

## 3.6. Solución propuesta

Tras considerar las opciones planteadas en el apartado anterior, se ha considerado descartar *Android Studio* en primer lugar, ya que sólo permite el desarrollo en *Android*, mientras que las otras soluciones permiten realizar el desarrollo en varias plataformas.

Una vez desechada una de las opciones, se ha comprobado que las soluciones restantes son compatibles con el proyecto, y prácticamente generan el mismo resultado. Por ello, en vez de considerar las plataformas, se ha realizado una comparación en función al lenguaje de programación con el que se trabaja en cada una de ellas, que son ***JavaScript*** en *React Native*, y ***C#*** para *Xamarin*. Esta comparativa se realizará en forma tabular.