



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN

**Desarrollo de aplicación web de apoyo al juego Dungeons
and Dragons 5ª Edición**

SERGIO CAMILO FERNÁNDEZ AYANCÁN

Profesor Guía: DANIEL MORENO CÓRDOVA

Memoria para optar al título de
Ingeniero Civil en Computación

Curicó – Chile
Agosto, 2018

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Two circular official stamps and handwritten signatures in blue ink. The left stamp is from the 'DIRECCIÓN SISTEMA DE BIBLIOTECAS' of the 'UNIVERSIDAD DE TALCA'. The right stamp is from the 'SISTEMA DE BIBLIOTECAS CAMPUS CURICÓ' of the 'UNIVERSIDAD DE TALCA'.

Curicó, 2022

Dedicado a mi familia.

AGRADECIMIENTOS

Agradezco a mis padres Luis y Nancy, y mi hermana Javiera. Por su apoyo incondicional durante toda mi vida. Gracias por todo.

A los amigos que hice durante estos años Mati, Josema, Seba, Cortinez, Victor, Marcelo, Adolfo y Marchant. Gracias por las largas noches de estudio y carrete.

Al profesor guía Daniel por ayudarme y orientarme durante este largo tiempo en el desarrollo de esta memoria.

A mi jefe de Soporte DTI Curicó Don Hernando por permitirme trabajar durante 5 años.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	VII
Índice de Tablas	X
Resumen	XI
1. Introducción	12
1.1. Descripción del contexto	12
1.2. Descripción del problema	13
1.3. Propuesta de solución	14
1.4. Objetivos	15
1.4.1. Objetivo general	15
1.4.2. Objetivos específicos	15
1.5. Alcance	15
1.6. Definiciones, acrónimos y abreviaturas	16
2. Marco teórico	17
2.1. Dungeons and Dragons	17
2.1.1. Roles de los participantes	17
2.1.2. Personajes	18
2.1.3. Combate	19
2.2. Manuales de juego de Dungeons and Dragons	19
2.2.1. Player's Handbook	20
2.2.2. Dungeon Master's Guide	20
2.2.3. Monster Manual	21
2.3. Propuestas similares	21

2.4.	Metodología de desarrollo de software	23
2.4.1.	Extreme Programming XP	23
2.5.	Patrones de diseño	24
2.5.1.	Modelo Vista Controlador	24
2.6.	Tecnologías y frameworks utilizados	25
2.6.1.	.NET Framework	25
2.6.2.	ASP.NET	25
2.6.3.	Razor	26
2.6.4.	Materialize	27
2.6.5.	JQuery	27
2.6.6.	SQL Server	28
2.6.7.	Entity Framework	28
2.7.	Pruebas	28
2.7.1.	Prueba de caja negra	28
2.7.2.	Prueba de usabilidad USE	29
3.	Metodología de desarrollo	31
3.1.	Metodología	31
3.2.	Planificación	31
3.2.1.	Etapas previas	32
3.2.2.	Iteraciones	32
4.	Requisitos	36
4.1.	Roles de jugadores	36
4.2.	Historias de usuario	36
4.3.	Casos de uso	38
4.3.1.	Caso de uso nivel cero	38
4.3.2.	Administrar NPC	39
4.3.3.	Administrar PC	40
4.3.4.	Administrar encuentro	41
4.3.5.	Ejecutar encuentro	42
5.	Diseño	43
5.1.	Arquitectura de la aplicación	43
5.1.1.	Ambiente operacional	43

5.1.2.	Arquitectura física	45
5.1.3.	Arquitectura lógica	46
5.1.4.	Componentes	47
5.2.	Diseño de la aplicación	47
5.2.1.	Administrar aplicación	48
5.2.2.	Administrar NPC	49
5.2.3.	Administrar PC	61
5.2.4.	Administrar Encuentro	64
6.	Desarrollo	67
6.1.	Iteraciones	67
6.1.1.	Iteración 0: Análisis de los manuales de juego	67
6.1.2.	Iteración 1: Documentación de apoyo	68
6.1.3.	Iteración 2: Desarrollo de módulo NPC (parte 1)	68
6.1.4.	Iteración 3: Desarrollo de módulo NPC (parte 2)	69
6.1.5.	Iteración 4: Desarrollo de módulo NPC (parte 3)	69
6.1.6.	Iteración 5: Desarrollo de módulo PC (parte 1)	70
6.1.7.	Iteración 6: Desarrollo de módulo PC (parte 2)	71
6.1.8.	Iteración 7: Desarrollo de módulo Encuentro (parte 1)	71
6.1.9.	Iteración 8: Desarrollo de módulo Encuentro (parte 2)	72
6.1.10.	Iteración 9: Pruebas de usabilidad	73
7.	Pruebas	74
7.1.	Pruebas de caja negra	74
7.1.1.	Administrar NPC	74
7.1.2.	Administrar PC	80
7.1.3.	Administrar aplicación	82
7.1.4.	Administrar Encuentro	83
7.2.	Pruebas de usabilidad	86
7.2.1.	Resultados	87
8.	Conclusiones y trabajo futuro	89
	Bibliografía	90

Anexos

A: Diagramas	94
A.1. Diagrama de clase: NPC	95
A.2. Diagrama de clase: PC	96
B: Pruebas de usabilidad	97
B.1. Manual de uso	97
B.1.1. Inicio de sesión	97
B.1.2. Crear NPCs	98
B.1.3. Crear PC	102
B.1.4. Crear encuentros	107
B.1.5. Ejecutar encuentros como DM	111
B.1.6. Ejecutar encuentros como Jugador	113
B.1.7. Agradecimientos y cuestionario	115
B.2. Participantes prueba de usabilidad	116
B.3. Resultados cuestionario USE	117
C: GNU General Public License	134

ÍNDICE DE FIGURAS

	página
2.1. <i>Statblock</i> de un “Ettin”. Fuente: Monster Manual.	18
2.2. Interacción en patrón: Modelo-Vista-Controlador.	24
2.3. Afirmación número 25 del formulario USE Adaptado.	29
4.1. Diagrama de caso de uso: Interacción de los usuarios.	38
4.2. Diagrama de caso de uso: Administrar NPC.	39
4.3. Diagrama de caso de uso: Administrar PC.	40
4.4. Diagrama de caso de uso: Administrar encuentro.	41
4.5. Diagrama de caso de uso: Ejecutar encuentro.	42
5.1. Diagrama de front-end.	44
5.2. Diagrama de back-end.	45
5.3. Diagrama de arquitectura física.	45
5.4. Diagrama de arquitectura lógica (MVC).	46
5.5. Diagrama de componentes.	47
5.6. Diagrama de flujo (parte I).	48
5.7. Representación clase User.	49
5.8. Representación de diagrama de clases reducido.	49
5.9. Extracto diagrama de clases: Clase NPC.	50
5.10. Extracto diagrama de clases: Clases AbilityScore y SavingThrow.	52
5.11. Extracto diagrama de clases: Clase Armor.	53
5.12. Tabla cálculo de <i>Armor Class</i> . Fuente: Player’s Handbook.	53
5.13. Extracto diagrama de clases: Clase Size.	54
5.14. Extracto diagrama de clases: Clase HitPoint.	54
5.15. <i>Hit Dice</i> por tamaño. Fuente: Monster Manual.	55
5.16. Extracto diagrama de clases: Clase Speed.	55
5.17. Extracto diagrama de clases: Clase Skill.	56
5.18. Relación entre <i>ability scores</i> y <i>skills</i> . Fuente: Player’s Handbook.	56
5.19. Extracto diagrama de clases: Clases Damage y Condition.	57
5.20. Extracto diagrama de clases: Clase Sense.	57
5.21. Extracto diagrama de clases: Clase Language.	58

5.22. Extracto diagrama de clases: Clase Challenge.	58
5.23. Puntos de experiencia por <i>Challenge Rating</i> . Fuente: Monster Manual. . .	59
5.24. <i>Proficiency Bonus</i> por <i>Challenge Rating</i> . Fuente: Monster Manual.	59
5.25. Extracto diagrama de clases: Clase SpecialTrait.	60
5.26. Extracto diagrama de clases: Clase Action.	60
5.27. Clase PC (versión acotada).	61
5.28. Diagrama de flujo (parte II).	65
7.1. Resultados pruebas de usabilidad.	87
A.1. Diagrama de clases: NPC.	95
A.2. Diagrama de clases: PC.	96
B.1. Usuarios prueba de usabilidad.	117
B.2. Email prueba de usabilidad.	117
B.3. Pregunta 1.	118
B.4. Pregunta 2.	118
B.5. Pregunta 3.	119
B.6. Pregunta 4.	119
B.7. Pregunta 5.	120
B.8. Pregunta 6.	120
B.9. Pregunta 5.	121
B.10. Pregunta 7.	121
B.11. Pregunta 8.	122
B.12. Pregunta 9.	122
B.13. Pregunta 10.	123
B.14. Pregunta 11.	123
B.15. Pregunta 12.	124
B.16. Pregunta 13.	124
B.17. Pregunta 14.	125
B.18. Pregunta 15.	125
B.19. Pregunta 16.	126
B.20. Pregunta 17.	126
B.21. Pregunta 18.	127
B.22. Pregunta 19.	127

B.23. Pregunta 20.	128
B.24. Pregunta 21.	128
B.25. Pregunta 22.	129
B.26. Pregunta 23.	129
B.27. Pregunta 24.	130
B.28. Pregunta 25.	130
B.29. Pregunta 26.	131
B.30. Pregunta 27.	131
B.31. Pregunta 28.	132
B.32. Pregunta 29.	132
B.33. Pregunta 30.	133

ÍNDICE DE TABLAS

	página
4.1. Rol y descripción de los participantes en D&D 5e.	36
5.1. Diferencias entre <i>race</i> : “Hill Dwarf” y “Human”.	62
5.2. Diferencias entre <i>class</i> : “Wizard” y “Cleric”.	63
5.3. Diferencias entre <i>background</i> : “Acolyte” y “Noble”.	64
5.4. Ejemplo de salida de crear Encuentros.	65
7.1. Prueba de caja negra: Propiedad name.	74
7.2. Prueba de caja negra: Propiedad size.	75
7.3. Prueba de caja negra: Propiedad Armor Class.	75
7.4. Prueba de caja negra: Propiedad Hit Points.	76
7.5. Prueba de caja negra: Propiedad Ability Score Modifier.	76
7.6. Prueba de caja negra: Propiedad Saving Throw.	77
7.7. Prueba de caja negra: Propiedad Skill.	78
7.8. Prueba de caja negra: Propiedad Speed.	78
7.9. Prueba de caja negra: Propiedad Challenge.	79
7.10. Prueba de caja negra: Propiedad Special Traits.	79
7.11. Prueba de caja negra: Propiedad Action.	80
7.12. Prueba de caja negra: Cambio de <i>race</i>	80
7.13. Prueba de caja negra: Cambio de <i>class</i>	81
7.14. Prueba de caja negra: Cambio de <i>background</i>	81
7.15. Prueba de caja negra: Registro de usuario.	82
7.16. Prueba de caja negra: Validación de usuario.	82
7.17. Prueba de caja negra: Inicio de sesión usuario.	83
7.18. Prueba de caja negra: Agregar participante a encuentro.	83
7.19. Prueba de caja negra: Agregar NPC a encuentro.	84
7.20. Prueba de caja negra: Agregar PC a encuentro.	84
7.21. Prueba de caja negra: Realizar daño a un personaje.	85
7.22. Prueba de caja negra: Curar a un personaje.	85
7.23. Prueba de caja negra: Agregar efecto a un personaje.	86
7.24. Prueba de caja negra: Avanzar al siguiente turno en combate.	86

RESUMEN

Dungeons and Dragons es un exitoso juego de rol lanzado al mercado por primera vez en 1974, en el cual los jugadores construyen historias colaborativamente, e interactúan en un mundo fabricado a través de reglas y el uso de dados de distinto tipo.

Con el pasar de los años se ha relacionado el juego con las plataformas digitales a través de videojuegos, en donde su popularidad ha aumentado cada vez más. Pese a esto, los jugadores no han dejado de jugarlo con lápiz y papel, lo que implica una larga y tediosa preparación previa. También, durante la realización de las partidas a los jugadores les toma tiempo anotar los cambios en las estadísticas de sus personajes.

Para solucionar este problema se propone desarrollar una aplicación web que permita a los usuarios simplificar el trabajo que realizan manualmente. El objetivo es apoyar a los jugadores en la creación de sus personajes. Además, de la generación y el seguimiento de combates, permitiendo así que los jugadores se centren en realizar su rol o actuación mientras juegan, haciendo las partidas más dinámicas.

Existen propuestas similares que generalmente su uso está sujeto a altos costos económicos. Por otro lado, existen otras alternativas en la que su uso no requiere de pagos, pero éstas carecen de interactividad en sus componentes.

En esta memoria, se presenta el desarrollo de una aplicación web de apoyo al juego Dungeons and Dragons 5ª Edición. La aplicación fue desarrollada usando la metodología ágil llamada Extreme Programming, que sigue un conjunto de reglas sencillas y que permite un flujo constante de retroalimentación.

Para la construcción de la aplicación se especificaron los requisitos mediante historias de usuarios. A través de estas, se generó el diseño y la arquitectura de la aplicación.

Como resultado del desarrollo de esta memoria, se generó una aplicación web que permite a los jugadores crear personajes del juego, y generar y realizar combates interactivos donde se calculan los cambios en sus estadísticas.

La aplicación fue validada a través de pruebas de caja negra y pruebas de usabilidad con el fin de asegurar la utilidad, satisfacción y facilidad de uso que la aplicación entrega a los jugadores.

Como trabajo futuro se propone agregar nuevas funcionalidades a la aplicación que permitan la creación de elementos adicionales para apoyar a los jugadores.

1. Introducción

1.1. Descripción del contexto

Los primeros juegos de mesa fueron creados hace siglos, con el fin de brindar entretenimiento a quienes los jugaban. A lo largo de los años, estos juegos han ido diversificándose en distintas áreas como estrategia, juegos de cartas, de tablero, entre otros. Uno de estos tipos son los llamados juegos de rol, en donde cada jugador encarna o actúa un personaje que interactúa en universos creados por otros jugadores. Es así como en los años 70 surgen propuestas para satisfacer a gran cantidad de usuarios que buscan tener aventuras fantásticas en que comparten con sus amigos.

Dungeons and Dragons (D&D) es un exitoso juego de rol publicado en 1974 en tres manuales, los cuales venían en una caja de madera [8]. El objetivo principal es que los jugadores de este construyan historias de manera colaborativa, realizando misiones como por ejemplo batallas contra enemigos mortales, realizar rescates, entre otros [6]. A través del tiempo el juego ha ido evolucionando en distintas versiones o ediciones, de las cuales la última entrega fue publicada en 2014 llamada D&D 5ª edición (D&D 5e).

En años posteriores, el juego se ha relacionado de distintas maneras con el área computacional. En 1988 comienza la interacción oficial del juego con computadoras mediante el lanzamiento del juego “*Pool of Radiance*” basado en la versión Advanced Dungeons and Dragons (AD&D). Por primera vez un videojuego contenía todas las reglas de un juego de rol de lápiz y papel sin perder su esencia [16]. Con el pasar de los años ha aumentado la interacción entre las plataformas digitales y el juego, debido a que tiene una gran popularidad y se ha creado una gran comunidad mundial de jugadores. Ejemplos de esto son los juegos CRPG (Computer Role-Playing Game) como Baldur’s Gate [10], y Neverwinter Nights [12], entre otros. Sin embargo, el juego de lápiz y papel no ha perdido fuerza, y

prueba de ello es que existen varias herramientas computacionales que han surgido para apoyarlo.

1.2. Descripción del problema

En D&D, los jugadores crean los personajes que usarán para desenvolverse en el mundo del juego. El Dungeon Master (DM) crea criaturas llamadas Non-Player Character (NPC), las cuales representan a habitantes del mundo del juego. En el proceso de creación de estos NPCs el DM agrega y calcula diferentes aspectos que caracterizan a estos personajes, tales como sus puntos de vida (HP), su armadura (Armor class), velocidad y tipo de movimiento (Speed), etc. Además, existe una lista con las acciones que pueden realizar en combate, las cuales pueden hacer daño o curar a los demás personajes que se encuentren en la partida, así como modificar las estadísticas de los afectados, o brindar oportunidades de “rolear” (actuar el rol del personaje). Por ejemplo, un dragón (NPC) puede atacar a un humano (Player Character) disminuyendo sus puntos de vida. Por otro lado, un brujo podría mediante el efecto de sus maldiciones, agregar una condición al dragón, como por ejemplo envenenarlo, cegarlo, paralizarlo, etc.

Cada jugador crea su personaje llamado Player Character (PC), los cuales los representan en el juego. Para ello realizan un proceso similar a la creación de NPCs, donde se seleccionan características como la raza (por ejemplo, humano o enano) o clase (por ejemplo, guerrero o mago). Dependiendo de esta selección su PC adquiere distintos atributos con los cuales puede interactuar en el juego [27].

Los NPCs y PCs interactúan simulando la vida real. Sin embargo, hay situaciones en las que el resultado es incierto o depende de un reto, los cuales se denominan “encuentros”. Para llevar a cabo un encuentro el Dungeon Master crea un ambiente de juego, en donde decide qué criaturas (NPCs) previamente creadas por él participarán de la partida y también tiene un resumen de las características de los personajes de los jugadores (PCs). A partir de esto, se calcula la iniciativa de ser necesaria. La iniciativa define el orden que cada personaje tanto NPCs y PCs, actuará a lo largo del combate.

Durante el desarrollo de un encuentro cada personaje interactúa en un turno, en el cual puede realizar distintas acciones que tienen efecto sobre él mismo o sobre otro participante. Para ello, el DM define las acciones de los NPCs y los jugadores de los PCs. En general se sigue los procedimientos seleccionar la acción de un personaje, determinar el o los objetivos, resolver el daño causado o efecto aplicado hacia el personaje objetivo y

finalmente repetir estos pasos hasta que se dicte el final del encuentro.

Todos los eventos que ocurren en los encuentros (por ejemplo, en un combate, restar HP a un NPC) son anotados por el DM, lo que implica una gran cantidad de aspectos que deben ser gestionados por él. Sumado a esto, en la etapa previa al encuentro debe crear todos los NPCs que se usarán, anotando en los *statblock* las características y realizando algunos cálculos. Análogamente los demás jugadores crean sus personajes en hojas escribiendo sus características. La cantidad de material (papel, notas) podría ser muy grande, y se vuelve engorroso ejecutar un simple encuentro.

El problema que aborda esta memoria es la necesidad de una herramienta cuya principal función sea ayudar a los jugadores a crear sus personajes NPCs y PCs de manera más sencilla. Además, la aplicación permitirá organizar combates en que los participantes puedan manejar y visualizar las estadísticas de sus personajes en tiempo real y sin tener que utilizar lápiz y papel.

1.3. Propuesta de solución

Para resolver el problema se propone diseñar y desarrollar una aplicación web de apoyo al juego D&D 5e, que permita:

- La creación de NPCs siguiendo los lineamientos establecidos por las reglas de creación de criaturas de los manuales del juego. Como resultado se obtendrán un bloque de estadísticas (*Statblock*) de la misma manera en que se realiza por parte de los jugadores.
- La creación PCs siguiendo los lineamientos establecidos por las reglas de creación de personajes de los manuales del juego. Como resultado se obtendrán un bloque de estadísticas que estará asociado a los jugadores que los creen.
- La realización de encuentros o combates de manera interactiva tal que los NPCs y PCs interactúen en turnos, de tal forma que el Dungeon Master controle a los personajes quienes pueden realizar acciones que tengan efecto sobre él mismo, o sobre otro, y éstos deben verse reflejados en el estado global del encuentro.

Cabe mencionar que la solución estará enfocada a la creación y el manejo de las estadísticas de los personajes y a los cambios que sufran en los encuentros, no a representar el rol y las situaciones que protagonizan los usuarios.

1.4. Objetivos

1.4.1. Objetivo general

- Apoyar a los jugadores del juego de mesa “Dungeons and Dragons 5ª Edición” mediante una aplicación web para la creación de personajes y para la realización de encuentros de manera interactiva en tiempo real.

1.4.2. Objetivos específicos

- Permitir a los jugadores crear NPCs siguiendo los lineamientos descritos en el Monster Manual y el Dungeon Master’s Guide (5e).
- Permitir a los jugadores crear PCs siguiendo los lineamientos simplificados descritos en el Player’s Handbook (5e).
- Facilitar a los jugadores a llevar a cabo encuentros de manera interactiva en tiempo real.
- Evaluar la aplicación a partir de métricas de usabilidad.

1.5. Alcance

El diseño de la aplicación está limitado sólo a:

- La creación de NPCs y PCs, siguiendo las normas establecidas en los manuales del juego (Monster Manual, Player’s Handbook)
- La simulación de encuentros (combates) entre NPCs y PCs.
- No se planea implementar ninguna funcionalidad adicional incluida en los manuales antes mencionados, ni introducida por suplementos publicados posteriormente.
- La aplicación a desarrollar es complementaria al juego Dungeons and Dragons 5ª Edición y en ningún caso es un sustituto completo a este.
- La interfaz de la aplicación será en inglés.

1.6. Definiciones, acrónimos y abreviaturas

- **D&D 5e:** Dungeons and Dragons 5^a Edición.
- **DM:** Dungeon Master. Es un tipo de jugador que administra las partidas.
- **NPC:** Non-Player Character. Son los personajes que habitan en el mundo creado por los Dungeon Master.
- **PC:** Player Character. Son los personajes usados por los jugadores para representarse dentro del mundo del juego.
- **Statblock:** Bloque de estadísticas de los NPCs.
- **USE:** *Usefulness, Satisfaction and Ease of use*. Es un cuestionario de usabilidad publicado por A. Lund [23].

2. Marco teórico

En el presente capítulo se describen los conceptos principales respecto a D&D, soluciones similares a la planteada, las tecnologías y frameworks, y patrones de diseño que se utilizaron en el desarrollo de la aplicación web. Además, se presentan las herramientas que se utilizan en la fase de pruebas de ésta.

2.1. Dungeons and Dragons

“Dungeons and Dragons” es un juego de rol creado por Gary Gygax y Dave Arneson en los años 70s. Se convirtió en el primer juego de rol de fantasía, en que los personajes son parte de una historia personalizada [8]. Estos personajes son guiados a través de una serie de eventos por los participantes del juego. Existen dos roles que los jugadores llevan a cabo al momento de participar en el mundo de D&D, que son: el Dungeon Master (DM) y jugador.

2.1.1. Roles de los participantes

Dungeon Master: Es el encargado de crear NPCs y ejecutar las aventuras donde participan los personajes NPCs y PCs a través de sesiones de juego. Una aventura puede requerir varias sesiones de juego para ser completada, y a su vez muchas aventuras forman una campaña que puede durar meses o años. Como administrador de las partidas, el DM decide cómo se van desarrollando las historias mientras el juego transcurre ubicando criaturas en distintas locaciones, poniendo trampas a los participantes o revelando tesoros. Además, hace cumplir las reglas del juego dirigiendo de manera objetiva las acciones que realizan los participantes basado en un conjunto de reglas flexibles [25].

Jugador: Los jugadores crean personajes (PC) que los representan en el juego y que junto a sus compañeros se unen en equipos que enfrentan las diferentes aventuras que el DM prepara [27].

2.1.2. Personajes

Non-Player Characters: Los NPCs son criaturas (monstruos) creadas por los DM, que representan a los habitantes del mundo, y con los que los jugadores interactúan. Estos NPCs poseen un bloque con sus estadísticas conocido popularmente como “Statblock”, el cual tiene la información que los DM necesitan para representar de manera correcta al personaje en cuestión. En la Figura 2.1 se muestra la representación del “Statblock” de un “Ettin” [26], un NPC que representa a un gigante de dos cabezas. Este contiene información de la armadura (AC) del personaje, capacidades, habilidades, y acciones, entre otras.

ETTIN						
<i>Large giant, chaotic evil</i>						
Armor Class 12 (natural armor)						
Hit Points 85 (10d10 + 30)						
Speed 40 ft.						
STR	DEX	CON	INT	WIS	CHA	
21 (+5)	8 (-1)	17 (+3)	6 (-2)	10 (+0)	8 (-1)	
Skills Perception +4						
Senses darkvision 60 ft., passive Perception 14						
Languages Giant, Orc						
Challenge 4 (1,100 XP)						
Two Heads. The ettin has advantage on Wisdom (Perception) checks and on saving throws against being blinded, charmed, deafened, frightened, stunned, and knocked unconscious.						
Wakeful. When one of the ettin's heads is asleep, its other head is awake.						
ACTIONS						
Multiattack. The ettin makes two attacks: one with its battleaxe and one with its morningstar.						
Battleaxe. <i>Melee Weapon Attack:</i> +7 to hit, reach 5 ft., one target. <i>Hit:</i> 14 (2d8 + 5) slashing damage.						
Morningstar. <i>Melee Weapon Attack:</i> +7 to hit, reach 5 ft., one target. <i>Hit:</i> 14 (2d8 + 5) piercing damage.						

Figura 2.1: *Statblock* de un “Ettin”. Fuente: Monster Manual.

Player Characters: Los PCs son personajes creados por los jugadores para representarlos en el mundo del juego. Estos poseen distintas cualidades dependiendo de su raza o clase, además de poseer una personalidad, apariencia e historia de trasfondo, que cada jugador se encarga de crear y luego actuar en la mesa de juego.

2.1.3. Combate

Un combate es un enfrentamiento típicamente entre dos bandos. Por un lado, los monstruos o NPCs, manejados por el DM, y por el otro un equipo de jugadores manejando a sus PCs. El Dungeon Master determina la “iniciativa” de cada participante, la cual define el orden en que actúa cada uno. Ésta se calcula a través de lanzamiento de dados, sumado a una estadística llamada “modificador de destreza”. Los combates se organizan en un ciclo de turnos en el que cada participante realiza los siguientes procedimientos:

1. **Seleccionar la acción:** Los personajes poseen una lista de acciones que pueden utilizar. El jugador al que se le asigna el turno actual en ese momento puede decidir si utilizar alguna acción que tenga en sus estadísticas. En el caso de los NPCs el encargado de elegir es el DM, y en los PCs es el jugador.
2. **Determinar el o los objetivos:** Luego de que se seleccionó la acción, el jugador actual decide a quién o a quiénes le aplicará los efectos especificados en la descripción de las acciones basado en su rango.
3. **Resolver el daño causado o efecto aplicado hacia el personaje objetivo:** Dependiendo de la acción, el objetivo, y el resultado del dado, el DM calcula como el personaje objetivo es afectado y actualiza sus estadísticas. Por ejemplo, un NPC recibe daño de 10 y su indicador de vida (HP) es 15 entonces se resuelve que su nuevo HP es de 5.
4. **Repetir:** Los procedimientos 1, 2 y 3 se repiten por cada uno de los personajes presentes en el combate en un bucle que termina cuando el DM considera que el combate finalizó.

2.2. Manuales de juego de Dungeons and Dragons

La última versión de D&D, 5ª edición, fue lanzada en 2014. Ésta incluye tres libros de reglas centrales llamados Player’s Handbook (PH), Dungeon Master’s Guide (DMG) y Monster Manual (MM). Estos tres libros son necesarios para jugar, aunque los jugadores sólo necesitan el PH, el Dungeon Master el DMG y el MM.

2.2.1. Player's Handbook

“Player's Handbook” o “Manual del Jugador” es un libro que tiene lo necesario para que los jugadores de D&D puedan crear sus personajes [13], además de explicar el proceso del juego. El libro está dividido en tres partes:

- La primera parte trata sobre las consideraciones, reglas y orientación que se debe seguir para crear un PC. También habla sobre la personalización que pueden tener los personajes tales como razas, clases, antecedentes, entre otras [27].
- La segunda parte explica las reglas que deben tener los personajes para llevar a cabo los combates. Incluyendo las tiradas de dados que determinan el éxito o fracaso para las tareas dentro del juego [27].
- La tercera parte detalla todas las reglas de lanzamientos de hechizos, la variedad, y los distintos efectos que estos tienen [27].

2.2.2. Dungeon Master's Guide

“Dungeon Master's Guide” son guías que usa el Dungeon Master para crear mundos y sus ciudades, criaturas NPCs, encuentros, entre otros [11]. El libro está organizado en tres partes:

- **Master of worlds:** Esta parte introduce al lector el mundo de D&D. Además, invita a los Dungeon Masters y jugadores a definir qué tipo de campañas desean llevar a cabo ya sea en sus propios universos, o a través de aventuras publicadas por parte de los desarrolladores del juego [25].
- **Master of adventures:** Esta parte es una guía para que el DM cree las aventuras que serán realizadas a lo largo de la campaña, incluyendo encuentros, tesoros, criaturas, etc. También ayuda a generar historias coherentes y convincentes, además de los personajes NPCs adecuados para cada ambiente desarrollado [25].
- **Master of rules:** En esta parte se describen las reglas necesarias para llevar a cabo las partidas de D&D. También se establece la necesidad de que exista un participante (DM) que sea imparcial al realizar los encuentros, ya que se deben aplicar reglas de daño entre los participantes, orden en que actúan, uso de los dados, uso de habilidades, entre otros [25].

2.2.3. Monster Manual

“Monster Manual” es un libro que contiene criaturas predefinidas en el mundo de D&D y que sirve como guía para que los DM tengan referencias e inspiración al crear sus NPCs. Además, contiene una gran cantidad de statblocks que poseen una reseña de la criatura y de su comportamiento. También posee una guía rápida para la construcción de personajes, explicando sus estadísticas [26].

2.3. Propuestas similares

En la actualidad existen distintas alternativas que permiten la creación de personajes y manejo y seguimiento de combates. Estas propuestas tienen sus pros y contras en comparación con la solución del problema planteado. Estas son algunas de las más populares:

1. **RpgTinker**: Es un sitio web que permite generar NPCs dados ciertos valores preestablecidos. El DM escoge una plantilla de entre las disponibles, y luego ajusta los valores de las capacidades, puntos de vida, y habilidades. Además, permite exportar los NPC resultantes en formato PDF.

- Pros:

- Crear NPCs de manera sencilla.
- Existen plantillas para la crear NPCs rápidamente.
- Acceso gratis para los usuarios.

- Contras:

- La creación de NPCs no se realiza en tiempo real (se necesita el uso de botones para generar los cambios).
- No permite crear PCs ni encuentros.

2. **OrcPub versión 2**: Es un sitio web que permite la generación de personajes, encuentros y contenido relacionado con D&D 5e.

- Pros:

- Permite buscar contenido relacionado con el juego como hechizos, armas.
- Acceso gratis para los usuarios.

- **Contras:**
 - La creación de NPCs no se realiza en tiempo real (se necesita el uso de botones para generar los cambios).
3. **Kobold Fight Club:** Permite manejar encuentros en los que se pueden agregar NPCs definidos en el Monster Manual y creados por los usuarios mediante hojas de cálculo de Google. Además permite agregar una lista PCs indicando algunas características básicas de estos.
- **Pros:**
 - Rapidez al momento de generar encuentros.
 - Algunos de los personajes creados por su comunidad están disponibles para todos los usuarios.
 - **Contras:**
 - Los NPCs y PCs creados muestran solo las estadísticas básicas de estos.
 - La creación de personajes depende del uso de herramientas externas.
 - El manejo de los encuentros no se realiza de manera interactiva y solo tiene la vista de Dungeon Master.
4. **Fantasy Grounds:** Aplicación multiplataforma que permite crear personajes y manejar encuentros de distintos juegos de rol (incluido D&D 5e). Esta tiene una gran cantidad de módulos que son cargados para obtener distintas misiones, personajes, mapas, entre otros. Estos pueden ser adquiridos por los jugadores pagándolos. También, los DM pueden crear salas virtuales de juego en donde puede comunicarse e interactuar con los demás jugadores a través de internet.
- **Pros:**
 - Capacidad de escalamiento a través de módulos actualizables.
 - Se encuentra licenciado oficialmente por Wizard of the Coast.
 - Creación de salas virtuales a las que los jugadores se conectan a través de internet.
 - Herramientas para realizar combates tales como chat y dados virtuales.
 - **Contras:**

- Los módulos tienen un alto costo económico.
 - Se necesita de suscripción mensual o un desembolso significativo para obtener la versión completa (*Premium*).
5. **Roll20:** Es una aplicación web que permite crear campañas, personajes, mapas, entre otros. Además, posee aplicaciones para multiplataforma, posibilidad de utilizar datos virtuales, creación de hojas de personajes, integración con chat de voz y vídeos de Google, entre otros.
- Pros:
 - Gran cantidad de herramientas relacionadas D&D.
 - Gran comunidad de usuarios y colaboración entre estos.
 - Contras:
 - Se necesita pagar para obtener funciones avanzadas.

2.4. Metodología de desarrollo de software

En esta sección se muestra la metodología de desarrollo de software utilizada en el desarrollo de la aplicación.

2.4.1. Extreme Programming XP

La metodología Extreme Programming (XP) posee un conjunto de reglas y prácticas agrupadas en planificación, manejo, diseño, desarrollo y pruebas. XP cuenta con 5 valores que ayudan a que los proyectos sean exitosos [17]:

- Simplicidad: En pequeños y simples pasos se llega al objetivo y se mitigan fallas a medida que ocurran.
- Comunicación: Diálogos frontales, cara a cara, entre desarrolladores y clientes, todos formando un equipo.
- Retroalimentación: Los comentarios del cliente son permanentes y se consideran cambios durante el desarrollo.

- Respeto: Todos los miembros del equipo aportan valor. Los desarrolladores respetan la experiencia del cliente y viceversa.
- Coraje: Si se encuentran errores en cualquier aspecto del proyecto, se debe tener el coraje para enfrentar su solución y adaptarse a los cambios cuando ocurran.

2.5. Patrones de diseño

En esta sección se presenta el patrón de arquitectura utilizado en el diseño de la aplicación web.

2.5.1. Modelo Vista Controlador

El patrón de arquitectura *Model-View-Controller* o Modelo-Vista-Controlador (MVC) separa la implementación en tres capas. Mediante objetos de distintas clases en estas, tienen operaciones relacionadas con el dominio de la aplicación (modelo), la visualización (vista), y la interacción entre el usuario con el modelo y las vistas (controlador) [20].

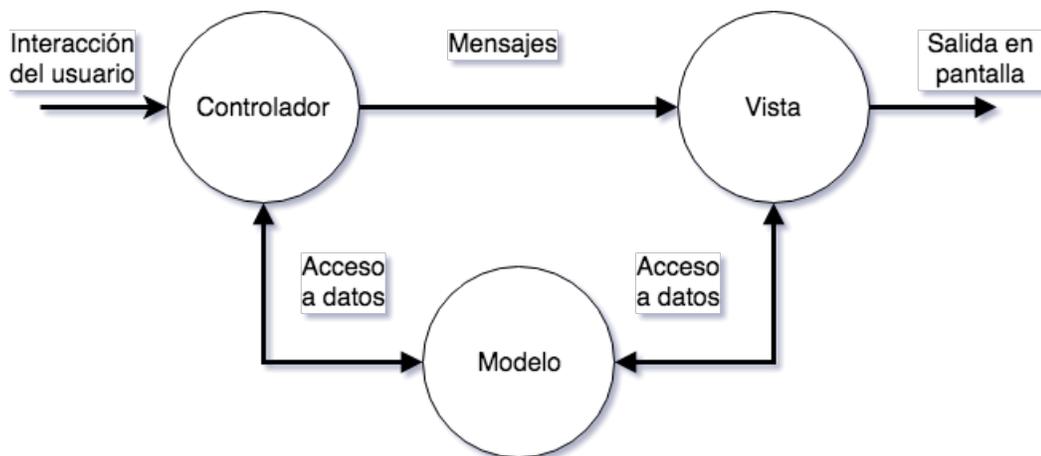


Figura 2.2: Interacción en patrón: Modelo-Vista-Controlador.

En la Figura 2.2 se observa la interacción cíclica entre los componentes. En este el controlador notifica al modelo para que realice sus operaciones. Luego, este avisa a al controlador y a la vista en caso de que haya ocurrido algún cambio de estado. Ahora, las vistas pueden actualizar el modelo mostrado. Además, los controladores pueden cam-

biar la manera en que interactúan con el usuario dependiendo de si es que el modelo ha cambiado su estado [20].

2.6. Tecnologías y frameworks utilizados

En esta sección se describen las tecnologías y frameworks utilizados durante el desarrollo de la aplicación web.

2.6.1. .NET Framework

.NET Framework es un entorno de ejecución administrado para Windows que proporciona servicios a las aplicaciones en ejecución [5] [19]. Tiene dos componentes principales:

- **Common Language Runtime (CLR):** Este administra la memoria, ejecución de los subprocesos, ejecución de código, compilación y otros servicios del sistema [9].
- **Biblioteca de clases:** Biblioteca de código reutilizable que se integran con el CLR. Esta puede ser utilizada por los desarrolladores de aplicaciones. Mediante esta se pueden realizar diversas tareas de programación, incluyendo código para distintos tipos de aplicaciones y servicios, tales como, aplicaciones de consola y de Windows, servicios, y ASP.NET para aplicaciones web [9].

2.6.2. ASP.NET

ASP.NET es un framework web para crear sitios y aplicaciones web con HTML, CSS y Javascript que ofrece tres entornos dependiendo de las necesidades de cada desarrollo [3] [19]. Estas son:

- **Web Forms:** Permite crear sitios web dinámicos arrastrando y colocando componentes que pueden ser controlados por eventos.
- **Web Pages:** Mediante la sintaxis *Razor* permite combinar código HTML y de servidor para crear contenido web dinámico.
- **ASP.NET MVC:** Es una plantilla de proyecto de ASP.NET que permite llevar a cabo desarrollo de aplicaciones web siguiendo el patrón Modelo Vista Controlador (MVC) [3] [2].

2.6.3. Razor

Razor es una sintaxis basada en C# que se utiliza en el desarrollo web con ASP.NET. Permite agregar código de servidor en una vista, siendo su extensión *.CSHTML* [7] [19]. Algunas de sus principales características son:

- **Agregar código a una página HTML:** Mediante el carácter “@” puede comenzar una expresión en línea, un bloque de instrucciones o un bloque de múltiples instrucciones [7], como se observa en el siguiente fragmento de código.

```
1 @{var x = 0;}
2 <p>x es igual a @x</p>
3
4 @{
5     var nombre = "Juan";
6     var dia = Datetime.Now.ToString();
7 }
8 <span>Hola @nombre, hoy es @dia </span>
```

- **Mostrar propiedades y ejecutar métodos de objetos:** Se pueden obtener y mostrar las características accesibles de un objeto [7]. Como se observa en el siguiente fragmento de código, el objeto Request muestra sus propiedades mediante el carácter “@”.

```
1 <h1>Request: @Request.Name</h1>
2 <p>@Request.Url</p>
3 <p>@Request.Date</p>
4 <p>@Request.IP()</p>
```

- **Tomar decisiones:** Una de las características de las webs dinámicas es que pueden definir que mostrar a partir de condicionales (if, else, else if, switch) [7]. Como se observa en el siguiente fragmento de código mediante la sentencia if y else se puede determinar que se va a mostrar.

```
1 @if(mensaje.Valido){
2     <p>El mensaje es: @mensaje.texto</p>
3 }
4 else{
5     <p style="color:red;">El mensaje es invalido</p>
6 }
```

- Bucles de código: Los datos pueden requerir mostrar cierta cantidad de veces o mostrar alguna colección. Para ello se puede utilizar las sentencias de ciclos (for, foreach, while) [7]. En el siguiente fragmento de código se muestra el uso de for each sobre una colección.

```
1 <ul>
2     @foreach (var item in Mensajes) {
3         <li>item.valor</li>
4     }
5 </ul>
```

2.6.4. Materialize

Materialize es un framework front-end para diseñar interfaces web modernas, responsivas y basadas en Material Design de Google. Su objetivo es proporcionar una experiencia unificada en todas las plataformas [1]. Las principales ventajas que ofrece este framework son:

- Fácil integración al proyecto: Se necesita solamente agregar al código de la aplicación los archivos CSS y Javascript. Opcionalmente se utiliza la fuente de Material Design (Roboto).
- Uso de cuadrículas (Grid) responsivas: Utiliza el estándar de grilla responsiva de 12 columnas, que ayuda a mantener el orden entre los elementos que son mostrados.
- Uso de componentes: Posee varios componentes tales como formularios, barras de navegación, botones, “modals”, entre otros.
- Animaciones: Los componentes tienen distintas animaciones que dan una sensación de movimiento, como por ejemplo al hacer foco en botones, su profundidad cambia.

2.6.5. JQuery

Jquery es una biblioteca de JavaScript que manipula el Document Object Model (DOM) en documentos HTML. Realiza manejo de eventos, animaciones y utiliza AJAX para comunicarse con la capa Back-End de las aplicaciones web [18].

2.6.6. SQL Server

SQL Server es un conjunto de herramientas para administrar bases de datos relacionales para el procesamiento de transacciones o datos analíticos en línea [15]. En el motor de base de datos se pueden crear tablas para almacenar datos como textos, fechas, números, etc. de manera segura y administrable [14].

2.6.7. Entity Framework

Entity Framework es un conjunto de tecnologías pertenecientes a ADO.NET del .NET Framework que permite a los desarrolladores de aplicaciones trabajar con datos contenidos en bases de datos en forma de objetos y propiedades. También permite usar funciones para manejar los datos como si fueran colecciones nativas de las aplicaciones en .NET mediante LINQ. Además, genera de mejor manera las conexiones hacia la capa de datos [4].

2.7. Pruebas

En esta sección se explica las técnicas que se utilizan para probar el correcto funcionamiento de la aplicación web.

2.7.1. Prueba de caja negra

Las pruebas de caja negra están enfocadas en probar el funcionamiento respecto a los requisitos funcionales del software. Mediante un conjunto de entradas se pueden probar por completo uno o varios requerimientos del sistema. Su objetivo principal es encontrar errores de funcionamiento, interfaz, en estructuras de datos o acceso a bases de datos, de comportamiento, y de inicialización y terminación [24].

Existen distintos métodos de prueba de caja negra, el utilizado para probar la aplicación web desarrollada se llama “partición de equivalencia”. Este método divide las entradas de los casos de prueba en casos ideales y de error mediante “clases de equivalencia”. Estas clases son un conjunto de entrada de datos de prueba válidos o inválidos [24]. Para su creación se pueden definir de acuerdo a los siguientes lineamientos [24]:

1. *“Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos inválidas.”*

2. “Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y dos inválidas.”
3. “Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una inválida.”
4. “Si una condición de entrada es booleana, se define una clase válida y una inválida.”

2.7.2. Prueba de usabilidad USE

“Usefulness, Satisfaction, and Ease of use” (USE) es un cuestionario que ayuda a medir la usabilidad centrado en tres conceptos, que son utilidad, satisfacción, y facilidad de uso. Sus desarrolladores observaron que existía una fuerte correlación entre la utilidad y la facilidad de uso. Ambos conceptos impulsan la satisfacción del usuario, que a la vez genera que los usuarios utilicen los sistemas (actual o a futuro). Además, la facilidad de uso se puede separar en dos factores fuertemente relacionados, que son facilidad de aprendizaje, y facilidad de uso [23].

Para evaluar el software de esta memoria se utiliza el cuestionario USE adaptado. Esta adaptación está construida en una escala tipo *Likert*¹ de siete puntos, que va desde 1 punto (totalmente en desacuerdo) hasta 7 (totalmente de acuerdo). Se utilizan 30 afirmaciones que los usuarios responden luego de haber interactuado con la aplicación web. Para responder deben marcar una casilla por pregunta. Cabe mencionar que si alguno de los usuarios que completan el cuestionario no puede responder una de las preguntas, este debe marcar la opción del centro (opción 4). En la Figura 2.3 se observa un ejemplo de las afirmaciones que los usuarios evalúan.

Se lo recomendaría a un amigo. *

	1	2	3	4	5	6	7	
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo						

Figura 2.3: Afirmación número 25 del formulario USE Adaptado.

¹La escala de “Likert” sirve para medir actitudes y conocer el grado de conformidad de los encuestados con afirmaciones propuestas [22].

Los resultados se calculan obteniendo un promedio de los puntos obtenidos en cada una de las preguntas. Además, se agrupan los resultados de cada pregunta en la dimensión a la que pertenecen (utilidad, satisfacción, facilidad de uso, facilidad de aprendizaje).

3. Metodología de desarrollo

En el presente capítulo se explica por qué se elige la metodología aplicada al desarrollo de la aplicación web. También se comenta de la necesidad de realizar documentación formal para entender el modelo de negocio. Finalmente, se detallan las iteraciones planificadas para llevar a cabo el desarrollo del software.

3.1. Metodología

La metodología de desarrollo de software utilizada es Extreme Programming (XP). Esta permite realizar un análisis y captura de requisitos durante el desarrollo. Además, cuenta con cortas iteraciones o ciclos con entregas funcionales al final de cada uno de estos. Estas características se ajustan a las necesidades de una aplicación web que necesita un desarrollo ágil, un constante flujo de retroalimentación, un equipo de trabajo reducido, y un enfoque principal en el código por sobre la documentación.

Durante el comienzo del desarrollo de la aplicación web se hizo necesario entender el funcionamiento del negocio de ésta. Por ello se generó documentación que no necesariamente es utilizada en la metodología de desarrollo elegida, pero que sirve como guía para comenzar a entender todos los aspectos relacionados a D&D.

Incluyendo la etapa previa mencionada, se planificaron varias iteraciones de duración variable. Estas principalmente están enfocadas en el desarrollo, recepción de retroalimentación y pruebas unitarias. Finalmente, se agrega la realización de pruebas de usabilidad.

3.2. Planificación

Para definir la planificación se comienza recopilando las “historias de usuario” (presentadas en el capítulo 4). Luego se estima la cantidad de tiempo que necesita cada una

de las iteraciones. En este caso la estimación del tiempo se realiza basado en las experiencias del autor. Cada historia de usuario contiene una lista de tareas específicas que por lo general están repartidas entre una a tres iteraciones.

3.2.1. Etapa previa

En esta etapa se realiza investigación acerca de cómo funciona el juego para tener una perspectiva completa de que es lo que se quiere hacer. Para ello se planifica revisar documentación formal [21] con el objetivo de conocer y analizar las distintas variables involucradas en la creación de personajes NPCs y PCs. Además de cómo estas interactúan en los encuentros. Se destaca esta parte de la planificación ya que en general la metodología utilizada no requiere de documentación formal.

3.2.2. Iteraciones

El desarrollo del software se divide en iteraciones que tienen una duración entre 7 y 20 días. Cada una a su vez está dividida en tareas que se presentan a continuación:

- **Iteración 0: Análisis de los manuales de juego** (15 días): Se planifica realizar una investigación a los manuales del juego con el fin de identificar las principales de características de D&D.

Tareas:

- Análisis Monster Manual.
- Análisis Dungeon Master's Guide
- Análisis Player's Handbook.
- Análisis de herramientas similares.

- **Iteración 1: Documentación de apoyo** (20 días): Se planifica realizar el modelado formal de los componentes para comprender la capa de negocio de la aplicación. Además, generar documentación para ser usada como guía en el desarrollo.

Tareas:

- Generar documentación de apoyo.

- **Iteración 2: Desarrollo de módulo NPC (parte 1)** (15 días): Se planifica desarrollar un módulo que permita mantener (crear, seleccionar, actualizar, eliminar) personajes NPC.

Tareas:

- Capturar requisitos.
- Definir diseño módulo NPC.
- Definir pruebas unitarias.
- Desarrollar prototipo funcional.

- **Iteración 3: Desarrollo de módulo NPC (parte 2)** (15 días): Se planifica terminar el desarrollo del prototipo funcional. Además, comenzar a desarrollar el módulo a partir de las actividades de anteriores.

Tareas:

- Desarrollar prototipo funcional.
- Desarrollar el módulo NPC.

- **Iteración 4: Desarrollo de módulo NPC (parte 3)** (7 días): Se planifica terminar el desarrollo del módulo. Luego, llevar a cabo pruebas unitarias al módulo con el fin de asegurar que el modelo de NPC sea válido según los lineamientos de los manuales del juego.

Tareas:

- Desarrollar el módulo NPC.
- Realizar pruebas unitarias.

- **Iteración 5: Desarrollo de módulo PC (parte 1)** (14 días): Se planifica desarrollar un módulo que permita mantener (crear, seleccionar, actualizar, eliminar) personajes PC.

Tareas:

- Capturar requisitos.
- Definir diseño módulo PC.

- Definir pruebas unitarias.
- Desarrollar prototipo funcional.

- **Iteración 6: Desarrollo de módulo PC (parte 2)** (14 días): Se planifica terminar el desarrollo del prototipo funcional. Además, comenzar a desarrollar el módulo a partir de las actividades de anteriores.

Tareas:

- Desarrollar prototipo.
- Desarrollar el modulo PC.

- **Iteración 7: Desarrollo de módulo Encuentro (parte 1)** (14 días): Se planifica desarrollar un módulo que permita a DMs crear encuentros y a los demás jugadores participar de estos. Además, el desarrollo de un módulo que permita a los usuarios mantener (crear, seleccionar, actualizar y eliminar) personajes NPCs y PCs.

Tareas:

- Capturar requisitos.
- Definir diseño módulo Encuentro.
- Definir pruebas unitarias.
- Desarrollar prototipo funcional.
- Desarrollar el módulo encuentros.
- Desarrollar módulo de usuarios.

- **Iteración 8: Desarrollo de módulo Encuentro (parte 2)** (28 días): Se planifica terminar de desarrollar los módulos de encuentros y usuarios. Finalmente, ajustar los módulos para que estos se integren completamente.

Tareas:

- Desarrollar el módulo encuentros.
- Desarrollar módulo de usuarios.
- Realizar pruebas de integración.
- Realizar pruebas unitarias.

- **Iteración 9: Pruebas de usabilidad** (16 días): Se planifica buscar y analizar métodos de evaluación de usabilidad que entreguen información relevante para concluir acerca de la experiencia al utilizar la aplicación. Luego, realizar estas pruebas con usuarios que utilicen la aplicación. Finalmente, analizar los resultados de las pruebas.

Tareas:

- Buscar y analizar métricas de usabilidad.
- Realizar pruebas de usabilidad.
- Analizar y concluir acerca de los resultados de las pruebas.

4. Requisitos

En el presente capítulo se detallan en primer lugar los roles existentes en D&D que son usuarios de la aplicación web. Luego, los requisitos representados como historias de usuario. Finalmente, algunos casos de uso que ayudan a la comprensión del negocio.

4.1. Roles de jugadores

Los usuarios que interactúan con la aplicación pueden tener ambos roles. También, pueden crear personajes NPCs y PCs, y participar de encuentros. Sin embargo, estos pueden tener distintos roles al momento de usar sus personajes creados, ya sea NPCs o PCs. Por ello, como se observa en el Cuadro 4.1, los usuarios pueden usar estos roles según lo estimen conveniente.

Cuadro 4.1: Rol y descripción de los participantes en D&D 5e.

Rol	Descripción
Dungeon Master	Puede crear NPCs y PCs. También, generar encuentros en los que agrega jugadores, personajes y define el orden de los participantes, para luego administrar los resultados de sus acciones durante este.
Jugador	Puede crear su personaje (PC) y participar de encuentros en el cual verá solo las estadísticas de su personaje.

4.2. Historias de usuario

A partir de entrevistas con el cliente se obtuvieron historias de usuario. A continuación, se detalla su descripción con un código asignado:

- **HU001:** Como Dungeon Master, quiero crear y visualizar NPCs de acuerdo a las guías establecidas por los manuales del juego D&D 5e para ser utilizados en los encuentros.
- **HU002:** Como Dungeon Master, quiero crear y visualizar PCs de acuerdo a las guías establecidas por los manuales del juego D&D 5e para ser utilizados en los encuentros.
- **HU003:** Como Dungeon Master y Jugador, quiero que al momento de crear NPCs los cambios en sus estadísticas se reflejen en tiempo real para que este proceso se realice de forma interactiva.
- **HU004:** Como Dungeon Master y Jugador, quiero que al momento de crear PCs los cambios en sus estadísticas se reflejen en tiempo real para que este proceso se realice de forma interactiva.
- **HU005:** Como Dungeon Master, quiero generar encuentros en los que se agregarán NPCs y PCs y se definir el orden en el cual realizarán acciones para ayudar a manejar la generación de encuentros.
- **HU006:** Como Dungeon Master, quiero ejecutar los encuentros generados en los que los NPCs y PCs interactúan mediante acciones en el orden determinado anteriormente para ayudar a los jugadores a llevar el valor de sus estadísticas.
- **HU007:** Como Jugador, quiero visualizar las estadísticas de mi personaje durante la realización de los encuentros para tener una idea de cuál es el estado de mí PC en el encuentro.
- **HU008:** Como Dungeon Master y Jugador, quiero entrar a la aplicación mediante algún navegador web para interactuar con esta.
- **HU009:** Como Dungeon Master y Jugador, quiero poder acceder a la aplicación desde una *tablet* en posición paisaje para interactuar con esta.
- **HU010:** Como Dungeon Master y Jugador, quiero que el idioma de la aplicación sea inglés para que sea coherente con las reglas del juego.

4.3. Casos de uso

Como parte de la “etapa previa” mencionada anteriormente, se generan casos de uso gráficos para tener una idea global de como será la aplicación web. Cada uno los casos de uso es presentado en esta sección mediante una figura y una descripción de esta.

4.3.1. Caso de uso nivel cero

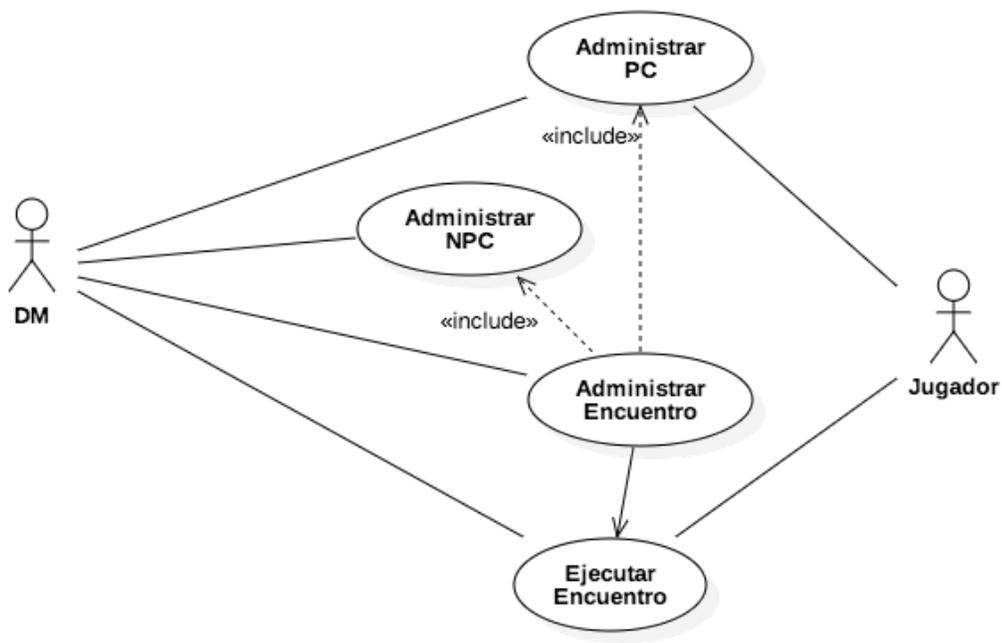


Figura 4.1: Diagrama de caso de uso: Interacción de los usuarios.

En la Figura 4.1 se pueden observar las acciones que pueden realizar los usuarios en la aplicación web. De este se puede destacar que para administrar los encuentros es necesario que existan personajes. Además, los DM pueden administrar los encuentros y los jugadores solo pueden ejecutar encuentros.

4.3.2. Administrar NPC

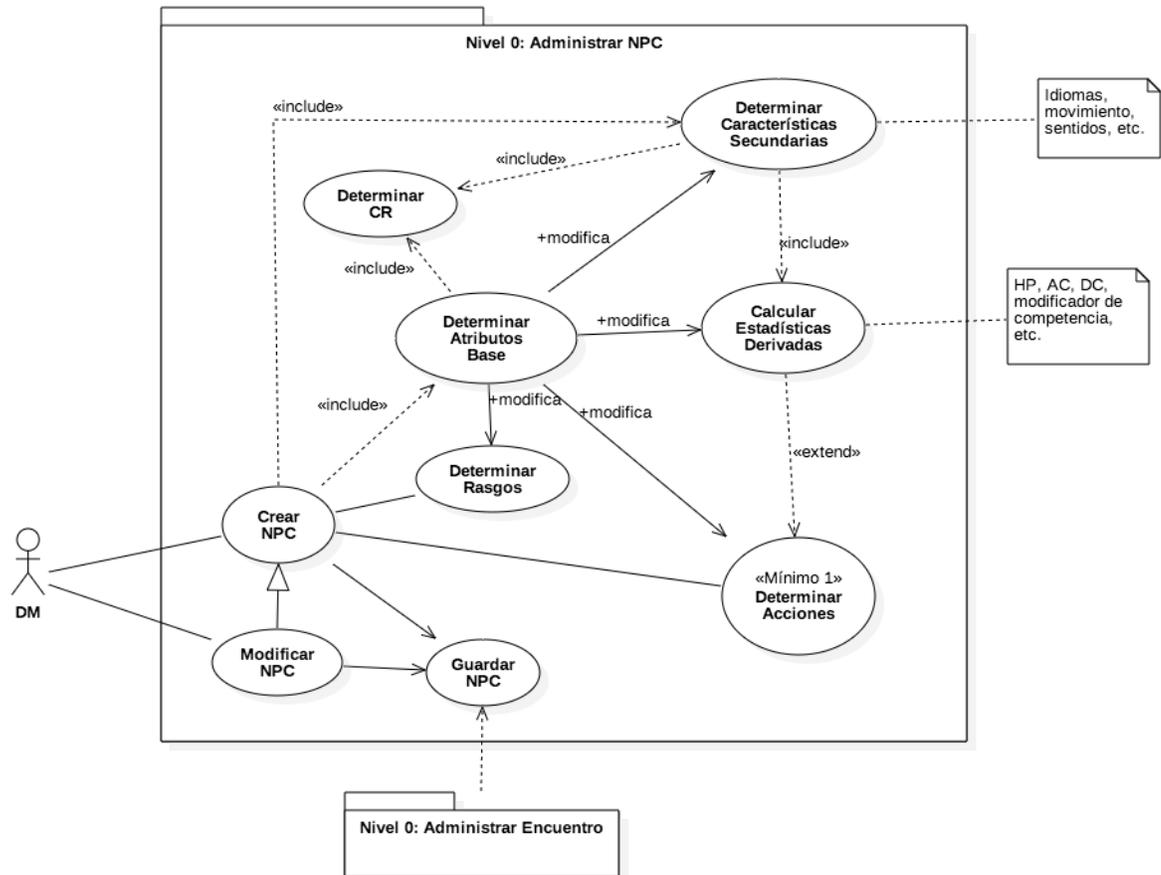


Figura 4.2: Diagrama de caso de uso: Administrar NPC.

En la Figura 4.3 se observa cómo los DM administran NPCs. Además, en el proceso de creación se determinan atributos base, determinan acciones, calculan distintas estadísticas, entre otros.

4.3.3. Administrar PC

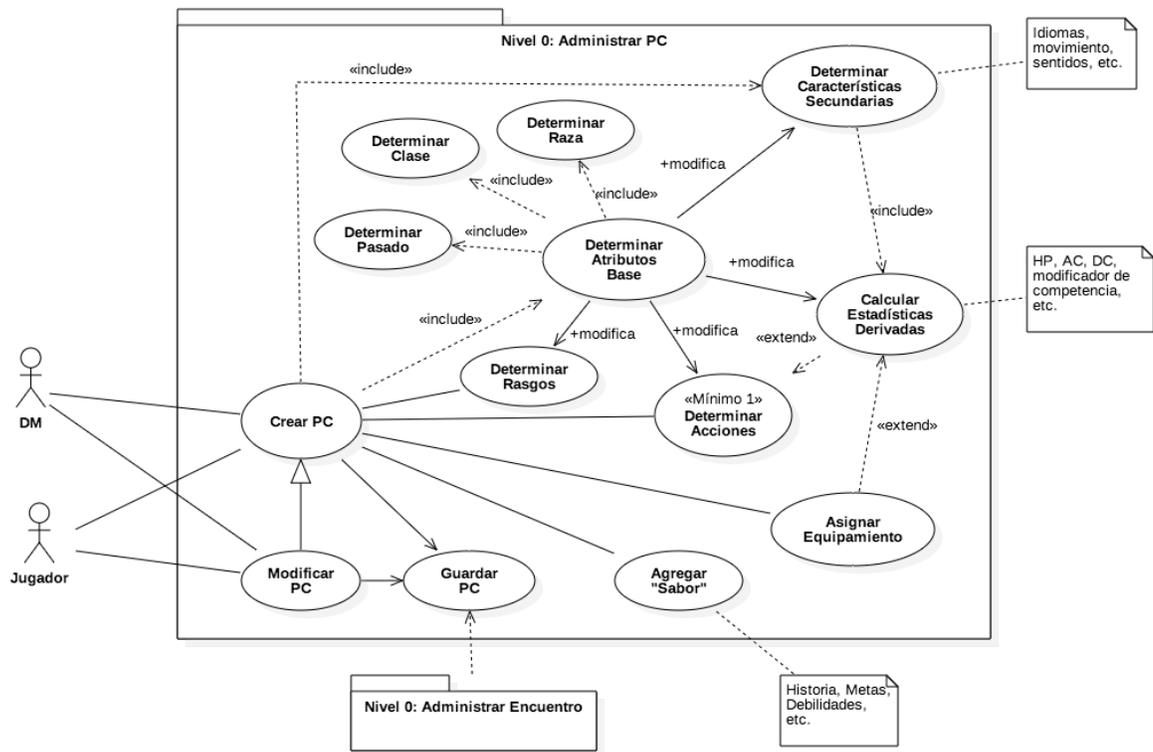


Figura 4.3: Diagrama de caso de uso: Administrar PC.

En la Figura 4.4 se observa cómo los Jugadores y DMs pueden administrar personajes PCs. A diferencia de la creación de NPCs, para los PCs se determinan algunos aspectos distintos como clase, raza y pasado.

4.3.4. Administrar encuentro

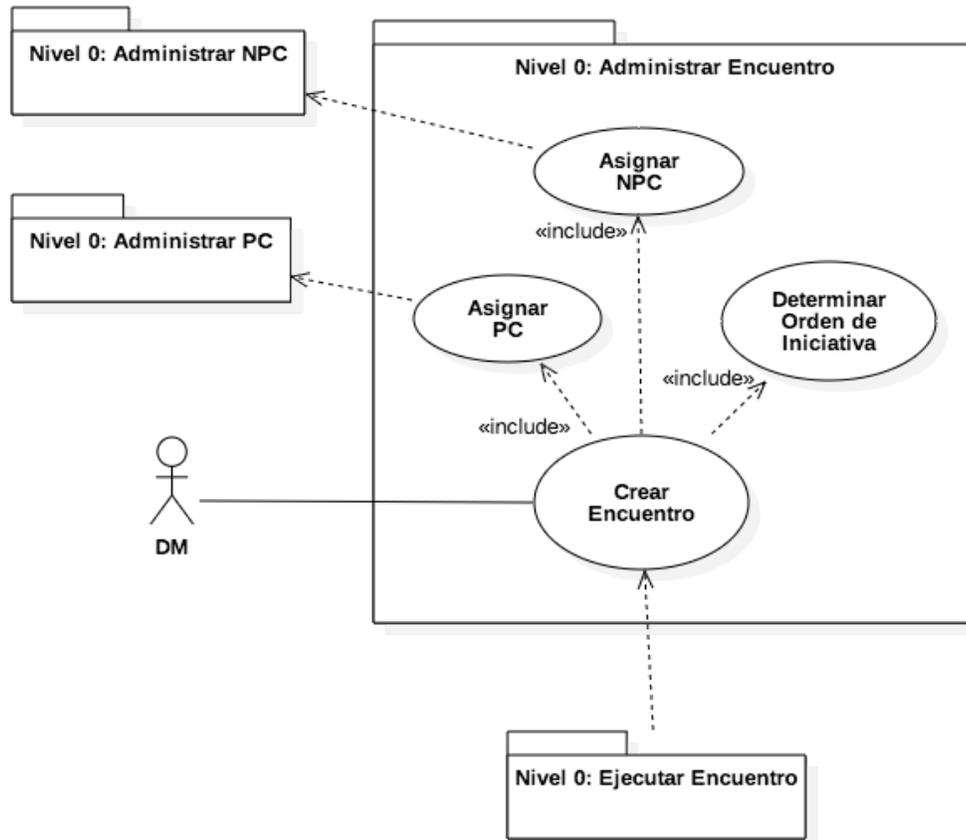


Figura 4.4: Diagrama de caso de uso: Administrar encuentro.

En la Figura 4.4 se observa cómo los DM pueden crear encuentros (combates). En estos es necesario asignar NPCs y PCs, y luego determinar el orden en que participarán (turnos).

4.3.5. Ejecutar encuentro

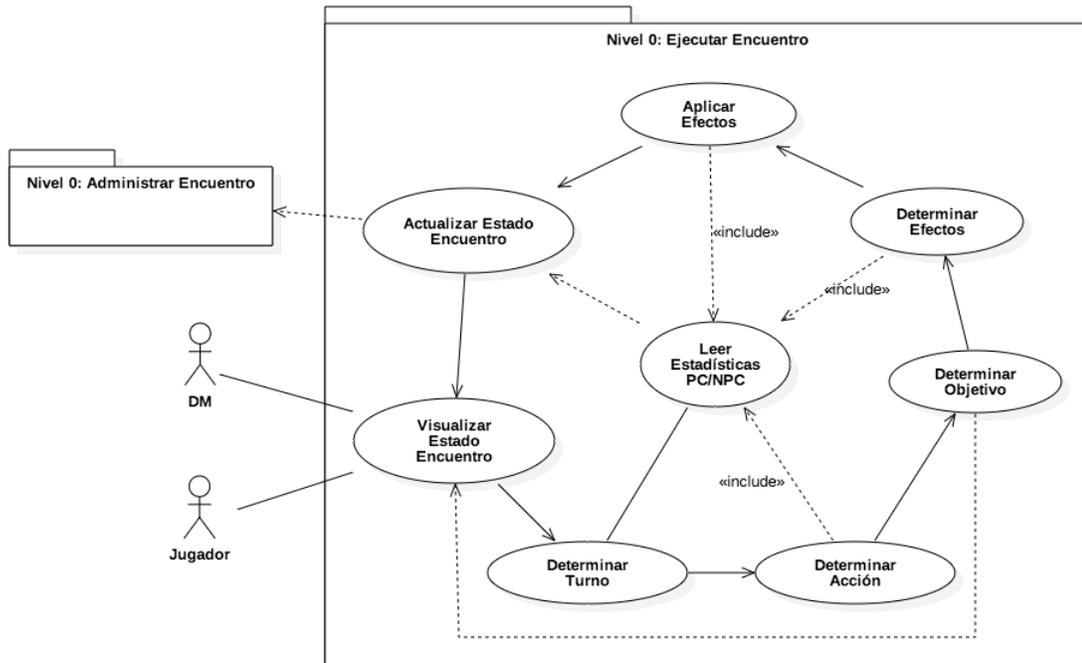


Figura 4.5: Diagrama de caso de uso: Ejecutar encuentro.

En la Figura 4.5 se observa cómo se ejecutan los encuentros de combate en que los Jugadores y DM pueden visualizar el estado de los encuentros. Además, las acciones de estos van ocurriendo de forma cíclica hasta que el DM determina el fin del encuentro.

5. Diseño

En el presente capítulo se describen la arquitectura de la aplicación web mediante la presentación de su ambiente operacional, arquitectura lógica y física, y componentes. Además de cómo fueron diseñados de los módulos, la navegación, las clases y las entidades del modelo relacional.

5.1. Arquitectura de la aplicación

En esta sección se presenta la arquitectura de la aplicación desde una vista general hacia una más específica.

5.1.1. Ambiente operacional

Se describe el ambiente operacional de la aplicación web en términos de sus ambientes lógico y físico:

- **Ambiente lógico**

- Front-end: HTML 5, CSS 3 y JavaScript (ECMAScript 6).
- Back-end: .NET Framework versión 4.5.
- Base de datos: SQL Server 2017.

- **Ambiente físico**

- Servidor Web: Internet Information Services (IIS 10).

La decisión de utilizar esta composición en el ambiente operativo se basa en la experiencia como desarrollador web con estas tecnologías y frameworks. Además, algunos aspectos relevantes que llevan a la utilización de estas son:

- En el Front-end, como se observa en la Figura 5.1, la utilización de la sintaxis “Razor” permite generar vistas dinámicas que mediante la lógica pueden mejorar la experiencia de usuario. A su vez, el manejo del DOM de “jQuery” entrega funcionamiento asíncrono a las interfaces de usuario. Por medio de AJAX que actualiza partes de la vista asíncronamente. También, “Materialize” provee componentes que hacen ver a las aplicaciones modernas e interactivas.
- Por el lado del Back-end, como se observa en la Figura 5.2, “ASP.NET” entrega una amplia colección de librerías para el desarrollo web y se ajusta perfectamente al modelo de capas utilizado con “ASP.NET MVC”. Además, “Entity Framework” facilita la interacción entre la capa de modelos y los datos. Mediante sus librerías simplifica las consultas hacia la base de datos, llevándolas a una sintaxis de más alto nivel. Finalmente, para la capa de base de datos “SQL Server” tiene una gran integración con “.NET Framework” y “Entity Framework”.

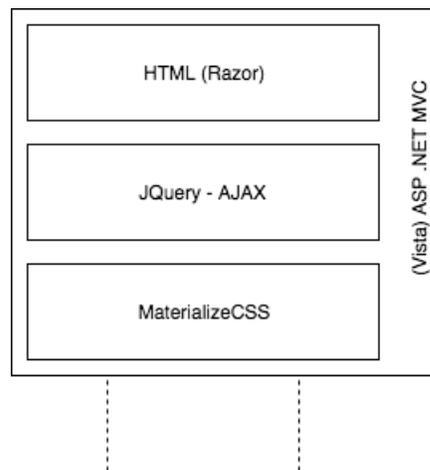


Figura 5.1: Diagrama de front-end.

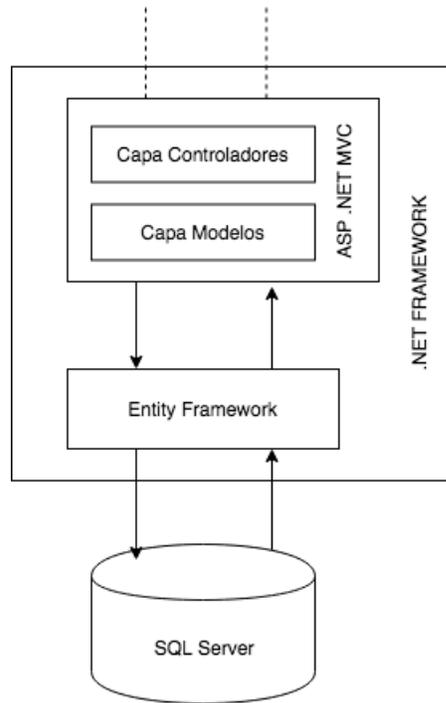


Figura 5.2: Diagrama de back-end.

5.1.2. Arquitectura física

La arquitectura física es cliente-servidor. Como se observa en la Figura 5.3 existe un solo servidor en donde se encuentra la aplicación web y la base de datos. Por otro lado, clientes que pueden ser usuarios de escritorio y/o móviles (Tablet), ambas capas conectadas a través de internet.

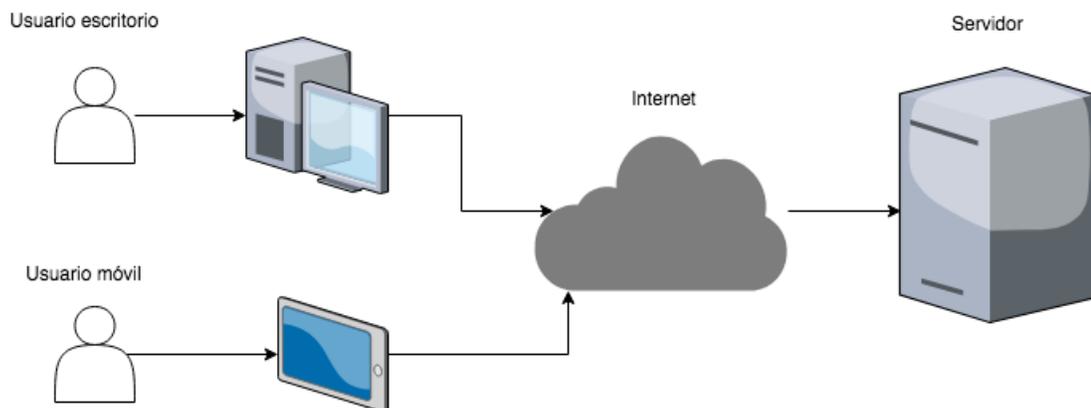


Figura 5.3: Diagrama de arquitectura física.

Se definieron los tipos de clientes o usuarios (escritorio y móvil) debido a las historias de usuario HU009 y HU010. El servidor de aplicación y base de datos están en una misma máquina para que exista un mejor rendimiento entre estos.

5.1.3. Arquitectura lógica

En la arquitectura lógica se utilizará arquitectura de tres capas Modelo Vista Controlador (MVC), como se observa en la Figura 5.4, la cual muestra un diagrama simplificado de esta arquitectura aplicada al desarrollo web de esta memoria. Cabe mencionar que se excluyen algunos modelos que agregan y componen a los principales módulos (NPC, PC y Encuentro) para no sobrecargar la figura presentada.

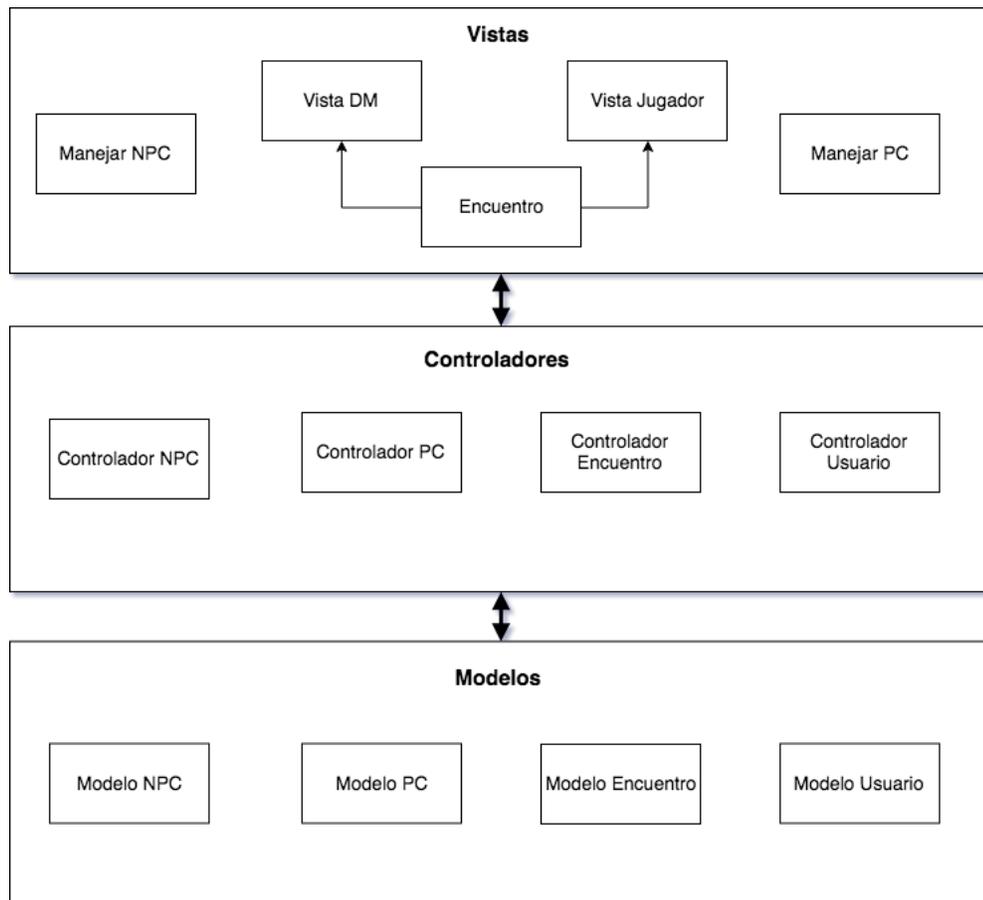


Figura 5.4: Diagrama de arquitectura lógica (MVC).

5.1.4. Componentes

En la Figura 5.5 se observa los componentes de la aplicación. Se aprecia a gran escala como los componentes interactúan.

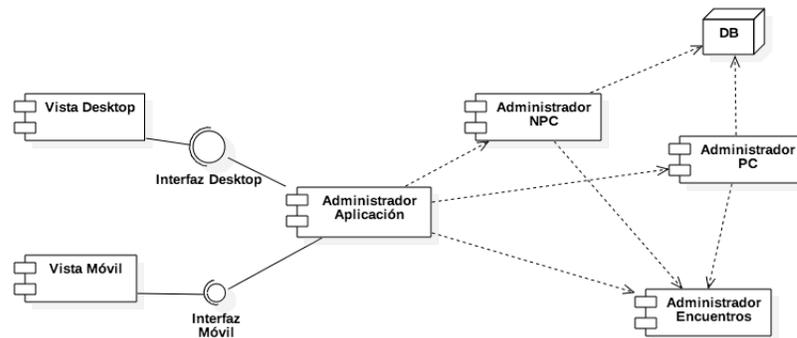


Figura 5.5: Diagrama de componentes.

5.2. Diseño de la aplicación

Para comenzar a describir el diseño de la aplicación, en la Figura 5.6 se observa el flujo de esta. En este diagrama se observan tres procesos principales administrar NPC, PC y encuentros. Cada uno de estos representa uno o más módulos que se describen a lo largo de esta sección.

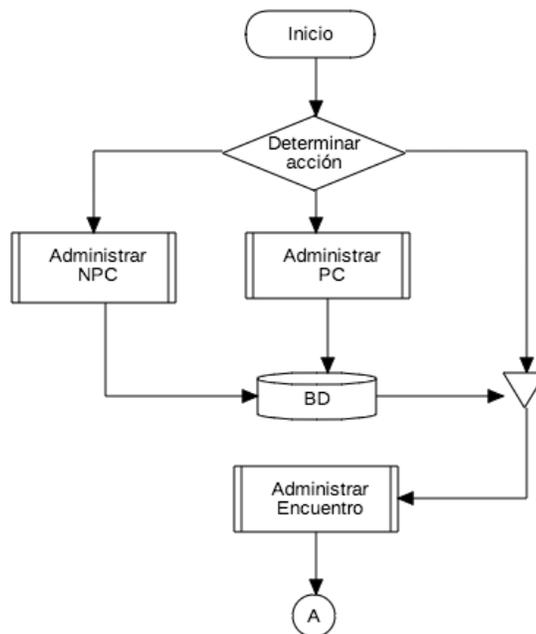


Figura 5.6: Diagrama de flujo (parte I).

5.2.1. Administrar aplicación

Este módulo está encargado de facilitar a los usuarios la realización de las acciones necesarias para acceder y navegar en la aplicación. Las acciones que abarca este módulo son:

- **Registro de usuario:** Para registrarse un usuario debe crear un nombre de usuario y contraseña, y asociar su correo electrónico. Tanto el nombre de usuario como el correo en la aplicación deben ser únicos.
- **Validación de usuario:** Para validar un usuario se debe generar un “token” que será una combinación de cinco caracteres alfanuméricos. Este se envía por correo para validar la autenticidad del correo.
- **Ingreso de usuario:** Para ingresar un usuario debe llenar un formulario de inicio de sesión, donde ingresa su nombre de usuario y contraseña.
- **Ver resumen:** Para ver un resumen un usuario va al inicio de la aplicación.

- **Navegación entre las distintas acciones:** Para navegar en la aplicación un usuario debe ir seleccionando los diferentes menús de la aplicación.

En las acciones descritas anteriormente se hace necesario definir un modelo de usuarios. En la Figura 5.7 se observa la clase “User” que representa este. Cabe mencionar de este que un usuario puede tener una lista de NPCs y PCs asociados al él.

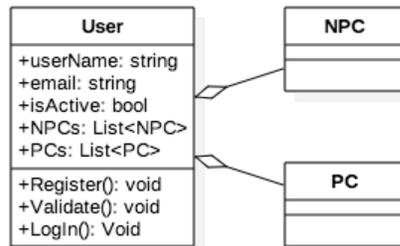


Figura 5.7: Representación clase User.

5.2.2. Administrar NPC

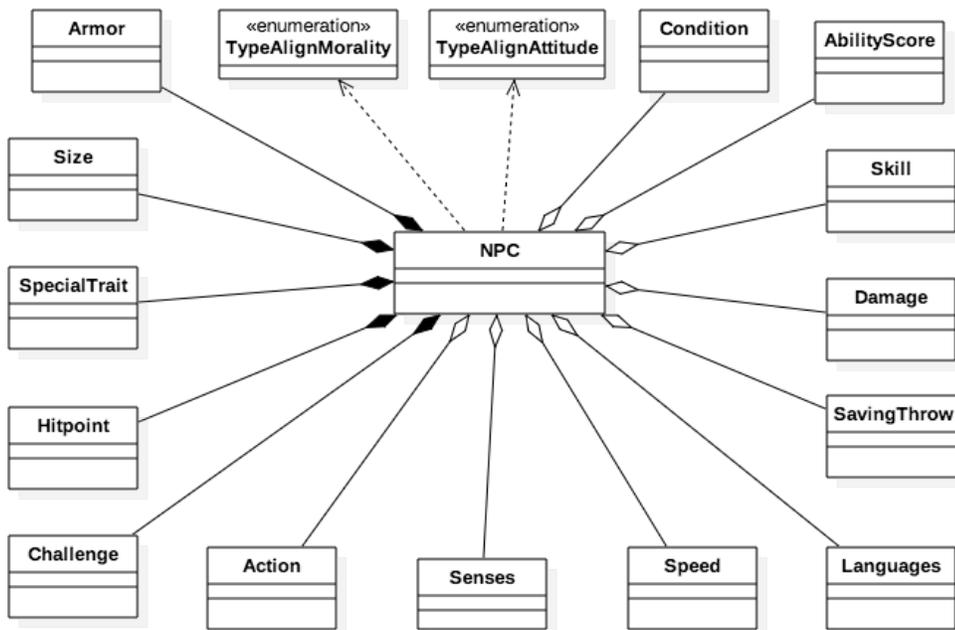


Figura 5.8: Representación de diagrama de clases reducido.

Este módulo es el encargado de todas las acciones que se pueden realizar con los NPCs. Estas acciones son crear, seleccionar, modificar y eliminar. Para entender la creación y en general como son representados estos personajes en la aplicación se presenta en la Figura 5.8 un diagrama de clases comprimido (una versión completa de este se encuentra en el Anexo A.1). Cada una de las clases y la utilidad que esta representa para los personajes son descritas a continuación.

- **NPC:** Esta clase está compuesta y agregada por varias clases. Todas estas representan una característica del personaje. Por ejemplo, los puntos de saludo (HP) están representados por una clase (Hit Point) que compone NPC.

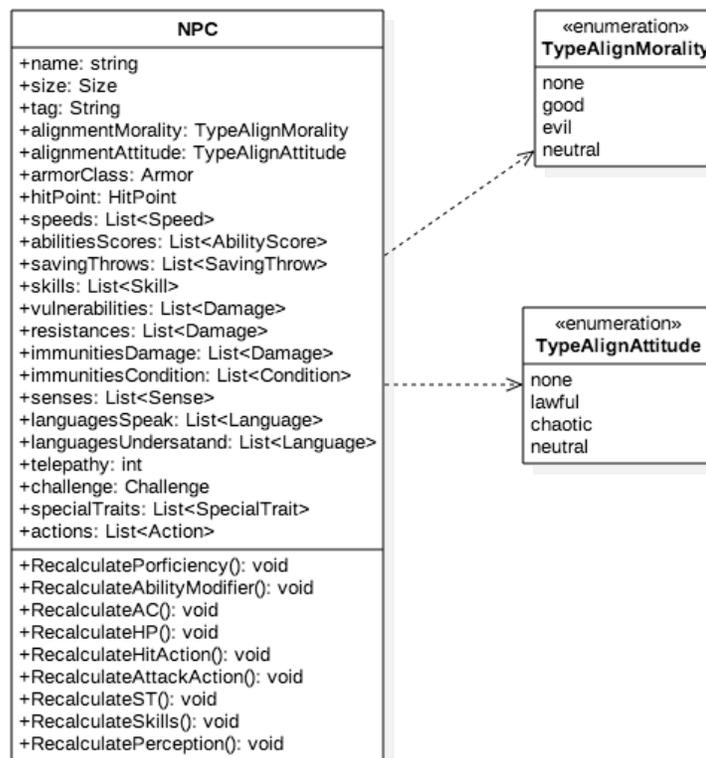


Figura 5.9: Extracto diagrama de clases: Clase NPC.

En la Figura 5.9 vemos la representación solo de la clase NPC en donde se pueden ver todas las características que posee esta. Cada una de las propiedades de la clase NPC equivale a algún aspecto del personaje. Algunas se pueden representar solo con un texto, un número o una enumeración. Estas son:

- *Name*: Representa como es llamada la criatura dentro del juego (su identificador).
- *Tag*: Las etiquetas le entregan características adicionales al personaje además de su *type* (tipo). Por ejemplo, muerto viviente, planta, etc.
- *Alignments*: Los alineamientos (*Alignment attitude y morality*) de un monstruo entrega pistas sobre su comportamiento. Por ejemplo, podrían representar a una criatura mala y caótica (*chaotic,evil*) o que sigue las reglas de la sociedad pero no le importa la moral (*lawful, neutral*).
- *Telepathy*: Indica que una criatura puede comunicarse con otra mediante telepatía, dentro de un rango especificado.

Cabe mencionar que los principales métodos que posee la clase son para recalcular estadísticas dependientes que cambian al actualizar otras. Por ejemplo, al cambiar el valor de la Destreza (“Dexterity”) del personaje, se modifica el valor de armadura (“Armor”) que tenga éste. Estos cambios se deben realizar siempre que una característica esté relacionada con otra (en el ejemplo anterior Destreza y Armadura). En general, las clases están relacionadas con ésta y otras clases, por lo que si se realizan cambios, éstos se ven reflejados en valores de otras clases.

- **Ability score y Saving throw**: Los NPCs tienen seis características (*ability scores*), que son Fuerza (*Strength*), Destreza (*Dexterity*), Constitución (*Constitution*), Inteligencia (*Intelligence*), Sabiduría (*Wisdom*) y Carisma (*Charisma*). Estas tienen un valor (*value*) que va dentro de un rango entre 1 y 30 (aunque el límite normal es 20 para PCs). Además, cada una de las *ability score* tiene un modificador que se calcula como: $modValue \leftarrow floor((value - 10)/2)$. Floor representa a una función matemática de aproximación de piso. Por ejemplo, si el valor de la destreza (DEX) es igual a 13 entonces:

$$modValue_{DEX} \leftarrow floor((value_{DEX} - 10)/2)$$

$$modValue_{DEX} \leftarrow floor((13 - 10)/2)$$

$$modValue_{DEX} \leftarrow floor(1,5)$$

$$modValue_{DEX} \leftarrow 1$$

Entonces, si se considera el rango de las ability score, el mínimo de los modificadores es -5 y el máximo 10 que se denota como “+10”. En la Figura 5.10 se observa la clase AbilityScore. Las propiedades name, value y modValue de esta fueron explicadas. “Short name” es la abreviación del nombre de las ability scores con las primeras tres letras de la palabra. En el ejemplo anterior Destreza (*Dexterity*) fue llamado “DEX” por esto. Adicionalmente, en la clase hay un booleano que indica si es que existe un “Saving Throw” para esa característica.

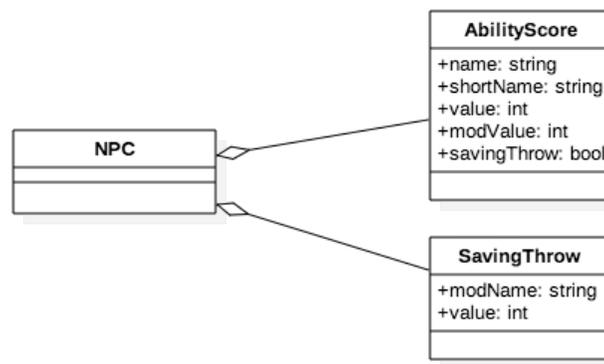


Figura 5.10: Extracto diagrama de clases: Clases AbilityScore y SavingThrow.

La clase “SavingThrow” o “Tirada de salvación” contiene el nombre corto de una *ability score* por ejemplo “CON” para *Constitution* y un valor (*value*). Este valor indica en el juego que algunos monstruos pueden resistir ciertos efectos de ataques e influye directamente en su HP. Para calcularlo se debe sumar el valor del modificador de la característica con su *proficiency* ó competencia (valor que es explicado en el ítem *Challenge*). Por ejemplo, si una criatura tiene +3 en el modificador “CON” y su *proficiency* es de +2 entonces la tirada de salvación de esta es +5.

- **Armor:** Un monstruo puede portar armaduras y/o escudo, además de tener una resistencia innata que representa la capacidad de esquivar, bloquear y desviar ataques. Cada una de estas entregan un valor al total de armadura (*total*).

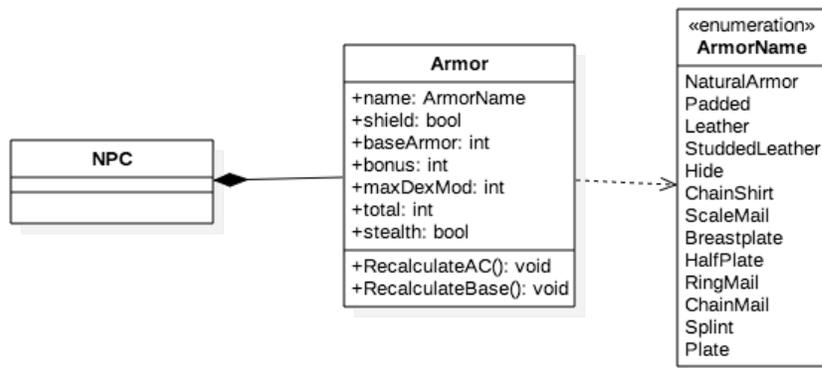


Figura 5.11: Extracto diagrama de clases: Clase Armor.

En la Figura 5.11 se observa la enumeración “ArmorName” que contiene todos los nombres de las armaduras del juego. Cada una de estas posee reglas que permiten calcular el valor de total de la armadura. Para ello se utiliza la tabla mostrada en la Figura 5.12. Como se puede observar se utiliza el modificador de Destreza para el cálculo del total. Además, si este usa o no escudo influye en el valor.

Armor	Cost	Armor Class (AC)	Strength	Stealth	Weight
<i>Light Armor</i>					
Padded	5 gp	11 + Dex modifier	—	Disadvantage	8 lb.
Leather	10 gp	11 + Dex modifier	—	—	10 lb.
Studded leather	45 gp	12 + Dex modifier	—	—	13 lb.
<i>Medium Armor</i>					
Hide	10 gp	12 + Dex modifier (max 2)	—	—	12 lb.
Chain shirt	50 gp	13 + Dex modifier (max 2)	—	—	20 lb.
Scale mail	50 gp	14 + Dex modifier (max 2)	—	Disadvantage	45 lb.
Breastplate	400 gp	14 + Dex modifier (max 2)	—	—	20 lb.
Half plate	750 gp	15 + Dex modifier (max 2)	—	Disadvantage	40 lb.
<i>Heavy Armor</i>					
Ring mail	30 gp	14	—	Disadvantage	40 lb.
Chain mail	75 gp	16	Str 13	Disadvantage	55 lb.
Splint	200 gp	17	Str 15	Disadvantage	60 lb.
Plate	1,500 gp	18	Str 15	Disadvantage	65 lb.
<i>Shield</i>					
Shield	10 gp	+2	—	—	6 lb.

Figura 5.12: Tabla cálculo de *Armor Class*. Fuente: Player’s Handbook.

- **Size:** En la Figura 5.13 se observa la clase *Size* que representa el tamaño del monstruo. La propiedad *size* define el tipo de tamaño que va desde *Tiny* a *Gargantuan*.

También, “space” define la cantidad de espacio que ocupa la criatura usando la medida pies (*feet, ft.*).

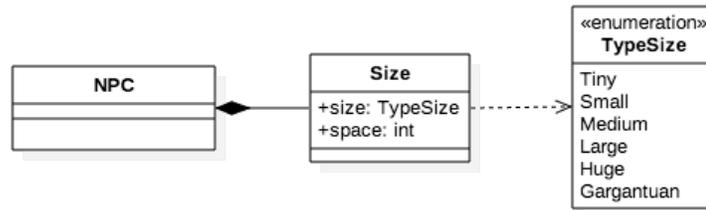


Figura 5.13: Extracto diagrama de clases: Clase Size.

- HitPoint:** Los Hit Points (HP) representan la “vida” de un NPC en términos de cuánto daño puede recibir antes de caer. Durante los encuentros estos van siendo modificados pudiendo llegar a cero o menos lo que indica que este personaje murió o está en proceso de morir.

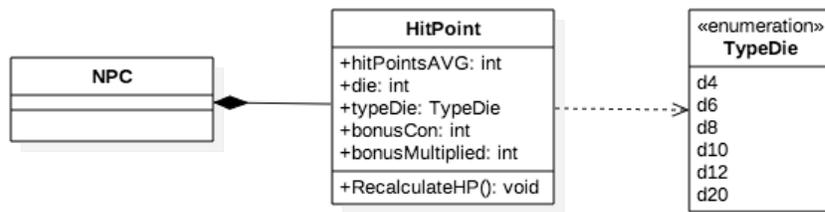


Figura 5.14: Extracto diagrama de clases: Clase HitPoint.

El total de HP de un NPC en la clase HitPoint, como se muestra en la Figura 5.14, es llamado *hitpointsAVG*. Este valor representa como el promedio por dado de HP mostrado en la Figura 5.15. Luego, este valor se multiplica por la cantidad de dados. Por ejemplo, (2d4) representa a 2 multiplicado por d4 (2.5) lo que da como resultado 5. Adicionalmente, a este promedio se le suma el modificador de constitución (*Constitution*) multiplicado por la cantidad de dados. Finalmente, si por ejemplo un monstruo tuviera 2 dados d4 y su valor modificador de constitución es +2, entonces el valor total de los HP es considerado como “2d4 + 4” y su resultado $2 * 2,5 + 2 * 2$ igual a 9.

HIT DICE BY SIZE		
Monster Size	Hit Die	Average HP per Die
Tiny	d4	2½
Small	d6	3½
Medium	d8	4½
Large	d10	5½
Huge	d12	6½
Gargantuan	d20	10½

Figura 5.15: *Hit Dice* por tamaño. Fuente: Monster Manual.

- **Speed:** La velocidad en D&D representa a la distancia (ft.) máxima que este puede moverse en un turno.

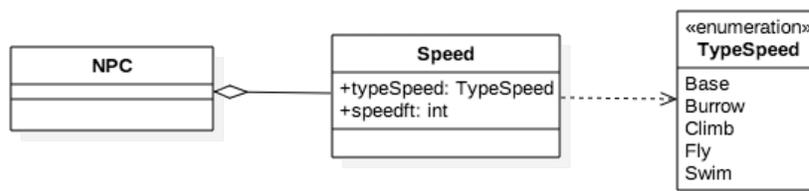


Figura 5.16: Extracto diagrama de clases: Clase Speed.

En la Figura 5.16 se muestran los tipos de velocidades que existen. Un NPC tiene una lista de velocidades. Todos los monstruos poseen capacidad de caminar (*Base*). También podría excavar (*Burrow*), trepar (*Climb*), volar (*Fly*) y nadar (*Swim*). Por ejemplo, la lista de velocidades de un dragón podría ser: “Speed: 30 ft., Fly 50 ft.” Que representa que puede caminar hasta 30 pies o volar hasta 50 pies en su turno.

- **Skill:** En la Figura 5.13 se observa la clase Skill (Habilidad). Cada habilidad en el juego está relacionada con una característica (ability score) en la que el monstruo tiene una gran especialidad.

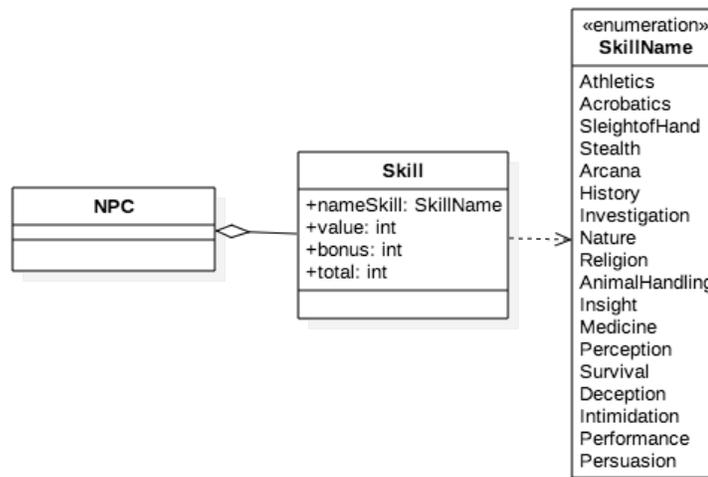


Figura 5.17: Extracto diagrama de clases: Clase Skill.

Para calcular el total del valor de las *skills* se considera el modificador del *ability score* asociado sumado con el *proficiency bonus* del personaje. Esta asociación se puede ver en la Figura 5.13. Adicionalmente, existe una propiedad de la clase llamada “bonus” en la cual se puede agregar un bono si es que el DM quisiera. Por ejemplo, para representar acondicionamiento o experiencia en realizar una tarea.

Strength	Wisdom
Athletics	Animal Handling
Dexterity	Insight
Acrobatics	Medicine
Sleight of Hand	Perception
Stealth	Survival
Intelligence	Charisma
Arcana	Deception
History	Intimidation
Investigation	Performance
Nature	Persuasion
Religion	

Figura 5.18: Relación entre *ability scores* y *skills*. Fuente: Player’s Handbook.

- Damage y Condition:** Algunos NPC pueden tener vulnerabilidades, resistencias e inmunidades a ciertos tipos de daño (damage). Tal y como se observa en la figura 5.19, éstos están relacionados con la clase NPC a través de listas de los tres aspectos. Nótese que un monstruo puede ser inmune a uno o más efectos (condition).

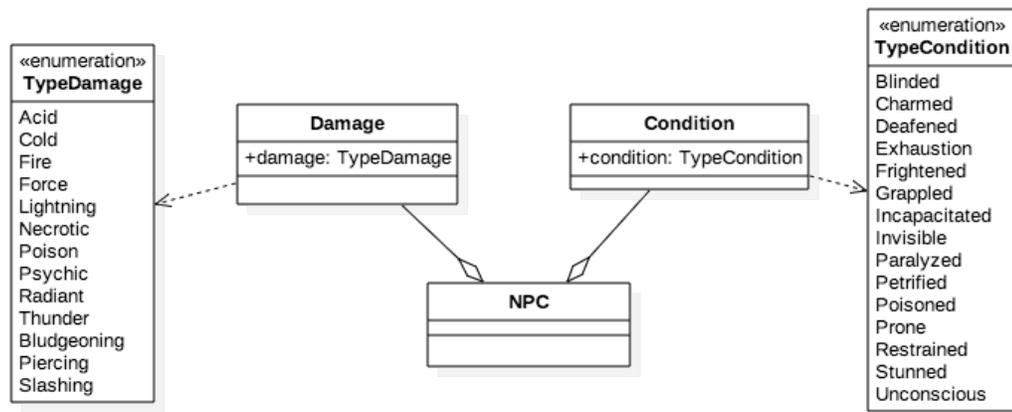


Figura 5.19: Extracto diagrama de clases: Clases Damage y Condition.

- **Sense:** En la Figura 5.13 se observa la clase “Sense” que en el modelo corresponde a una lista de “sentidos” o formas de detectar a otras criaturas. Por defecto todos los NPC tienen “passive Perception” y su valor se calcula como el modificador de sabiduría (wisdom) sumado al proficiencyBonus. Los demás tipos de sentidos son opcionales estos pueden ser los que se ven en la enumeración “TypeSense” con un rango en pies(ft.).

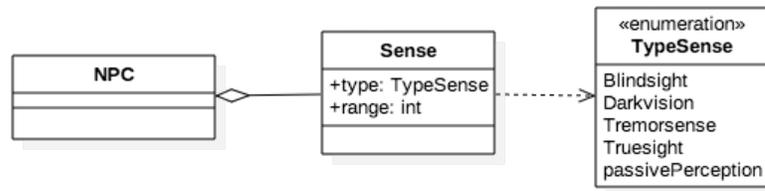


Figura 5.20: Extracto diagrama de clases: Clase Sense.

- **Language:** En la Figura 5.21 se puede observar la clase Language. Un NPC puede tener una lista de idiomas que conoce (esto es, que habla y potencialmente lee y escribe). Pese a esto, un monstruo podría entender un idioma, pero no hablarlo. Por ello, para modelar esto se utilizan dos listas de lenguajes en la clase NPC, una para los idiomas que habla (*languageSpeak*) y otra para los que entiende (*languageUnderstand*). Para definir el nombre de los lenguajes se utiliza la enumeración

LanguageName.

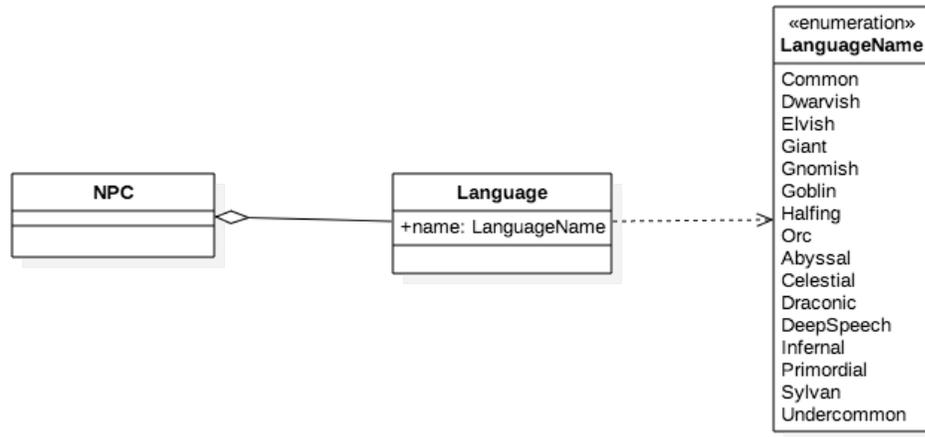


Figura 5.21: Extracto diagrama de clases: Clase Language.

- **Challenge:** En la Figura 5.22 se observa la clase Challenge que representa el “Challenge Rating” (CR) que es un indicador del grado de desafío o peligro de una criatura. Tiene tres propiedades:



Figura 5.22: Extracto diagrama de clases: Clase Challenge.

- La primera propiedad, “value” representa el valor del desafío, es decir, qué tan grande es la amenaza presentada por el NPC.
- La segunda “XP” es la “recompensa” que se obtiene al derrotar a un monstruo, en términos de experiencia. En la Figura 5.23 se observa la correlación entre estas dos propiedades.

Challenge	XP	Challenge	XP
0	0 or 10	14	11,500
1/8	25	15	13,000
1/4	50	16	15,000
1/2	100	17	18,000
1	200	18	20,000
2	450	19	22,000
3	700	20	25,000
4	1,100	21	33,000
5	1,800	22	41,000
6	2,300	23	50,000
7	2,900	24	62,000
8	3,900	25	75,000
9	5,000	26	90,000
10	5,900	27	105,000
11	7,200	28	120,000
12	8,400	29	135,000
13	10,000	30	155,000

Figura 5.23: Puntos de experiencia por *Challenge Rating*. Fuente: Monster Manual.

- La tercera propiedad, “proficiencyBonus”, potencia algunas características del personaje.

Challenge	Proficiency Bonus	Challenge	Proficiency Bonus
0	+2	14	+5
1/8	+2	15	+5
1/4	+2	16	+5
1/2	+2	17	+6
1	+2	18	+6
2	+2	19	+6
3	+2	20	+6
4	+2	21	+7
5	+3	22	+7
6	+3	23	+7
7	+3	24	+7
8	+3	25	+8
9	+4	26	+8
10	+4	27	+8
11	+4	28	+8
12	+4	29	+9
13	+5	30	+9

Figura 5.24: *Proficiency Bonus* por *Challenge Rating*. Fuente: Monster Manual.

- **Special trait:** Los “Special Traits” son características que pueden ser relevantes

durante los encuentros, pero que no necesariamente influyen de manera inmediata sobre el combate. En la Figura 5.25 se encuentra la clase `SpecialTrait` que en el modelo de NPC representa a una lista de estas. En la clase se puede ver que tiene un nombre y una descripción.

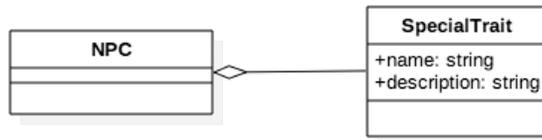


Figura 5.25: Extracto diagrama de clases: Clase `SpecialTrait`.

- Action:** Un NPC tiene una lista de acciones que utilizar en los combates. En el diagrama de estas la clase “`Action`” como se puede ver en la Figura 5.26 tiene varias propiedades. Las que son utilizadas dependiendo del tipo de acción. Por ejemplo, un ataque cuerpo a cuerpo tiene una cantidad de daño, esta limitado a un cierto alcance, con objetivos definidos, el tipo de daño que aplica, entre otros aspectos.

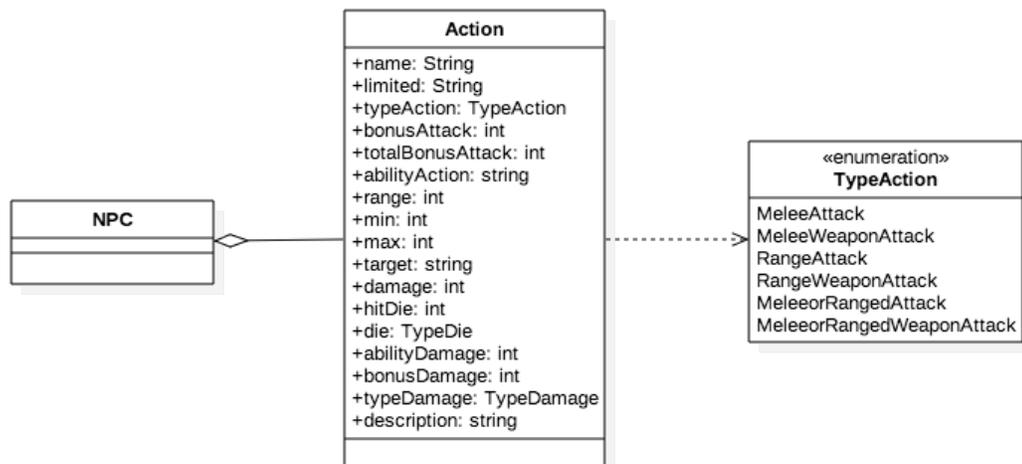


Figura 5.26: Extracto diagrama de clases: Clase `Action`.

5.2.3. Administrar PC

Este módulo es el encargado de administrar los PC. Similar al módulo de NPCs, está encargado de crear, seleccionar, modificar y eliminar. Para definir el diseño en la Figura 5.27 se muestra la clase PC en una versión acotada. Se puede apreciar que varias de las propiedades de NPC se repiten (por lo que no son mostradas en el diagrama), aunque poseen grandes diferencias en el modo en que esta modelado.

En la Figura 5.27 se observan tres enumeraciones que representan aspectos que definen la personalidad del PC. Estas son:

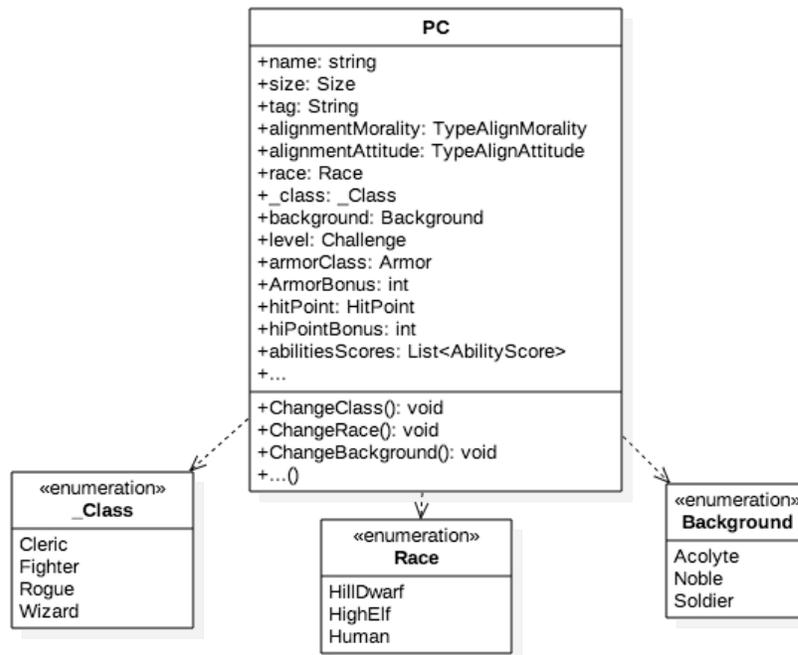


Figura 5.27: Clase PC (versión acotada).

- Race:** En el mundo de D&D existen distintas razas, cada una de las cuales tiene ciertas características que las diferencian unas de otras. Para cada una de las razas modeladas se agregan características o se adquieren bonos especiales. Para ejemplificar los cambios, en el Cuadro 5.1 se observan las diferencias que entre las razas “Human” y “Hill Dwarf”.

Cuadro 5.1: Diferencias entre *race*: “Hill Dwarf” y “Human”.

Race	Hill Dwarf	Human
Ability Score Increase	+2 Constitution, +1 Wisdom	+1 Strength, +1 Dexterity, +1 Constitution, +1 Intelligence, +1 Wisdom, +1 Charisma
Size	Medium	Medium
Senses	Darkvision 60 ft.	-
Speed	25 ft.	25 ft.
Languages	Common, Dwarvish	Common
Resistances	Poison	-
Special Traits	<p>-Dwarven Resilience. <i>You have advantage on saving throws against poison, and you have resistance against poison damage.</i></p> <p>- Hardy. <i>Your speed is not reduced by wearing heavy armor.</i></p> <p>-...</p>	<p>- Additional Language. <i>You can speak, read, and write one extra language of your choice.</i></p>

- **Class:** Las “clases” determinan el “rubro” en que se desempeñan los PCs, y determinan sus especializaciones y sus habilidades, e incluso sus acciones. Por ejemplo, luchador, mago, entre otros. Similar a “Race” este cambia características y bonos. Por ejemplo, al cambiar de “Wizard” a “Cleric”. Las diferencias entre estas se observan en el Cuadro 5.2.

Cuadro 5.2: Diferencias entre *class*: “Wizard” y “Cleric”.

Class	Wizard	Cleric
Hit Dice	1d6 (4)	1d8 (5)
Saving Throws	Intelligence, Wisdom	Wisdom, Charisma
Special Traits	<p>-Proficiencies. <i>Daggers, darts, slings, quarterstaves, light crossbows.</i></p> <p>- Skills. <i>Choose two from Arcana, History, Insight, Investigation, Medicine, and Religion.</i></p>	<p>- Proficiencies. <i>Light armor, medium armor, shields; All simple weapons.</i></p> <p>- Skills. <i>Choose two from History, Insight, Medicine, Persuasion, and Religion.</i></p>
Actions	<p>-Longsword. <i>melee, one creature, 1d8 slashing damage, versatile (1d10).</i></p> <p>-Light Crossbow. <i>ranged, 1d8 piercing damage, one creature, range 80/320, ammunition, loading, two handed.</i></p>	<p>-Mace. <i>melee, one creature, 1d6 bludgeoning damage, reach 5 ft.</i></p> <p>-Light Crossbow. <i>1d8 piercing damage, one creature, range 80/320, ammunition, loading, two handed.</i></p>

- **Background:** Los PCs tienen una historia previa que determina algunas de sus características, representando su vida antes de iniciar su entrenamiento en su clase. Por ejemplo, noble, soldado, entre otros. Al igual que “Race” y “Class” cambia características al personaje como en el caso de “Acolyte” a “Noble” del Cuadro 5.3.

Cuadro 5.3: Diferencias entre *background*: “Acolyte” y “Noble”.

Background	Acolyte	Noble
Skills	Insight, Religion	History, Persuasion
Special Traits	<p>-Languages. <i>Two of your choice. You can speak, read, and write one extra language of your choice.</i></p> <p>- Shelter of the Faithful. <i>As an acolyte, you command the respect of those who share your faith, and you can perform the religious ceremonies of your deity...</i></p>	<p>-Proficiencies. <i>One type of gaming set.</i></p> <p>- Languages. <i>One of your choice.</i></p> <p>Position of Privilege. <i>Thanks to your noble birth, people are inclined to think the best of you. You are welcome in high society, and people assume you have the right to be wherever you are...</i></p>

Cabe mencionar que para aplicar un cambio entre alguno de estos aspectos se utilizan los métodos “ChangeRace”, “ChangeClass” y “ChangeBackground”.

5.2.4. Administrar Encuentro

Este módulo es presentado a través de la continuación del diagrama de flujo de la Figura 5.6. De esta manera se pone por encima el diseño del funcionamiento del módulo sin presentar demasiadas clases como en las explicaciones de módulos pasados. Este módulo está dividido en dos partes principales: generación y ejecución de encuentros.

- **Crear Encuentro:** Para que un Dungeon Master cree un encuentro, primero es necesario tener creado uno o más NPCs, y que existan otros participantes con sus respectivos PCs creados. Si se observa la Figura 5.6 los personajes (NPCs y PCs) deben estar almacenados en la base de datos de la aplicación, y al generar un encuentro se van a usar los que se encuentren ahí.

En primer lugar, para cada uno de los personajes que participen se debe calcular el valor de iniciativa. Este proceso lo realiza el DM mediante lanzamiento de dados. El resultado de este lo ingresa a la aplicación.

Cuadro 5.4: Ejemplo de salida de crear Encuentros.

Personaje	Participante	Iniciativa
NPC1	DM	15
NPC2	DM	14
PC1	Jugador1	13
PC3	Jugador3	11
PC2	Jugador2	10

Finalmente, como se observa en el Cuadro 5.4, para pasar a la siguiente fase (ejecutar encuentro) se tiene una lista de NPCs asociados al DM y PCs asociados a sus respectivos jugadores. En ambos casos se debe tener la iniciativa ingresada o calculada. Cabe destacar que esta lista debe ser ordenada de mayor a menor por el valor de la iniciativa (mayor iniciativa actúa antes).

- **Ejecutar Encuentro:** Teniendo como entrada la lista descrita en el ítem anterior se puede dar comienzo a un encuentro, el cual está organizado en turnos. En cada uno de éstos se sigue la estructura descrita en la Figura 5.28.

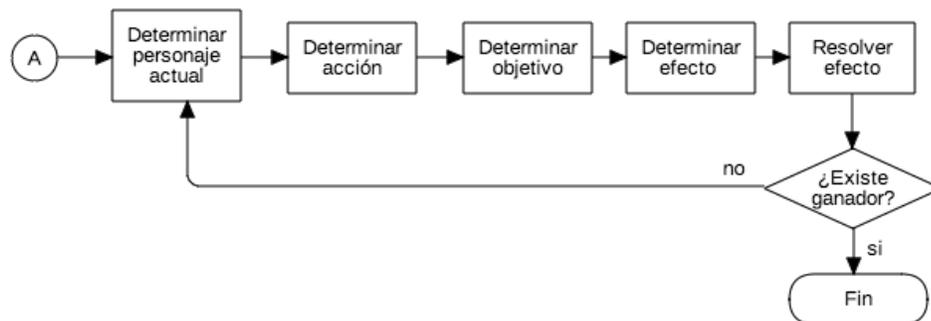


Figura 5.28: Diagrama de flujo (parte II).

Los procesos del flujo de los encuentros son:

- **Determinar personaje actual:** Según el orden establecido por la iniciativa un personaje es asignado al turno actual.

- **Determinar acción:** El participante del personaje actual (DM o Jugador) puede elegir una acción que desea realizar, de acuerdo a las ya establecidas por las reglas.
- **Determinar objetivo:** El participante del personaje actual debe elegir el o los objetivos de la acción seleccionada anteriormente.
- **Determinar efecto:** El DM determina el efecto y lo ingresa como entrada. Este efecto puede ser hacer daño, curar, o agregar un estado. El daño o la curación se considera como una modificación a la cantidad de HP que tenga el objetivo. Aplicar un efecto o “estado” agrega un elemento a una lista de efectos que posee cada personaje.
- **Resolver efecto:** La aplicación calcula el daño o curación modificando el HP del personaje, o agregando o removiendo algún efecto de la lista de efectos (en caso el efecto termine o sea removido por otro efecto).
- **¿Existe ganador?:** El DM puede poner fin al encuentro si es que estima que este terminó. Lo que termina este ciclo de procesos. En caso contrario se vuelve a realizar el proceso con el siguiente personaje en la lista.

Los usuarios que participan de los encuentros desempeñan distintos roles. Por un lado, uno de los usuarios crea una instancia de encuentro, por lo que en esta instancia su rol es Dungeon Master. El DM tienen la posibilidad de manejar el encuentro con una interfaz que le entrega herramientas para avanzar en los turnos, aplicar los efectos de acciones, etc. Por otro lado, el resto de los participantes del encuentro tienen rol de Jugador. Estos jugadores, tienen una interfaz acotada y solo pueden ver la actualización de sus estadísticas.

6. Desarrollo

En el presente capítulo se describe cómo se realiza la implementación de la aplicación web durante las iteraciones planificadas. Cabe mencionar que se utiliza el tiempo gramatical (pasado) distinto al resto del documento para realizar de mejor manera el contraste con la planificación.

6.1. Iteraciones

Cada una de las iteraciones descritas en esta sección explican cómo se llevó a cabo el desarrollo de la aplicación según lo planificado en la sección 3.2 Planificación.

6.1.1. Iteración 0: Análisis de los manuales de juego

En la iteración 0 estaban planificadas tareas de análisis principalmente a los manuales de D&D y herramientas existentes. Cada una de estas fue afrontada de la siguiente forma:

- **Análisis Monster Manual:** Se analizaron principalmente los NPCs que aparecen en este manual. Obteniendo de esto, un modelado de las criaturas (NPC) y un bosquejo del diagrama de clase de esta.
- **Análisis Dungeon Master's Guide:** Se analizó este manual buscando aspectos más específicos que complementan al “Monster Manual”. Mediante esto, se pudieron identificar de mejor forma cómo se construyen los NPCs y cómo estos interactúan en el mundo del juego.
- **Análisis Player's Handbook:** Se analizaron las tablas para realizar cálculos de estadísticas de los personajes que aparecen en este manual. Con ello, se complementó el modelado de la lógica que poseen los personajes.

- **Análisis de herramientas similares:** Se analizaron las herramientas de creación de personajes. Se buscaron las carencias que tienen estas soluciones principalmente en la interactividad, facilidad de uso y experiencia de uso. No se revisaron herramientas de manejo de encuentros, con el fin de no tener una idea influenciada y atender a los requerimientos del cliente.

Se decidió que el análisis de los manuales del juego se realizara como tarea durante todo el desarrollo porque existen bastantes aspectos que no alcanzaron a ser cubiertos en una sola etapa. Además, se replanificó una nueva iteración que al comienzo del desarrollo de la aplicación fue necesaria, incluso antes de capturar las historias de usuario, llamada documentación de apoyo (considerada como iteración 1). Todo esto, con el fin de comprender de mejor manera el negocio y agilizar el comienzo del desarrollo de la aplicación.

6.1.2. Iteración 1: Documentación de apoyo

En esta iteración se planificó generar documentación de apoyo como fue descrito en la sección 3.1. La actividad que se realizó fue:

- **Generar documentación de apoyo:** Se realizó una serie de diagramas de componentes, de flujo y de casos de uso. Con el fin de entender el negocio y para tener una guía en el desarrollo de la aplicación. Se generó este tipo de documentación que no se realiza cuando se utiliza la metodología de desarrollo escogida.

6.1.3. Iteración 2: Desarrollo de módulo NPC (parte 1)

Desde este punto, se comenzó con el desarrollo de software. En esta iteración se comenzó con el desarrollo del módulo de NPCs mediante las tareas que se habían planificado. Estas tareas fueron:

- **Capturar requisitos:** Se capturaron las historias de usuario: **HU001** y **HU003**. Para comenzar con su análisis y desarrollo.
- **Definir diseño módulo NPC:** Usando el análisis realizado en la iteración 0, se definió el diseño del módulo con el objetivo de abarcar la solución de las historias de usuario. Esto sirvió como punto de inicio para comenzar el desarrollo.

- **Definir pruebas unitarias:** Se definieron las pruebas a ejecutar de cada una de las características de los NPCs. Por ejemplo, cálculo de la armadura, de HP, etc. Con esto, se definió la pauta para realizar las pruebas de caja negra.
- **Desarrollar prototipo funcional:** Se comenzó a desarrollar el prototipo funcional que será fundamental para la obtener retroalimentación durante la construcción de la aplicación. Con esto, se obtuvo un “mockup” (maqueta) de la creación de NPC.

Cabe mencionar que durante esta iteración se definió qué tecnologías y frameworks se utilizaron. Además, se planificó el desarrollo del software, debido a que ya se contaba con una mejor idea de lo que se iba a hacer, a diferencia de las iteraciones pasadas.

6.1.4. Iteración 3: Desarrollo de módulo NPC (parte 2)

Esta iteración fue una continuación de la anterior. Esta fue una base para continuar con el desarrollo del módulo NPC. Las tareas que estaban planificadas fueron:

- **Desarrollar prototipo funcional del módulo NPC:** Se continuó con el desarrollo del prototipo mejorándolo mediante la obtención de retroalimentación. Con esto, se pudo obtener un grado de avance que permitió terminar la maquetación funcional del módulo.
- **Desarrollar el módulo NPC:** Se comenzó a desarrollar el módulo de NPC. Se logró un rápido avance dada la base que entrega el prototipo funcional. Sin embargo, se retrasó debido a complicaciones con las tecnologías utilizadas. Se desarrolló una parte del código del módulo de la aplicación.

Esta iteración tomó más tiempo de lo previsto. Esto se debió a retrasos en la implementación de funcionalidades web asíncronas (jQuery, AJAX). Se decidió replanificar y agregar una nueva iteración para cubrir el tiempo de demora que se generó.

6.1.5. Iteración 4: Desarrollo de módulo NPC (parte 3)

En esta iteración se planeó terminar el desarrollo del módulo NPC y ejecutar las pruebas unitarias ya creadas anteriormente para este. Las tareas realizadas fueron:

- **Desarrollar el módulo NPC:** Se terminó el desarrollo del módulo durante esta iteración. Sin embargo, la aplicación solo manejaba la creación, selección, actualización y eliminación de una sola instancia de NPC. Pese a esto, se continuó con la etapa de pruebas con el fin de acelerar el desarrollo.
- **Realizar pruebas unitarias:** Se probaron todas las funcionalidades de creación, edición y eliminación del personaje, con el fin de validar el desarrollo realizado. Con esto, se completaron las pruebas de caja negra.

Como se menciona anteriormente, quedó pendiente el manejo de varias instancias de NPCs en la aplicación. Esto se realizó posteriormente en conjunto con el desarrollo del módulo PC, debido a que se debían realizar tareas similares.

6.1.6. Iteración 5: Desarrollo de módulo PC (parte 1)

Para esta iteración estaba planificado comenzar el desarrollo del módulo PC. Estas fueron:

- **Capturar requisitos:** Se capturaron las historias de usuario: **HU002** y **HU004**.
- **Definir diseño módulo PC:** Se definió el diseño del módulo PC. Este se simplificó dado los análisis realizados anteriormente y porque algunos componentes del módulo NPC son reutilizables. En general, esto agilizó el proceso de desarrollo del módulo, lo que tuvo como resultado un esquema para utilizar en el desarrollo del módulo.
- **Definir pruebas unitarias:** Se definieron las pruebas unitarias que consistían principalmente en cambiar la *race*, *class* y *background* de un PC. Con esto, se obtuvo la pauta para realizar las pruebas de caja negra.
- **Desarrollar prototipo funcional:** Se comenzó a desarrollar el prototipo funcional del módulo. Esto tomó menos tiempo de lo esperado, por lo cual solo quedaron detalles para terminarlo por completo en la siguiente iteración.

En general, se intentó realizar esta iteración lo más rápido posible para terminar el desarrollo del módulo PC.

6.1.7. Iteración 6: Desarrollo de módulo PC (parte 2)

En esta iteración se tenía planificado terminar el desarrollo del módulo PC. Se realizaron las tareas:

- **Desarrollar prototipo:** Se puso fin rápidamente al desarrollo de la maquetación del módulo PC, mediante la base generada en la iteración anterior. Con esto, se obtuvo por completo el prototipo funcional de PC.
- **Desarrollar el módulo PC:** Se desarrolló por completo este módulo. Cabe mencionar, se completó la tarea pendiente del módulo NPC. Como resultado de esto, se dio por finalizado el desarrollo de creación de personajes (NPCs y PCs).
- **Realizar pruebas unitarias:** Se realizaron las respectivas pruebas de caja negra del módulo PC. Lo que permitió validar el desarrollo del módulo.

Luego de terminar estas tareas, los módulos que involucran el manejo de personajes quedaron completados. Este hito fue importante ya que el grado de avance de la aplicación sirvió para satisfacer las demandas de tiempo que se necesitan en el desarrollo de esta memoria.

6.1.8. Iteración 7: Desarrollo de módulo Encuentro (parte 1)

Para las iteraciones 7 y 8 estaba planificado desarrollar el módulo de combates, pero además se hizo necesario definir un módulo que manejara la aplicación y los usuarios. Por lo que se agregó a estas etapas el nuevo módulo que se desarrolló paralelamente. Las tareas que se describen en ambas iteraciones hacen relación al desarrollo de ambos módulos. Estas son:

- **Capturar requisitos:** Se capturaron las historias de usuario: HU005, HU006 y HU007.
- **Definir diseño:** Se definió el diseño del módulo, que al igual que en tareas pasadas de diseñar módulos, en este se utilizó el análisis realizado anteriormente. Se logró obtener una esquemmatización de los módulos de usuario y encuentro.
- **Definir pruebas unitarias:** Se definieron las pruebas de caja negra a realizar, que principalmente son para validar el módulo de usuarios. En cambio, como el módulo

de encuentros abarca la totalidad de las funcionalidades de la aplicación, se decidió realizar además pruebas de integración. De esto se obtuvo una pauta para realizar estas pruebas.

- **Desarrollar prototipo funcional:** Se decidió omitir el prototipo de los módulos, debido a que era necesario agilizar el desarrollo de los módulos. Como resultado de esto, se reemplazó estas tareas por las de desarrollar el módulo de encuentros y usuarios.
- **Desarrollar el módulo encuentros:** Se comenzó el desarrollo de este módulo a la par con el módulo de usuarios. Debido a que están fuertemente relacionados. Como resultado de esto, se alcanzó un grado de avance en cuanto al código del módulo en comparación con el módulo de usuarios.
- **Desarrollar módulo de usuarios:** Se comenzó a desarrollar este módulo, de manera que el mayor grado de avance se centrara en este módulo en comparación con el de usuarios. El motivo de esto fue que para continuar con el desarrollo del módulo encuentros era necesario que existieran usuarios que agregar a los combates. Con esto, se obtuvo un avance casi completo del módulo.

En general, el tiempo que tomó la realización de las tareas de este fue corto. De hecho, se intentó agilizar de gran manera las tareas para comenzar a realizar en el desarrollo lo más rápido posible.

6.1.9. Iteración 8: Desarrollo de módulo Encuentro (parte 2)

Se planificó en esta iteración terminar por completo la aplicación. Para ello el tiempo que se estimó fue mayor al de las demás iteraciones. Se realizaron las tareas:

- **Desarrollar módulo de usuarios:** Se terminó de desarrollar el módulo rápidamente debido al avance de la iteración anterior. Con esto, se completó el desarrollo del código del módulo de usuarios.
- **Desarrollar el módulo encuentros:** Se terminó de desarrollar el módulo de encuentros. Debido a la complejidad de este, se completó en un tiempo más largo de lo esperado. Como resultado de esto, se completó el desarrollo por completo de la aplicación.

- **Realizar pruebas unitarias:** Se realizaron las pruebas de caja negra definidas en la iteración anterior. Como resultado se validaron los aspectos definidos en estas.
- **Realizar pruebas de integración:** Se realizaron las pruebas de integración que permiten validar por completo la aplicación. Como en iteraciones anteriores se había probado cada componente, estas pruebas tuvieron un gran índice de éxito. Como resultado de esto, se dio por cerrado validación del desarrollo de la aplicación.

En esta iteración se decretó el fin del desarrollo de software como tal. Además, debido a que el tiempo que se planificó para realizar esta iteración poseía un margen mayor en caso de que existieran retrasos, se completó en el plazo estimado.

6.1.10. Iteración 9: Pruebas de usabilidad

En esta iteración se realizaron las pruebas de usabilidad a la aplicación. Las tareas que se realizaron fueron:

- **Buscar y analizar métricas de usabilidad:** Se analizaron tipos de formularios para realizar las pruebas de usabilidad, como resultado de esto, se llegó a la decisión de elegir el formulario USE que permite evaluar la usabilidad centrados en la utilidad, satisfacción y facilidad de uso. Además, para que los usuarios ejecutaran las pruebas se generó un manual de usuario que contiene una serie de pasos que permiten probar por completo todos los módulos de la aplicación. Sumado a esto se creó el formulario a través de la plataforma “Google Forms”.
- **Realizar pruebas de usabilidad:** Se realizaron las pruebas de usabilidad a la aplicación, que fue publicada en un servidor. Posterior a esto, los participantes de las pruebas realizaron el cuestionario. Como resultado de esto, se obtuvieron los resultados de los cuestionarios que posteriormente serán utilizados para concluir acerca de estos.
- **Analizar y concluir acerca de los resultados de las pruebas:** Se analizaron los resultados de las pruebas. Para esto, se graficaron los resultados agrupados en las categorías (Utilidad, Satisfacción, Facilidad de uso y facilidad de aprendizaje) pertenecientes a cada una de las preguntas. Con esto, se llegó a concluir acerca de esta actividad.

7. Pruebas

En el presente capítulo se presentan los resultados de las pruebas a la aplicación web. En primer lugar, se muestran los resultados de las pruebas de caja negra organizadas por módulos. Además, se muestran los resultados de la prueba de usabilidad que se realiza mediante el cuestionario USE.

7.1. Pruebas de caja negra

En algunas iteraciones durante el desarrollo de la aplicación se ejecutaron pruebas unitarias. En esta sección se muestran las más relevantes agrupadas por el módulo al que pertenecen. Cada una de las pruebas está representada por un cuadro que contiene la prueba de caja negra de la funcionalidad.

7.1.1. Administrar NPC

Cuadro 7.1: Prueba de caja negra: Propiedad name.

ID	PCN001
Nombre	PCN_ChangeName
Descripción	Cambiar la propiedad name.
Precondición	El texto ingresado no debe estar vacío.
Resultado esperado	El nuevo valor de la propiedad name debe ser el ingresado.
Entrada a probar	<i>name</i> ← “RedDragon”
Resultado	La propiedad ha sido cambiada a <i>name</i> ← “RedDragon”

Cuadro 7.2: Prueba de caja negra: Propiedad size.

ID	PCN002
Nombre	PCN_ChangeSize
Descripción	Cambiar la propiedad size.
Precondición	El nuevo valor debe pertenecer a $L_{size} \leftarrow \{Tiny, Small, Medium, Large, Huge, Gargantuan\}$
Resultado esperado	El nuevo valor de la propiedad size debe ser el ingresado y pertenecer a L_{size} .
Entrada a probar	$size \leftarrow Tiny$
Resultado	La propiedad ha sido cambiada a $size \leftarrow Tiny$

Cuadro 7.3: Prueba de caja negra: Propiedad Armor Class.

ID	PCN003
Nombre	PCN_ChangeAC
Descripción	Calcular el valor de “armor”. Este se obtiene mediante la lista de armaduras que aparece en la Figura 5.12.
Precondición	El valor de “armorName” debe pertenecer a $L_{armorname} \leftarrow \{NaturalArmor, Padded, Leather, StuddedLeather, Hide, ChainShirt, ScaleMail, Breastplate, HalfPlate, RingMail, ChainMail, Splint, Plate\}$, shield debe ser booleano. Adicionalmente, el valor de “ $modifier_{DEX}$ ” debe estar en $-5 < modifier_{DEX} < 10$, en este caso $modifier_{DEX} \leftarrow 1$.
Resultado esperado	El nuevo valor de “total” debe ser 14.
Entrada a probar	$armorName \leftarrow Leather, shield \leftarrow True$
Resultado	$total \leftarrow 14$

Cuadro 7.4: Prueba de caja negra: Propiedad Hit Points.

ID	PCN004
Nombre	PCN_ChangeHP
Descripción	Calcular el valor de “hitpointAVG”. Este se obtiene mediante la fórmula: $hit\ point\ AVG \leftarrow die * typedie + die * modifier_{CON}$.
Precondición	El valor de typedie debe pertenecer a $L_{typeDie} \leftarrow \{d4, d6, d8, d10, d12, d20\}$, $die > 0$, y el valor de “modifierCON” debe estar en $-5 < modifier_{CON} < 10$. En este caso $modifier_{CON} \leftarrow 2$.
Resultado esperado	El nuevo valor de “hitpointAVG” debe ser 18.
Entrada a probar	$die \leftarrow 4, typeDie \leftarrow d4$
Resultado	$hit\ point\ AVG \leftarrow 18$

Cuadro 7.5: Prueba de caja negra: Propiedad Ability Score Modifier.

ID	PCN005
Nombre	PCN_ChangeModifierAbilityScore
Descripción	Calcular el valor de la propiedad modValue de la clase AbilityScore, en donde este parámetro se calcula como: el mayor o igual entero en $((AbilityScore.value - 10)/2)$
Precondición	El valor debe ser $AbilityScore.value \in \mathbb{Z} / 1 \leq AbilityScore.value \leq 30$
Resultado esperado	El cálculo debe ser $modValue \in \mathbb{Z} / -5 \leq modValue \leq 10$
Entrada a probar	$value \leftarrow 12$
Resultado	$modValue \leftarrow 6$

Cuadro 7.6: Prueba de caja negra: Propiedad Saving Throw.

ID	PCN006
Nombre	PCN_AddSavingThrow
Descripción	<p>Agregar a la lista de <i>SavingThrow</i> un elemento. Además, realizar el cálculo de la propiedad “value” de la clase. Este se obtiene mediante $value \leftarrow abilityScoreModifier + proficiency$.</p>
Precondición	<p>El valor de “<i>abilityScoreModifier</i>” debe estar en $-5 \leq modifier_{CON} \leq 10$. En este caso $modifier_{CON} \leftarrow 2$. Además, <i>proficiency</i> debe estar en $2 \leq proficiency \leq 9$. En este caso $proficiency \leftarrow 2$.</p>
Resultado esperado	<p>Se debe agregar a la lista de <i>SavingThrow</i> el tipo de <i>abilityScore</i> y el <i>total</i>. En este caso <i>dexterity</i> + 4.</p>
Entrada a probar	$L_{SavingThrow}.add(dexterity)$
Resultado	Se agregó a $L_{SavingThrow}$ el valor <i>dexterity</i> + 4.

Cuadro 7.7: Prueba de caja negra: Propiedad Skill.

ID	PCN007
Nombre	PCN_AddSkill
Descripción	Agregar a la lista de Skill un elemento. Además, realizar el cálculo de la propiedad “value” de la clase. Este se obtiene mediante $value \leftarrow abilityScoreModifier + proficiency + bonus$.
Precondición	El valor de “ <i>abilityScoreModifier</i> ” debe estar en $-5 \leq modifier_{CON} \leq 10$. En este caso $modifier_{CON} \leftarrow 2$. También, <i>proficiency</i> debe estar en $2 \leq proficiency \leq 9$. En este caso $proficiency \leftarrow 2$. El valor de bonus debe ser $bonus > 0$.
Resultado esperado	Se debe agregar a la lista de Skill el <i>skillName</i> asociado a <i>abilityScore</i> . En este caso $deception + 7$.
Entrada a probar	$L_{Skill}.add(deception)$
Resultado	Se agregó a $L_{SavingThrow}$ el valor $deception + 7$.

Cuadro 7.8: Prueba de caja negra: Propiedad Speed.

ID	PCN008
Nombre	PCN_AddSpeed
Descripción	Agregar a la lista de Speeds un elemento. Para ello, se agrega el tipo y valor de la clase.
Precondición	El nuevo valor debe pertenecer a $L_{speed} \leftarrow \{Base, Burrow, Climb, Fly, Swim\}$
Resultado esperado	Se debe agregar a la lista de Speed un elemento. En este caso <i>fly</i> 30.
Entrada a probar	$L_{Speed}.add(fly, 30)$
Resultado	Se agregó a L_{Speed} el valor <i>fly</i> 30.

Cuadro 7.9: Prueba de caja negra: Propiedad Challenge.

ID	PCN009
Nombre	PCN_ChangeChallenge
Descripción	Cambiar la propiedad Challenge.
Precondición	El nuevo valor debe pertenecer a $L_{challenge} \leftarrow \{0, 1/8, 1/4, 1/2, 1, 2, 3, \dots, 28, 29, 30\}$, con sus respectivos valores de xp y $proficiency$ como en las figuras 5.23 y 5.24.
Resultado esperado	El nuevo valor de “challenge” debe ser 5. Por lo tanto, $xp \leftarrow 1,800$ y $proficiency \leftarrow 3$.
Entrada a probar	$value \leftarrow 5$
Resultado	$challenge \leftarrow 5$

Cuadro 7.10: Prueba de caja negra: Propiedad Special Traits.

ID	PCN010
Nombre	PCN_AddSpecialTraits
Descripción	Agregar a la lista de SpecialTraits un elemento. Además, agregar el nombre y descripción de la clase.
Precondición	-
Resultado esperado	El nuevo valor de “name” y “description” debe ser el ingresado.
Entrada a probar	$name \leftarrow \text{“AngelicWeapons”}$, $description \leftarrow \text{“The solar’s weapon attacks are magical. When the solar hits with any weapon, the weapon deals an extra 6d8 radiant damage (included in the attack).”}$.
Resultado	Se agregó a $L_{SpecialTraits}$ el elemento “Angelic Weapons.”.

Cuadro 7.11: Prueba de caja negra: Propiedad Action.

ID	PCN011
Nombre	PCN_AddAction
Descripción	Agregar a la lista de Actions un elemento.
Precondición	-
Resultado esperado	La nueva acción es agregada a $L_{actions}$.
Entrada a probar	$name \leftarrow \text{"Fire ray"}, typeAction \leftarrow RangeAttack,$ $bonusAttack \leftarrow 0, minAction \leftarrow 5, maxAction \leftarrow$ $10, hitDieAction \leftarrow 1, dieAction \leftarrow d4,$ $bonusDamageAction \leftarrow 1, typeDamageAction \leftarrow$ $Fire$
Resultado	Se agregó a $L_{actions}$ el elemento "Fire ray".

7.1.2. Administrar PC

Cuadro 7.12: Prueba de caja negra: Cambio de race.

ID	PCN012
Nombre	PCN_ChangeRace
Descripción	Cambiar <i>race</i> de un PC.
Precondición	El PC debe tener una <i>race</i> distinta seleccionada.
Resultado esperado	Las propiedades de la nueva <i>race</i> deben cambiarse y/o agregarse. Además, estas propiedades no pueden ser quitadas.
Entrada a probar	$race \leftarrow \text{"HighElf"}(1)$
Resultado	Se agregaron y/o cambiaron las propiedades asociadas a "HighElf"(1).

Cuadro 7.13: Prueba de caja negra: Cambio de class.

ID	PCN013
Nombre	PCN_ChangeClass
Descripción	Cambiar <i>class</i> de un PC.
Precondición	El PC debe tener una <i>class</i> distinta seleccionada.
Resultado esperado	Las propiedades de la nueva <i>class</i> deben cambiarse y/o agregarse. Además, estas propiedades no pueden ser quitadas.
Entrada a probar	<i>class</i> ← “Wizard”(3)
Resultado	Se agregaron y/o cambiaron las propiedades asociadas a “Wizard”(3).

Cuadro 7.14: Prueba de caja negra: Cambio de background.

ID	PCN014
Nombre	PCN_ChangeBackground
Descripción	Cambiar <i>background</i> de un PC.
Precondición	El PC debe tener un <i>background</i> distinto seleccionado.
Resultado esperado	Las propiedades del nuevo <i>background</i> deben cambiarse y/o agregarse. Además, estas propiedades no pueden ser quitadas.
Entrada a probar	<i>background</i> ← “Soldier”(2)
Resultado	Se agregaron y/o cambiaron las propiedades asociadas a “Soldier”(2).

7.1.3. Administrar aplicación

Cuadro 7.15: Prueba de caja negra: Registro de usuario.

ID	PCN015
Nombre	PCN_Register
Descripción	Registrar un usuario.
Precondición	El “ <i>userName</i> ” y “ <i>email</i> ” no deben estar registrados con anterioridad.
Resultado esperado	Un usuario registrado en la aplicación.
Entrada a probar	<i>userName</i> ← “ <i>sergiodnd</i> ”, <i>email</i> ← “ <i>sc-fa.sergio@gmail.com</i> ”, <i>password</i> ← “ <i>qwe123abc</i> ”, <i>repeatPassword</i> ← “ <i>qwe123abc</i> ”.
Resultado	El usuario ha sido registrado en la base de datos. Además, se envió al email utilizado un <i>token</i> de activación.

Cuadro 7.16: Prueba de caja negra: Validación de usuario.

ID	PCN016
Nombre	PCN_Validate
Descripción	Validar un usuario mediante un <i>token</i> enviado al momento de registrarse.
Precondición	El usuario debe estar registrado, pero no activado.
Resultado esperado	La propiedad “ <i>isActive</i> ” de “ <i>User</i> ” debe ser “ <i>true</i> ”.
Entrada a probar	<i>userName</i> ← “ <i>sergiodnd</i> ”, <i>password</i> ← “ <i>qwe123abc</i> ”, <i>token</i> ← “ <i>BYR4C</i> ”
Resultado	La propiedad “ <i>isActive</i> ” es “ <i>true</i> ”.

Cuadro 7.17: Prueba de caja negra: Inicio de sesión usuario.

ID	PCN017
Nombre	PCN_Login
Descripción	Iniciar sesión mediante credenciales de usuario.
Precondición	El usuario que inicia sesión debe tener registrado y validado.
Resultado esperado	Una cookie de sesión es creada y almacenada.
Entrada a probar	$userName \leftarrow "sergiodnd", password \leftarrow "qwe123abc"$.
Resultado	La <i>cookie</i> fue creada.

7.1.4. Administrar Encuentro

Cuadro 7.18: Prueba de caja negra: Agregar participante a encuentro.

ID	PCN018
Nombre	PCN_AddUserCombat
Descripción	Agregar un usuario a un encuentro.
Precondición	El usuario debe estar registrado.
Resultado esperado	El usuario sea agregado a la lista de participantes pertenecientes a un encuentro.
Entrada a probar	$L_{participantes}.add(Masterdnd)$
Resultado	El usuario fue asociado al encuentro como participante.

Cuadro 7.19: Prueba de caja negra: Agregar NPC a encuentro.

ID	PCN019
Nombre	PCN_AddNPC
Descripción	Agregar un NPC a un encuentro.
Precondición	El NPC debe haber sido creado con anterioridad por el DM.
Resultado esperado	El NPC sea agregado a la lista de participantes pertenecientes a un encuentro.
Entrada a probar	<i>L_{npc}.add("RedDragon")</i>
Resultado	El NPC fue agregado al encuentro.

Cuadro 7.20: Prueba de caja negra: Agregar PC a encuentro.

ID	PCN020
Nombre	PCN_AddPC
Descripción	Agregar un PC a un encuentro.
Precondición	El PC debe haber sido creado con anterioridad por algún participante de este.
Resultado esperado	El PC sea agregado a la lista de participantes pertenecientes a un encuentro.
Entrada a probar	<i>L_{pc}.add("Name")</i>
Resultado	El PC fue agregado al encuentro.

Cuadro 7.21: Prueba de caja negra: Realizar daño a un personaje.

ID	PCN021
Nombre	PCN_DoDamageCharacter
Descripción	Aplicar daño a un personaje seleccionado.
Precondición	El NPC o PC debe haber sido seleccionado con anterioridad para dañarlo. El HP del personaje seleccionado para este caso es 6.
Resultado esperado	El personaje disminuye sus HP, dependiendo del valor ingresado.
Entrada a probar	$damage \leftarrow 5$
Resultado	El nuevo valor de HP es 1.

Cuadro 7.22: Prueba de caja negra: Curar a un personaje.

ID	PCN022
Nombre	PCN_HealCharacter
Descripción	Aplicar daño a un personaje seleccionado.
Precondición	El NPC o PC debe haber sido seleccionado con anterioridad para curarlo. El HP del personaje seleccionado para este caso es 1.
Resultado esperado	El personaje aumenta sus HP, dependiendo del valor ingresado.
Entrada a probar	$heal \leftarrow 5$
Resultado	El nuevo valor de HP es 6.

Cuadro 7.23: Prueba de caja negra: Agregar efecto a un personaje.

ID	PCN023
Nombre	PCN_AddEffectCharacter
Descripción	Agregar efecto un personaje seleccionado.
Precondición	El NPC o PC debe haber sido seleccionado con anterioridad para agregar efecto.
Resultado esperado	El personaje agrega a su lista de <i>Effects</i>
Entrada a probar	<i>Effects.add(blinded(0))</i>
Resultado	El efecto fue agregado a la lista de efectos del personaje.

Cuadro 7.24: Prueba de caja negra: Avanzar al siguiente turno en combate.

ID	PCN024
Nombre	PCN_NextTurnCombat
Descripción	Avanzar al siguiente turno, lo que cambia el personaje actual al siguiente en orden de iniciativa.
Precondición	El combate debe tener 2 o más participantes.
Resultado esperado	El personaje que sigue al activo en la lista de personajes en el encuentro, debe ser el nuevo personaje activo.
Entrada a probar	<i>nextTurn()</i>
Resultado	El personaje activo cambió.

7.2. Pruebas de usabilidad

La prueba de usabilidad USE fue aplicada a 9 usuarios. Para esto se les entrega un manual de uso de la aplicación (una copia de este manual se encuentra en el Anexo B.1) con las tareas que se pueden realizar en la aplicación. Estas son:

1. Registro de usuario.
2. Inicio de sesión.
3. Crear NPCs.

4. Crear PCs.
5. Crear Encuentros.
6. Ejecutar encuentros como DM.
7. Ejecutar encuentros como Jugador.
8. Desarrollar cuestionario.

Durante la realización de estas pruebas la aplicación fue publicada en un servidor web. Además, el cuestionario fue realizado a través de Google Forms y su realización de este puede ser visto en el Anexo B.3.

7.2.1. Resultados

Los resultados que se obtuvieron en la realización de la prueba de usabilidad se muestran en la Figura 7.1. En este gráfico de barras se observa el valor promedio de cada pregunta agrupado en las categorías (*Usefulness*, *Ease of use*, *Ease of learning*, *Satisfaction*). Además, la desviación estándar de cada una de estas categorías.

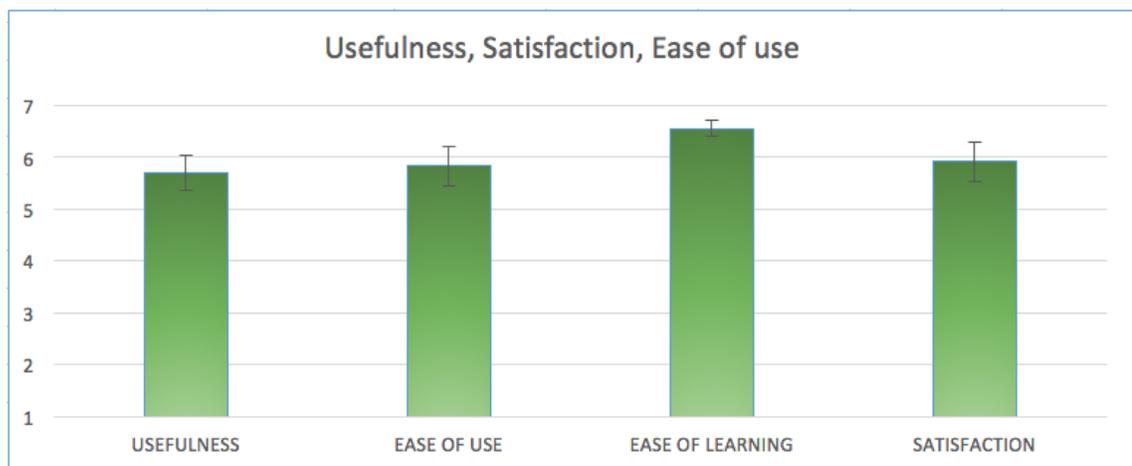


Figura 7.1: Resultados pruebas de usabilidad.

De este gráfico se puede concluir que los promedios de cada uno de los aspectos del gráfico es mayor a los 5,6 puntos de 7. Esto se traduce en que las calificaciones están por encima del 80 % de aprobación. Específicamente, cada aspecto completa un porcentaje de:

- *Usefulness*: Los usuarios consideraron en un 81,35 % promedio que la aplicación les sería útil.
- *Ease of use*: Los usuarios consideraron en un 83,26 % promedio que la aplicación es fácil de usar.
- *Ease of learning*: Los usuarios consideraron en un 93,65 % promedio que la aplicación es fácil de aprender.
- *Satisfaction*: Los usuarios consideraron en un 84,58 % promedio quedaron satisfechos.

Se puede inferir que, con estos niveles de utilidad, satisfacción, facilidad de uso y aprendizaje, la usabilidad de la aplicación web es alta. Lo que se traduce en que los usuarios que probaron la aplicación aprendieron fácilmente a usarla y quedaron satisfechos con la interacción que tuvieron con ella.

8. Conclusiones y trabajo futuro

En el presente documento de memoria se detalla el desarrollo de una aplicación web de apoyo al juego D&D 5e. Esta aplicación tiene la finalidad de ayudar a los jugadores a crear personajes de manera interactiva. Además, permite que jugadores pueden ser partícipes de combates en los que visualizan sus estadísticas en tiempo real.

Para su desarrollo se utilizó una metodología ágil de una cantidad constante de retroalimentación del usuario, para concretar esto, se planificaron diez iteraciones que abarcan los objetivos específicos comprometidos (sección 1.4.2). Además, se diseñaron, desarrollaron, e implementaron módulos en la aplicación web para satisfacer a los tres primeros objetivos que apuntan a permitir la creación de PCs, NPCs y, facilitar los encuentros interactivos y en tiempo real. Además, se realizaron pruebas de caja negra para las funcionalidades de la aplicación.

También, se realizaron pruebas de usabilidad mediante participación de usuarios, que arrojaron resultados positivos en cuanto a la utilidad, satisfacción, facilidad de uso y de aprendizaje. Dado esto, se puede inferir que la aplicación cumple el objetivo general planteado (Apoyar a los jugadores).

Como trabajo futuro en el ámbito de desarrollo, se pueden agregar funcionalidades a la aplicación tales como la creación de campañas y de mapas, con el objetivo entregar más herramientas a los jugadores. En general, seguir escalando el software verticalmente con el objetivo de propiciar a que se continúe el desarrollo. La aplicación es de código abierto bajo la licencia GNU GPL (General Public License, ver anexo C) con el propósito de entregar a la comunidad una base sólida que permita competir con las aplicaciones existentes.

Bibliografía

- [1] About materialize. <http://archives.materializecss.com/0.100.2/about.html>. Consultado el 6 de mayo de 2018.
- [2] Asp.net mvc 5. <https://docs.microsoft.com/es-es/aspnet/mvc/mvc5>. Consultado el 3 de mayo de 2018.
- [3] Asp.net overview. <https://docs.microsoft.com/es-es/aspnet/overview>. Consultado el 3 de mayo de 2018.
- [4] Entity framework overview. <https://docs.microsoft.com/es-mx/dotnet/framework/data/adonet/ef/overview>. Consultado el 3 de mayo de 2018.
- [5] Get started with the .net framework. <https://docs.microsoft.com/es-es/dotnet/framework/get-started>. Consultado el 2 de mayo de 2018.
- [6] History forty years of adventure. <http://dnd.wizards.com/dungeons-and-dragons/what-dd/history/history-forty-years-adventure>. Consultado el 29 de abril de 2018.
- [7] Introduction to asp.net web programming using the razor syntax (c#). <https://docs.microsoft.com/es-mx/aspnet/web-pages/overview/getting-started/introducing-razor-syntax-c>. Consultado el 2 de mayo de 2018.
- [8] New to the game what is d&d? <http://dnd.wizards.com/dungeons-and-dragons/what-is-dd>. Consultado el 09 de abril de 2017.
- [9] Overview of the .net framework. <https://docs.microsoft.com/es-es/dotnet/framework/get-started/overview>. Consultado el 2 de mayo de 2018.

- [10] Product overview: Baldur's gate enhanced edition. <http://dnd.wizards.com/products/digital-games/pcmac/baldurs-gate-pcmac>. Consultado el 15 de junio de 2018.
- [11] Product overview: Dungeon master's guide. <http://dnd.wizards.com/products/tabletop-games/rpg-products/dungeon-masters-guide>. Consultado el 3 de mayo de 2018.
- [12] Product overview: Neverwinter. <http://dnd.wizards.com/products/digital-games/console-pcmac/neverwinter>. Consultado el 15 de junio de 2018.
- [13] Product overview: Player's handbook. http://dnd.wizards.com/products/tabletop-games/rpg-products/rpg_playershandbook. Consultado el 3 de mayo de 2018.
- [14] Sql server database engine. <https://docs.microsoft.com/es-mx/dotnet/framework/data/adonet/ef/overview>. Consultado el 4 de mayo de 2018.
- [15] Sql server documentation. <https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation?view=sql-server-2017>. Consultado el 3 de mayo de 2018.
- [16] The top 11 dungeons & dragons games of all-time. <http://www.ign.com/articles/2014/02/05/the-top-11-dungeons-dragons-games-of-all-time>. Consultado el 29 de abril de 2018.
- [17] The values of extreme programming. <http://www.extremeprogramming.org/values.html>. Consultado el 30 de mayo de 2018.
- [18] What is jquery? <http://jquery.com>. Consultado el 3 de mayo de 2018.
- [19] Adam Freeman. *Pro ASP .NET MVC 5*. NY: Apress, 2013.
- [20] Glenn E Krasner, Stephen T Pope, et al. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, 1(3):26–49, 1988.

- [21] Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. Prentice Hall PTR, 2004.
- [22] Oriol LLauradó. La escala de likert: qué es y cómo utilizarla. *Obtenido de Blog de Netquest: <http://www.netquest.com/blog/es/la-escala-delikert-que-es-y-como-utilizarla>*, 2014.
- [23] Arnold Lund. Measuring usability with the use questionnaire. 8, 2001.
- [24] Roger S Pressman and Jose Maria Troya. *Ingeniería del software*. McGraw Hill, 1988.
- [25] Wizards RPG Team. *Dungeon Master's Guide (D&D Core Rulebook)*. Wizards of the Coast, 2014.
- [26] Wizards RPG Team. *Monster Manual (D&D Core Rulebook)*. Wizards of the Coast, 2014.
- [27] Wizards RPG Team. *Player's Handbook (Dungeons & Dragons)*. Wizards of the Coast, 2014.

ANEXOS

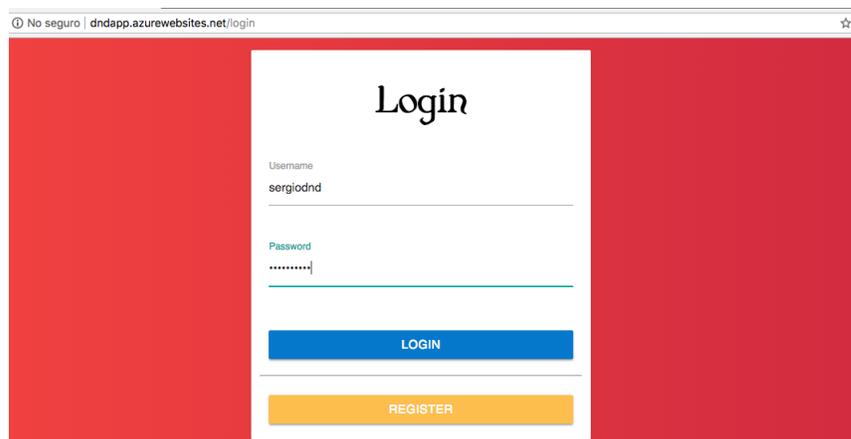
A. Diagramas

B. Pruebas de usabilidad

B.1. Manual de uso

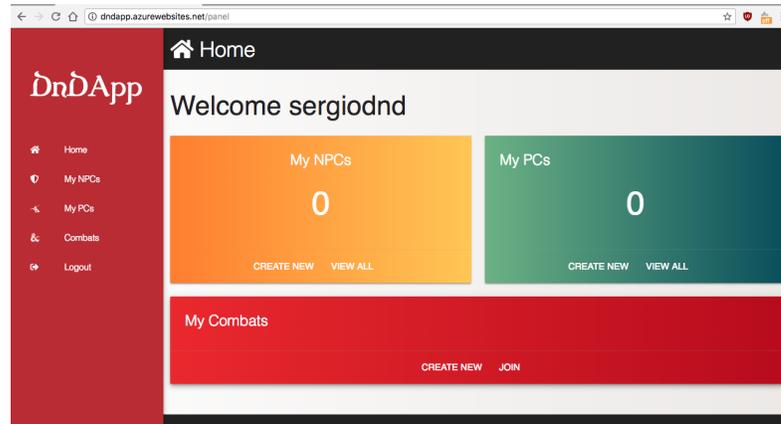
B.1.1. Inicio de sesión

1. Entre a la siguiente URL: <http://dndapp.azurewebsites.net/login> y complete sus datos:



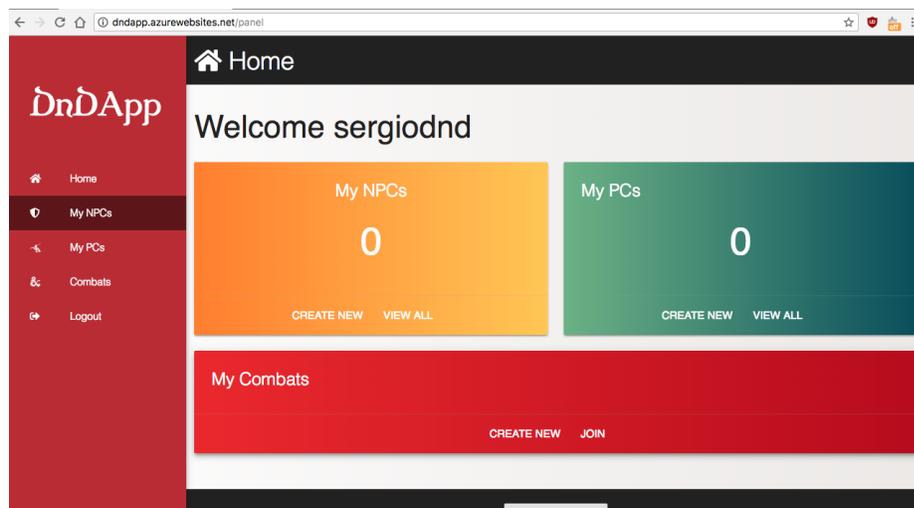
The screenshot shows a web browser window with the address bar displaying "No seguro | dndapp.azurewebsites.net/login". The main content area has a red background. In the center, there is a white box titled "Login". Inside this box, there are two input fields: "Username" with the text "sergiodnd" and "Password" with masked characters. Below these fields are two buttons: a blue "LOGIN" button and a yellow "REGISTER" button.

2. Si el inicio de sesión es correcto, entonces llegarás al inicio de la aplicación.

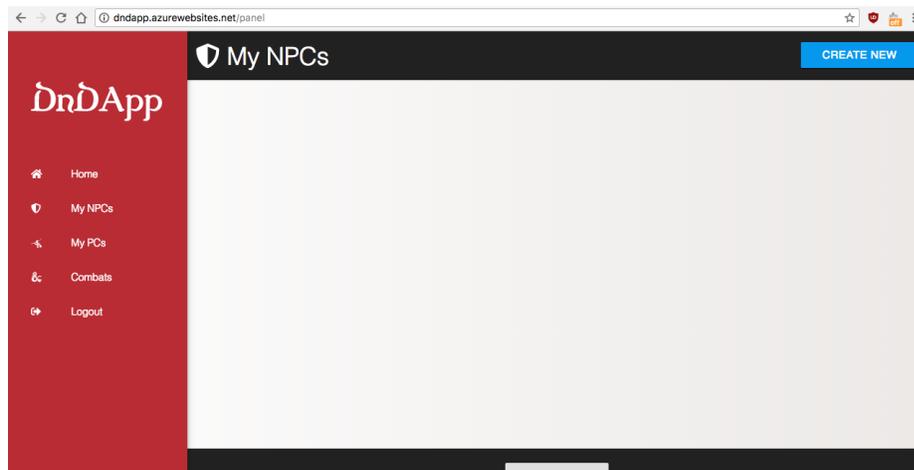


B.1.2. Crear NPCs

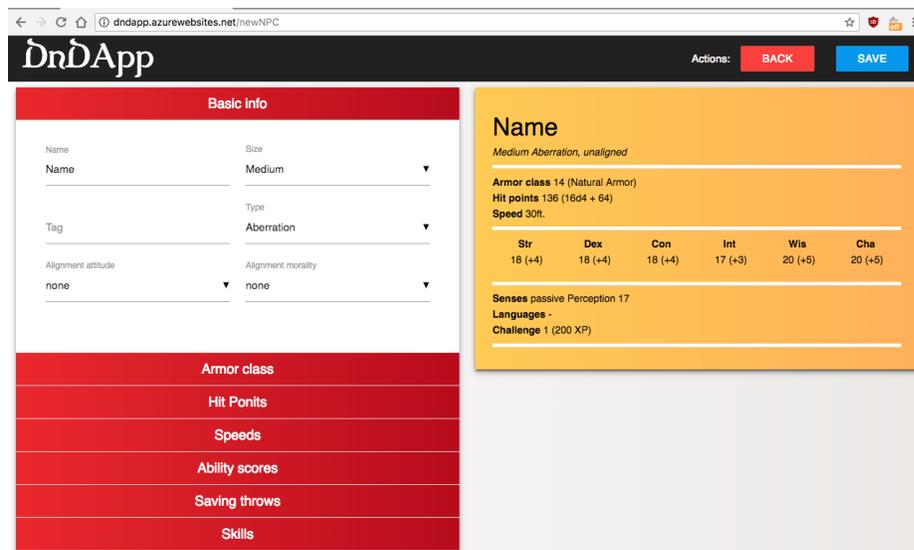
1. Para crear personajes NPC debe seleccionar **MY NPCs** en el panel izquierdo:



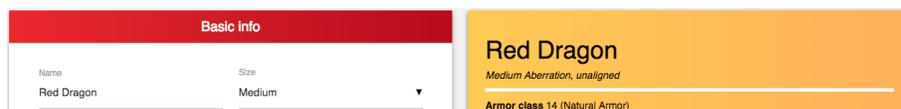
2. Luego en la parte superior de derecha seleccionar la opción **CREATE NEW**



3. Podrá comenzar a crear el personaje:

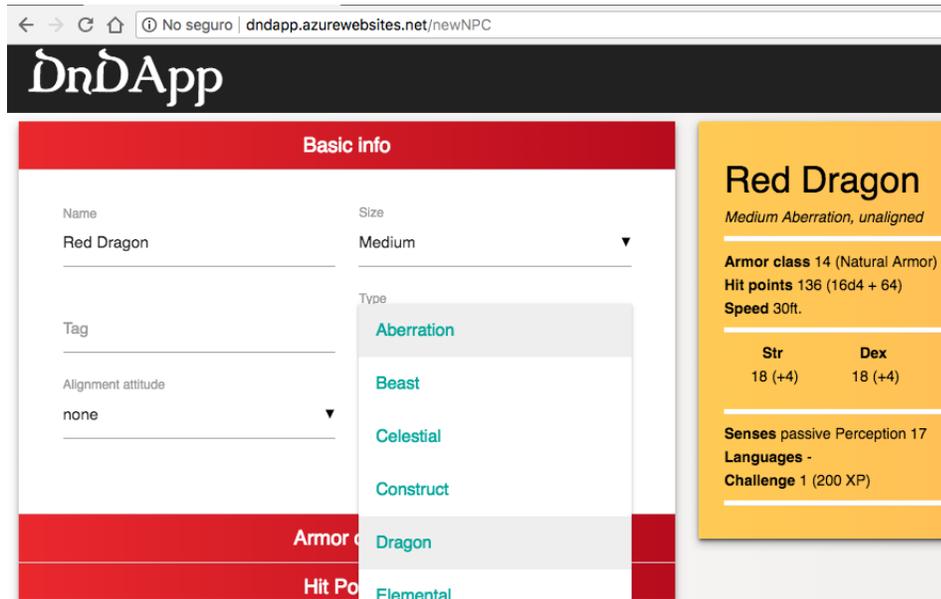


4. Podrá modificar los parámetros del NPC al lado izquierdo y ver sus cambios en el lado derecho. Por ejemplo, para modificar el nombre del personaje, debe cambiarlo en el campo *name* del formulario y luego dejar de hacer foco en este.

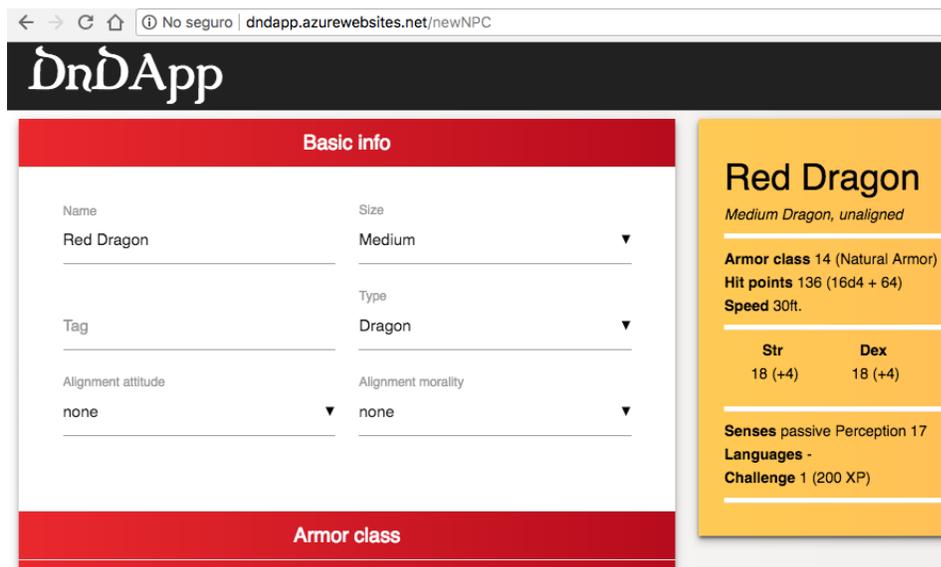


5. También hay características que cambian seleccionando el valor en un lista. Por

ejemplo, el parámetro *type* en donde solo es necesario seleccionar el valor para que se actualice.



cambia a...



6. Puede navegar entre las distintas acciones seleccionando en los nombres de las características que desea cambiar de su NPC:

The screenshot shows the 'Basic Info' section of the DnDApp interface. The left sidebar contains a menu with the following items: Basic Info, Armor class, Hit Points, Speeds, Ability scores, and Saving throws. The main content area displays the following information for a Red Dragon:

- Red Dragon** (Medium Dragon, neutral)
- Armor class 14 (Natural Armor)
- Hit points 152 (16d10 + 64)
- Speed 30ft.

Str	Dex	Con	Int	Wis	Cha
18 (+4)	18 (+4)	18 (+4)	17 (+3)	20 (+5)	20 (+5)

Additional details include: Saving Throws Con +6, Cha +7; Senses passive Perception 17; Languages -; and Challenge 1 (200 XP). Below the ability scores, there are checkboxes for Strength, Dexterity, Constitution, Intelligence, Wisdom, and Charisma. Constitution and Charisma are checked.

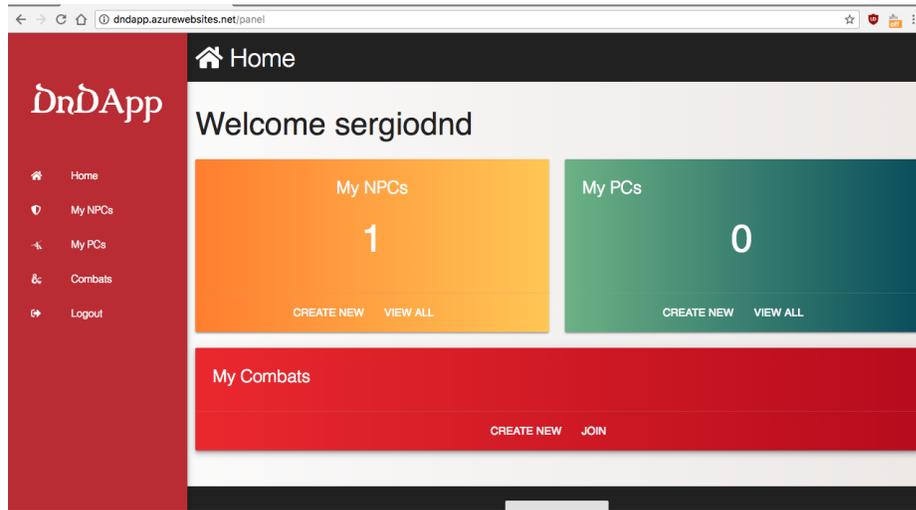
The screenshot shows the 'Actions' section of the DnDApp interface. The left sidebar contains a menu with the following items: Actions. The main content area displays the following information for a Red Dragon:

- Red Dragon** (Medium Dragon, neutral)
- Armor class 14 (Natural Armor)
- Hit points 152 (16d10 + 64)
- Speed 30ft.

Str	Dex	Con	Int	Wis	Cha
18 (+4)	18 (+4)	18 (+4)	17 (+3)	20 (+5)	20 (+5)

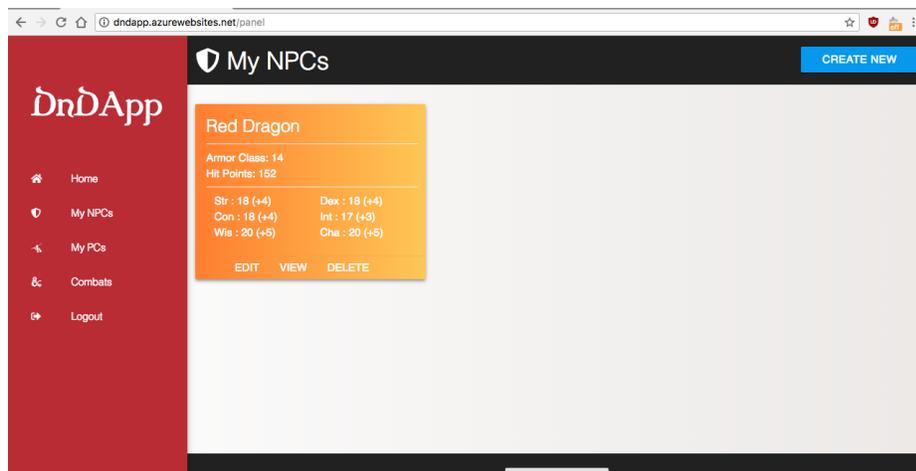
Additional details include: Saving Throws Con +9, Cha +10; Skills Stealth +9, History +8; Damage Vulnerabilities Force; Senses Truesight 20 ft., passive Perception 20; Languages -; and Challenge 16 (15000 XP). Below the ability scores, there is an 'Actions' section with the following text: **Fire ray.** Melee Weapon Attack: +9 to hit, reach 10ft., Hit: 6 (1 d4 + 4) Fire damage.

7. Cuando termine de crear su personaje puede guardarlo seleccionando la opción **SAVE** en la esquina superior derecha. Luego volverá al menú principal, y podrá ver como se ha creado su personaje. Luego OK.



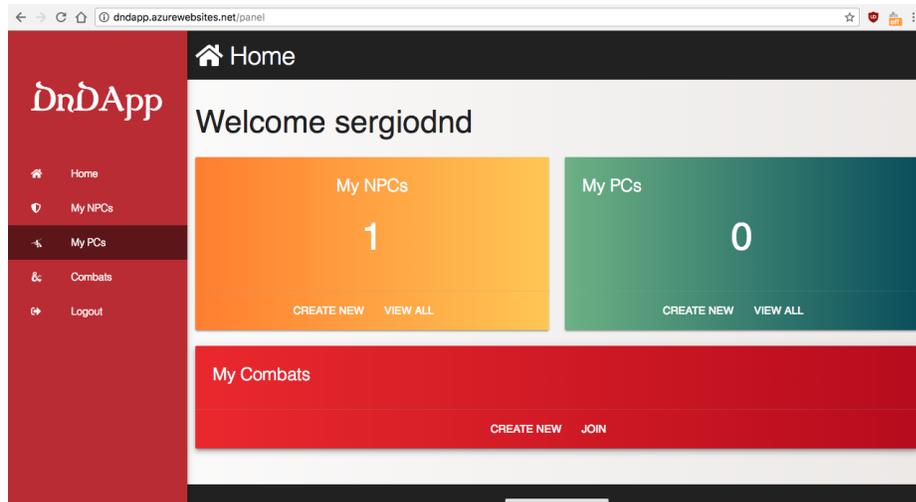
Notamos que el contador de sus NPCs se ha incrementado.

8. Volviendo a seleccionar la opción **MY NPCs** en la barra de navegación izquierda, podrá ver su personaje recién creado

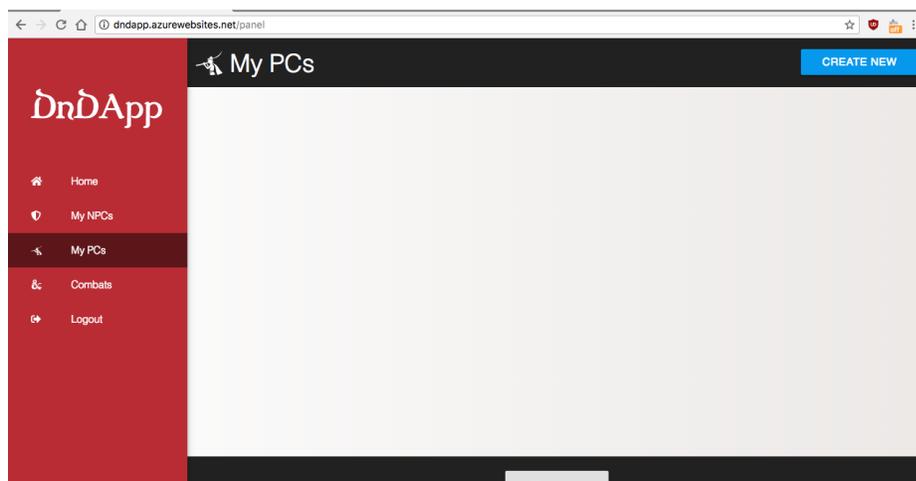


B.1.3. Crear PC

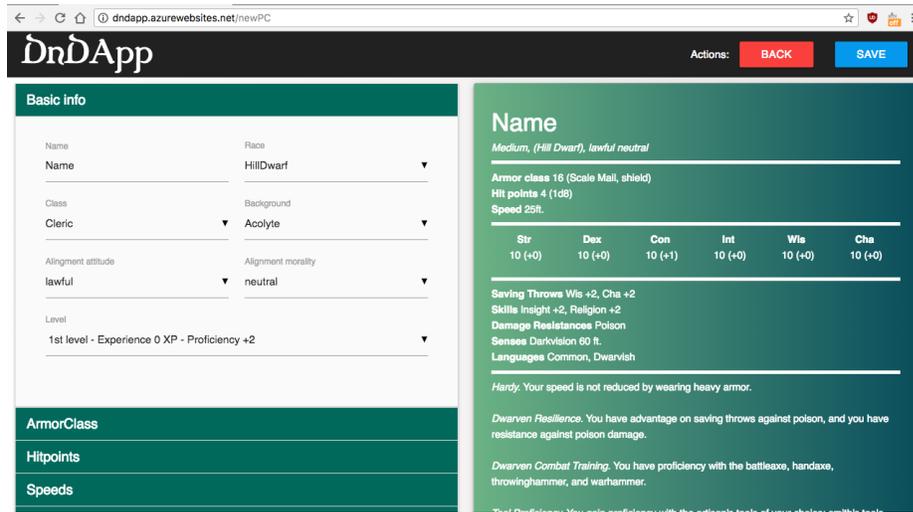
1. Para crear personajes PCs debe seleccionar **MY PCs** en el panel izquierdo:



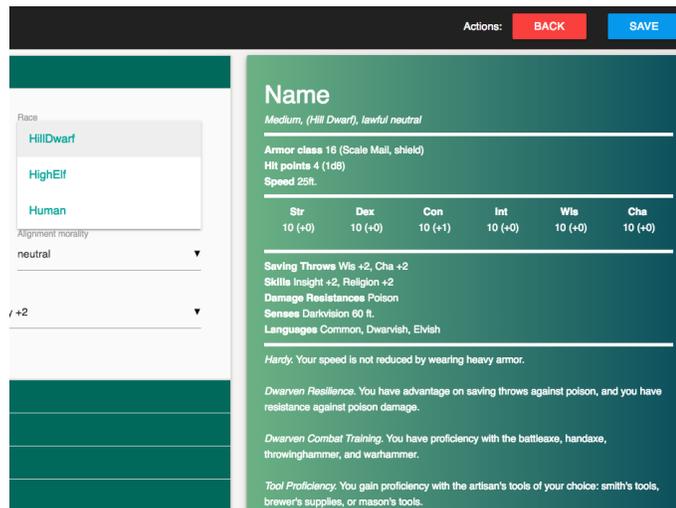
2. Luego en la parte superior de derecha seleccionar la opción **CREATE NEW**



3. Podrá comenzar a crear el personaje:



4. Podrá elegir que raza será su personaje, lo que cambiará automáticamente las características específicas de ella:



Race
HighElf ▼

Background
Acolyte ▼

Alignment morality
neutral ▼

+2 ▼

Name

Medium, (High Elf), lawful neutral

Armor class 17 (Scale Mail, shield)
Hit points 4 (1d8)
Speed 30ft.

Str	Dex	Con	Int	Wis	Cha
10 (+0)	10 (+1)	10 (+0)	10 (+0)	10 (+0)	10 (+0)

Saving Throws Wis +2, Cha +2
Skills Insight +2, Religion +2
Damage Resistances Poison
Senses Darkvision 60 ft.
Languages Common, Elvish, Dwarvish

Keen Senses. You have proficiency in the Perception skill.

Fey Ancestry. You have advantage on saving throws against being charmed, and magic can't put you to sleep.

Trance. Elves don't need to sleep. Instead, they meditate deeply, remaining semiconscious, for 4 hours a day. (The Common word for such meditation is 'trance.'). While meditating, you can dream after a fashion; such dreams are actually mental exercises that have become reflexive through years of practice. After resting in this way, you gain the same benefit that a human does from 8 hours of sleep.

Race
Human ▼

Background
Acolyte ▼

Alignment morality
neutral ▼

+2 ▼

Name

Medium, (Human), lawful neutral

Armor class 16 (Scale Mail, shield)
Hit points 4 (1d8)
Speed 25ft.

Str	Dex	Con	Int	Wis	Cha
10 (+0)	10 (+0)	10 (+0)	10 (+0)	10 (+0)	10 (+0)

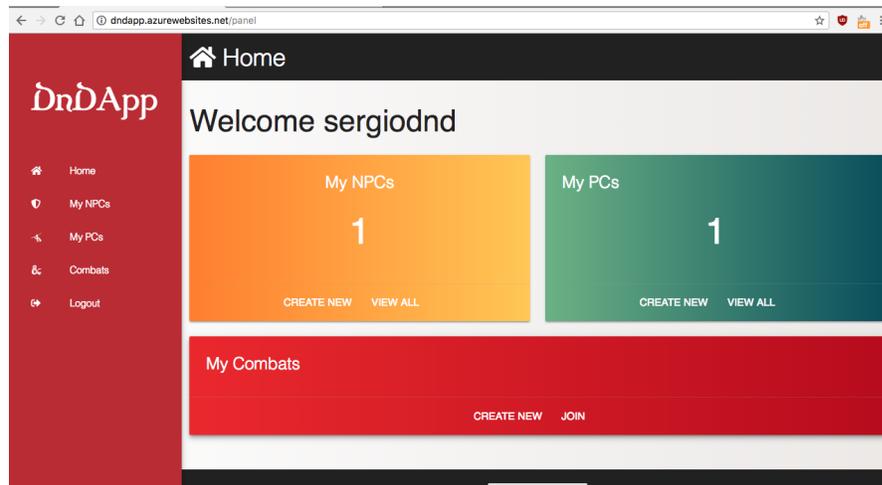
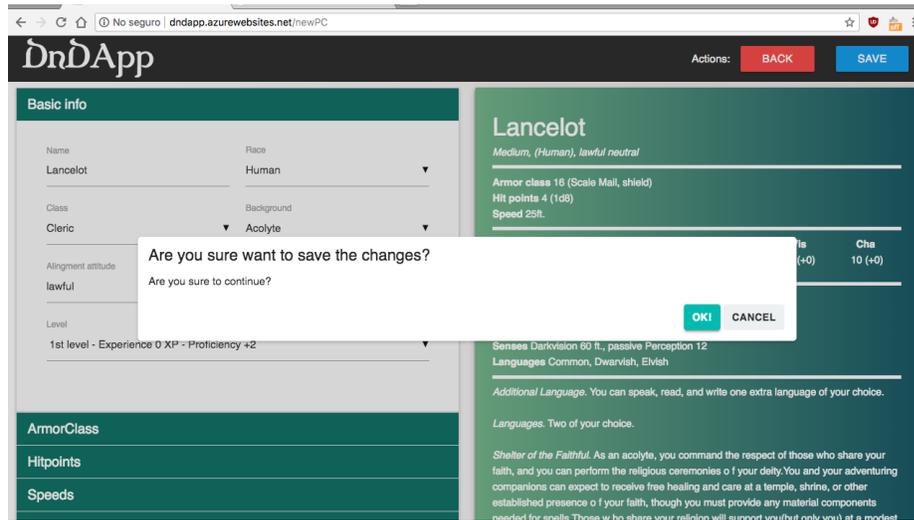
Saving Throws Wis +2, Cha +2
Skills Insight +2, Religion +2
Damage Resistances Poison
Senses Darkvision 60 ft.
Languages Common, Dwarvish, Elvish

Additional Language. You can speak, read, and write one extra language of your choice.

Languages. Two of your choice.

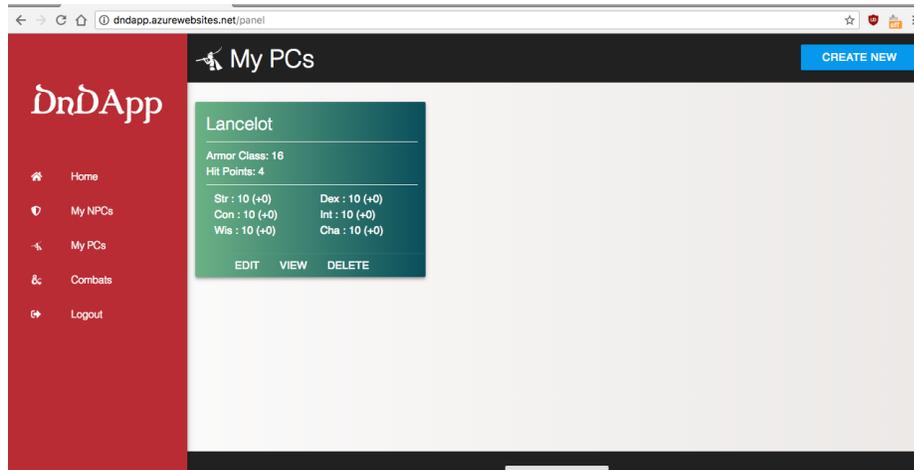
Shelter of the Faithful. As an acolyte, you command the respect of those who share your faith, and you can perform the religious ceremonies of your deity. You and your adventuring companions can expect to receive free healing and care at a temple, shrine, or other established presence of your faith, though you must provide any material components needed for spells. Those who share your religion will support you (but only you) at a modest lifestyle. You might also have ties to a specific temple dedicated to your chosen deity or

5. Cuando termine de crear su personaje puede guardarlo seleccionando la opción **SAVE** en la esquina superior derecha. Luego volverá al menú principal, y podrá ver como se ha creado su personaje. Luego OK.



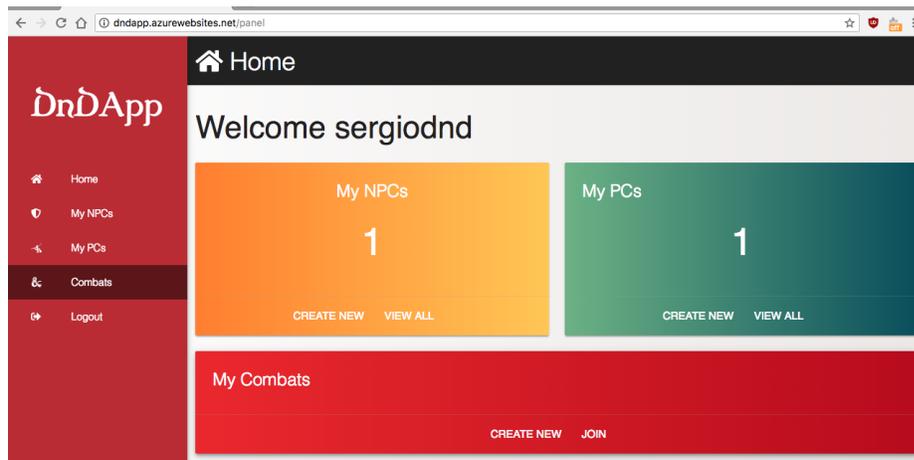
Notamos que el contador de sus PCs se ha incrementado.

6. Volviendo a **MY PCs** podrá ver su personaje creado:

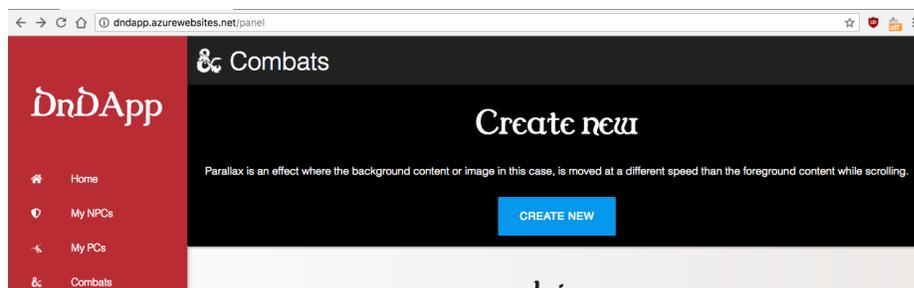


B.1.4. Crear encuentros

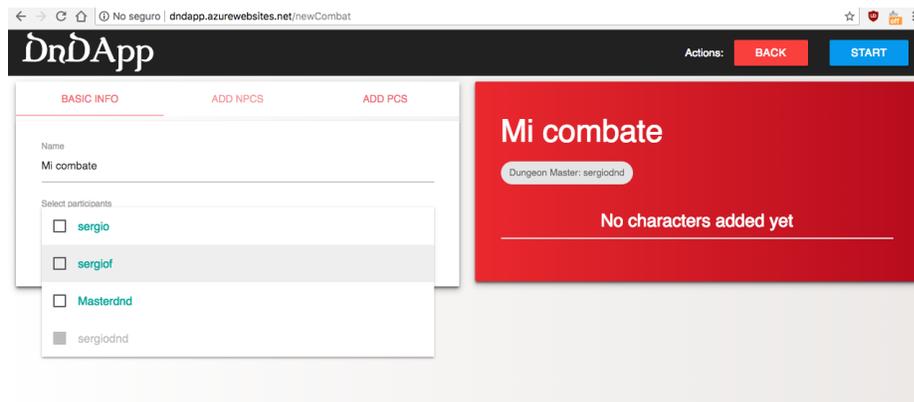
1. Puede ser el DM de un encuentro creándolo. Para ello debe seleccionar **Combat**



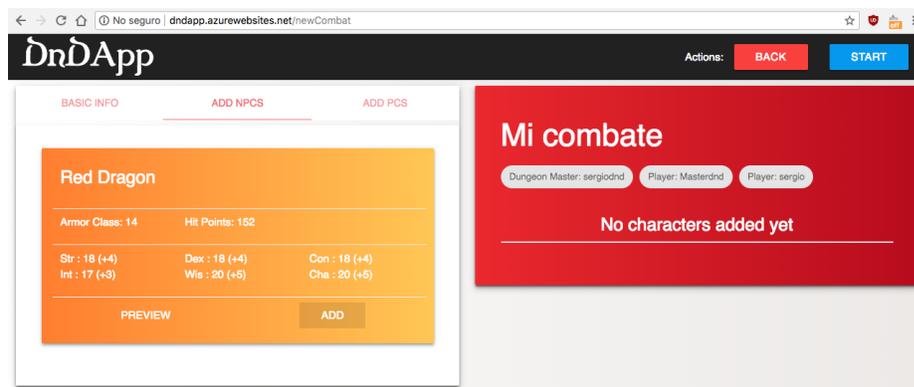
2. Luego, seleccionar la opción **CREATE NEW**



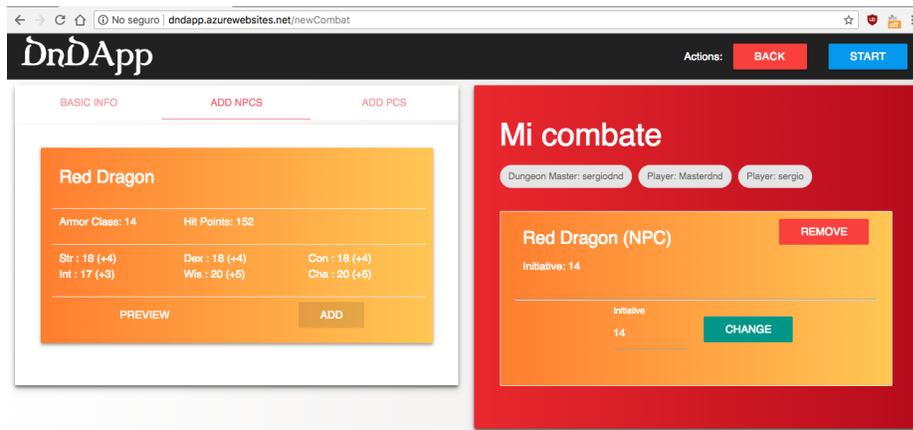
- Primero, podrá poner nombre y agregar los participantes. (Ud ya esta seleccionado automáticamente.)



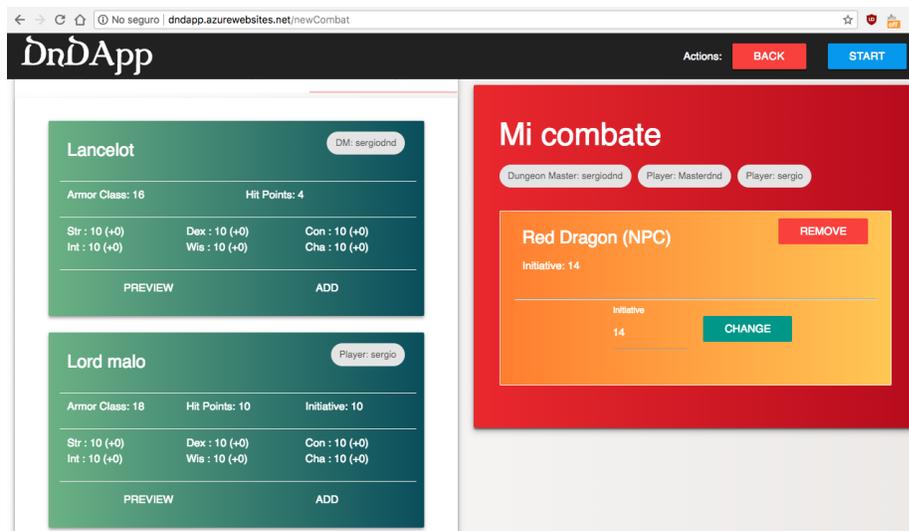
- Segundo, podrá agregar personajes NPC a su combate. Seleccionando ADD en el NPC que quiera agregar.



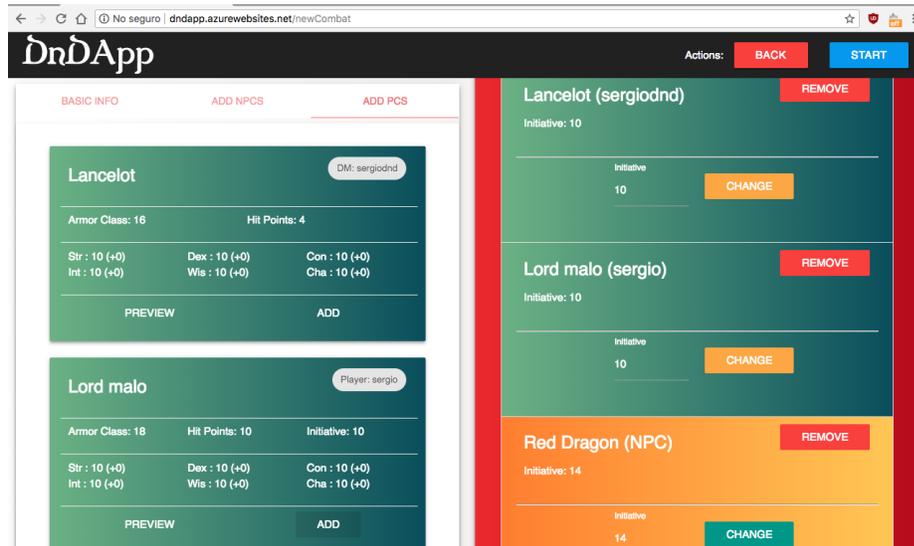
Se cargará a su encuentro el NPC seleccionado.



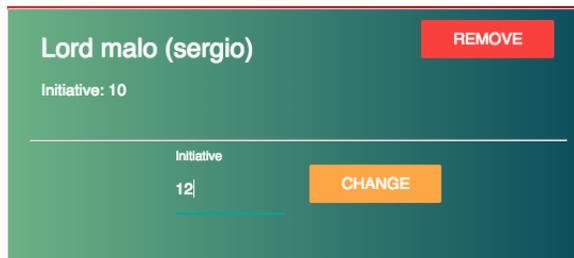
5. Tercero, podrá agregar personajes PC a su combate, suyo o de los participantes del combate.



Se cargará una lista con los PC de los participantes para agregar al combate.



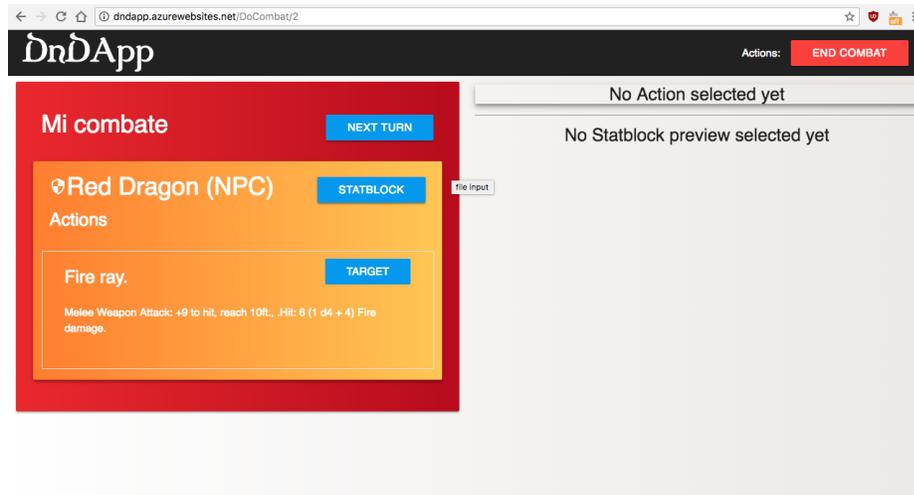
6. Puede ajustar la iniciativa de los personajes, mediante el cuadro de texto de abajo de su nombre y seleccionando CHANGE.



Cambia de 10 a 12



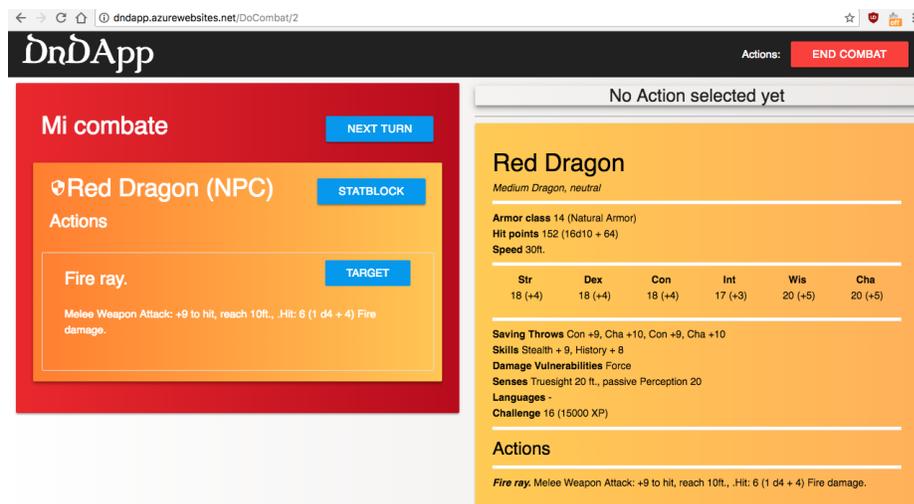
7. Si estima que todo es correcto, puede comenzar el combate seleccionando la opción **START** en la esquina superior derecha.



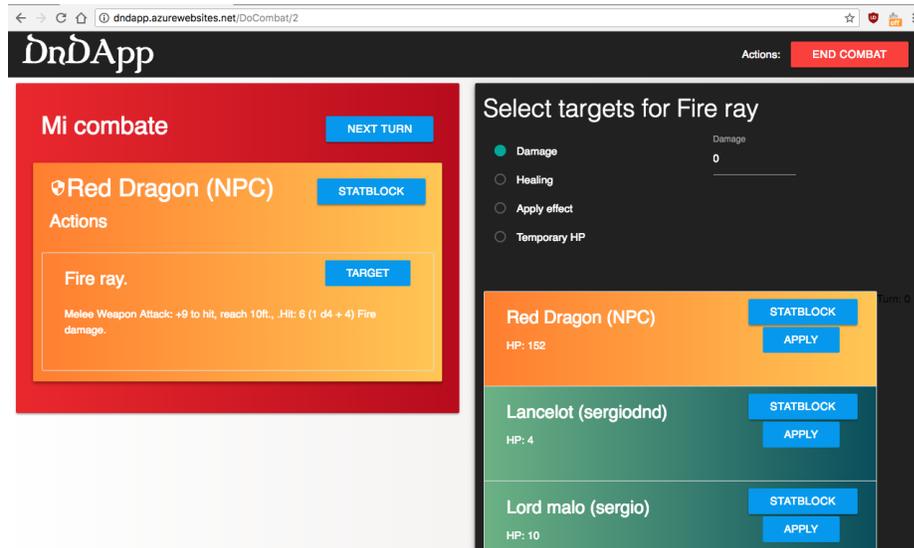
Ahora esta listo para comenzar su combate.

B.1.5. Ejecutar encuentros como DM

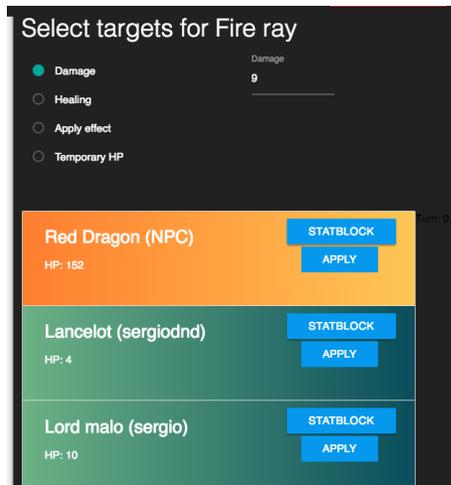
1. Siguiendo los pasos de la sección anterior ahora podrá ejecutar el combate como un DM. Verá a su izquierda al personaje que le corresponde el turno. Puede ver su statblock completo si selecciona la opción **STATBLOCK**.



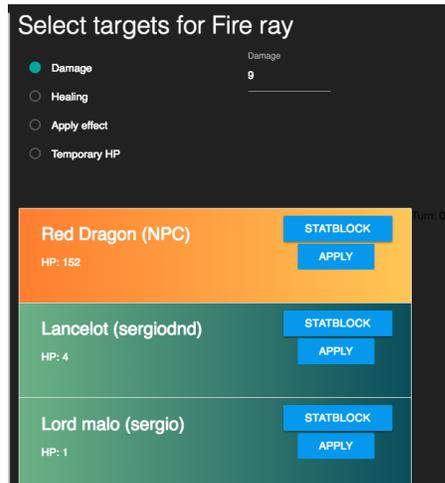
Además, puede seleccionar ejecutar una acción de su personaje, seleccionando su objetivo en la opción `fcolorboxWhiteNavyBlueTARGET`.



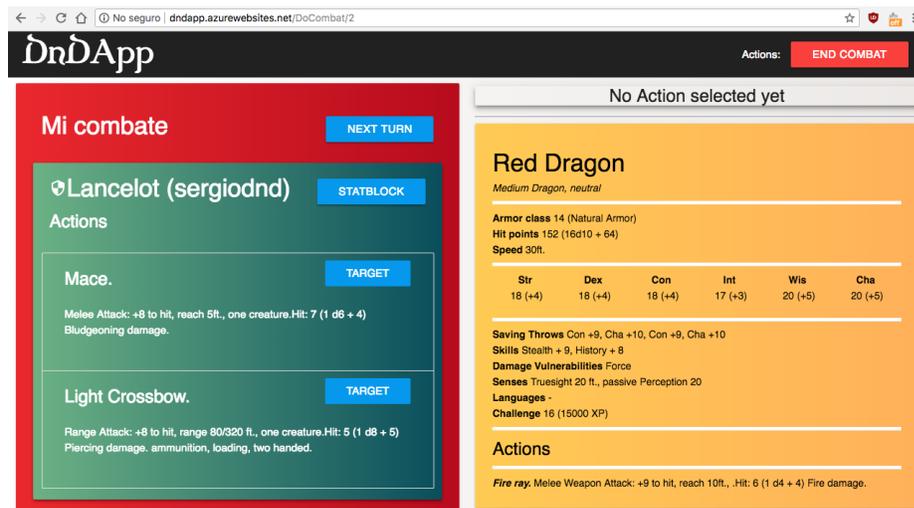
2. Ahora según avance su partida, ud. puede elegir entre las opciones [Damage, Healing, Apply effect] y llenar el cuadro de texto de al lado para dañar o curar o verá una lista de efectos que puede aplicar a los personajes.



En este caso se calculó que Red Dragon hará 9 puntos de daño a Lord Malo.



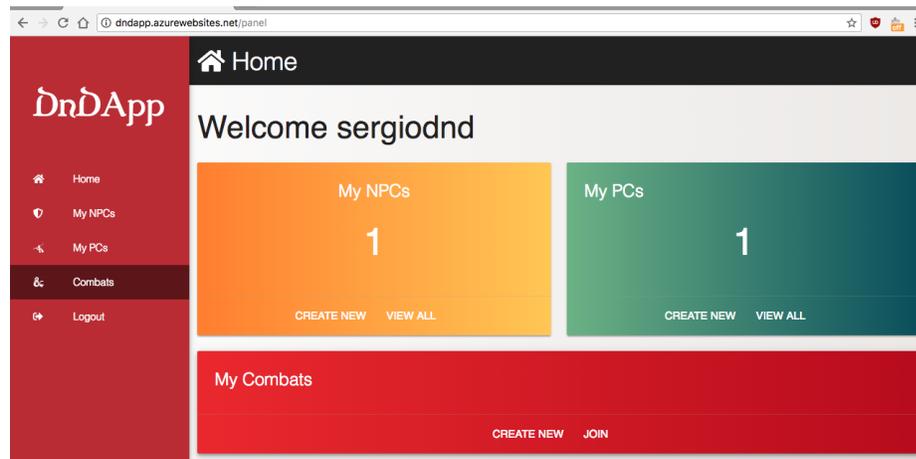
- Para avanzar en los turnos puede seleccionar la opción **NEXT TURN**. Lo cual cambiará el personaje actual por el siguiente en orden de iniciativa.



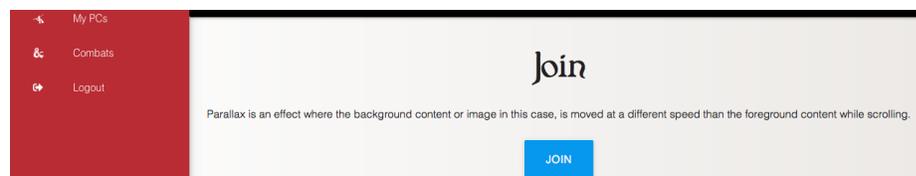
- Siga avanzando a través de su combate, hasta que decida que se decrete el fin de su combate. Seleccionando la opción **END COMBAT** en la esquina superior derecha.

B.1.6. Ejecutar encuentros como Jugador

- Puede participar en un encuentro que un DM haya creado. Para ello debe seleccionar **Combat**



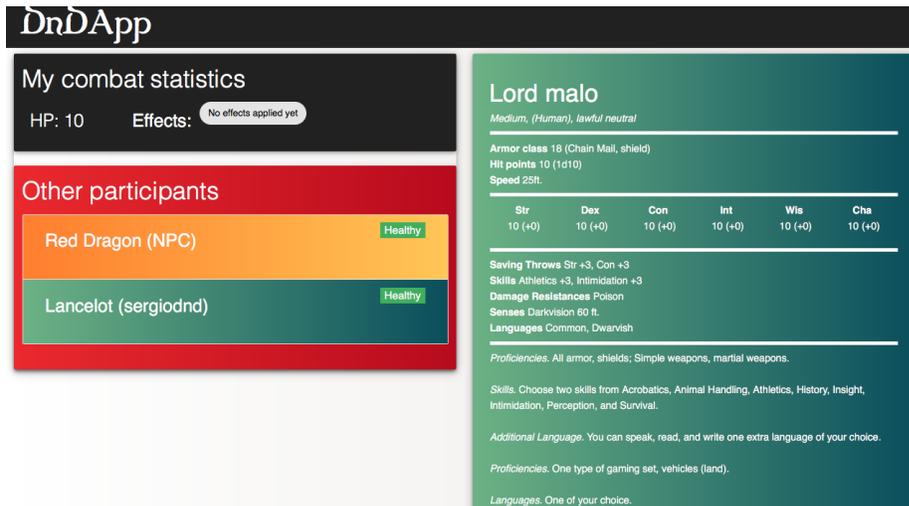
2. Luego seleccionando la opción **JOIN**.



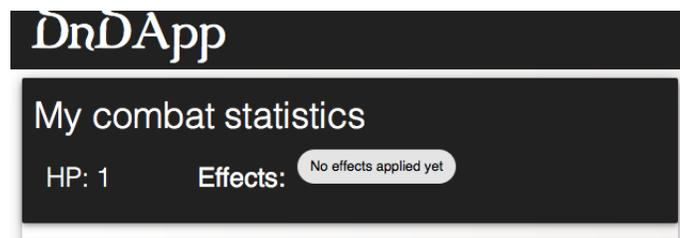
3. Verá los combates en los que esta participando. Para unirse a uno debe seleccionar la opción JOIN



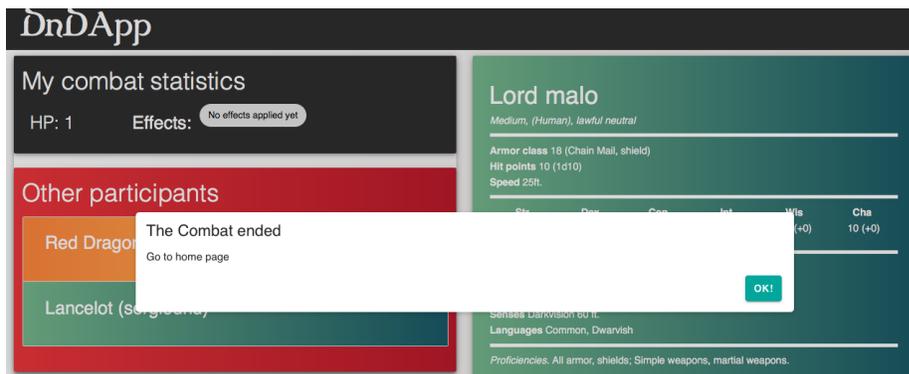
4. Aquí, verá sus estadísticas en tiempo real por lo que si recibe daño sus puntos de vida disminuirán, etc.



En este caso recibió daño, por lo que sus puntos de daño disminuirán.



5. Puede seguir viendo sus estadísticas hasta que el DM concluya en encuentro. La aplicación avisará que este terminó.



B.1.7. Agradecimientos y cuestionario

Agradezco su participación en la prueba de la aplicación web. Ahora lo invito a completar la siguiente encuesta en <https://goo.gl/forms/kobdmDsUZfNC2DyD2>.

Muchas Gracias.

B.2. Participantes prueba de usabilidad

ID	USERNAME	EMAIL	PASSWORD
7	RedDevil	sebafariasg@gmail.com	\$2a\$10\$L4mLIPvx4ahhEZ0y2g8gX...
8	LinkiShinoda	marcelo.araneda26@gmail.com	\$2a\$10\$fbVw2v9MbNN9hc1FuHc...
9	Kusanagi	jose.cortinez@gmail.com	\$2a\$10\$NSXqROu.rj8OHfcfexAs9...
11	Camila	camila.garcia.cgn@gmail.com	\$2a\$10\$XSPGHlqoriEeJ9138hRr7...
12	Onedeadshot	luciferondeath@gmail.com	\$2a\$10\$RwdfLEB1q35ATE3lh2jcze...
13	Litier	litier.4@gmail.com	\$2a\$10\$kEyv4N9k4bfcGUfc7pZ1o...
14	CossiLeon	javyfernandez9@gmail.com	\$2a\$10\$aQLg5/geQd6VqBFxt8ixS...
15	marcelmarch95	marcelmarch95@gmail.com	\$2a\$10\$zyzweEzEoHB2hMYSZ/u8h...
16	Josema	josema.vb@gmail.com	\$2a\$10\$Zw/Vyyx6mEjp8dRljTqD0...

B.3. Resultados cuestionario USE

Nombre de usuario

9 respuestas

RedDevil
LinkiShinoda
Kusanagi
Camila
Onedeashot
Litier
CossiLeon
marcelmarch95
Josema

Figura B.1: Usuarios prueba de usabilidad.

¿Quiénes respondieron?

Correo electrónico
sebafariasg@gmail.com
marcelo.araneda26@gmail.com
jose.cortinez@gmail.com
camila.garcia.cgn@gmail.com
luciferondeath@gmail.com
javier.arosmendoza@gmail.com
javyfernandez9@gmail.com
marcelmarch95@gmail.com
Josema.vb@gmail.com

Figura B.2: Email prueba de usabilidad.

1. Me ayudó a ser más efectivo.

9 respuestas

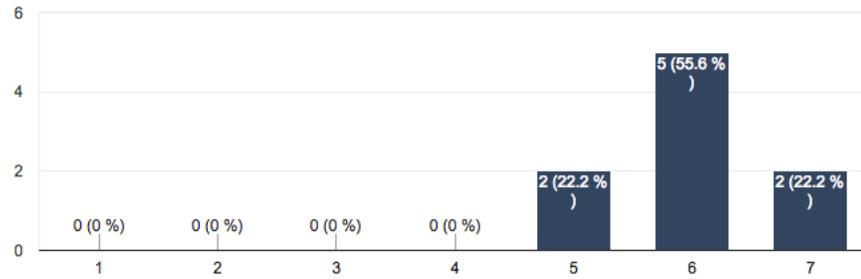


Figura B.3: Pregunta 1.

2. Me ayudó a ser más productivo.

9 respuestas

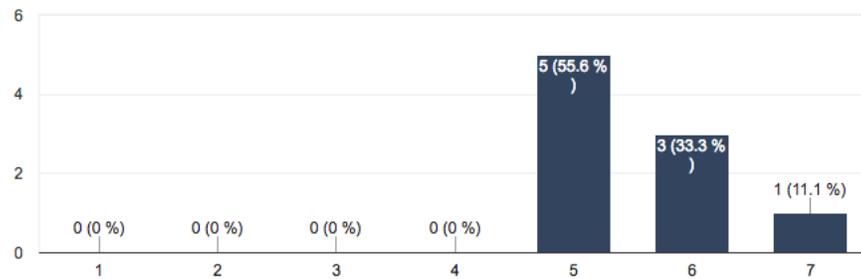


Figura B.4: Pregunta 2.

3. Es útil.

9 respuestas

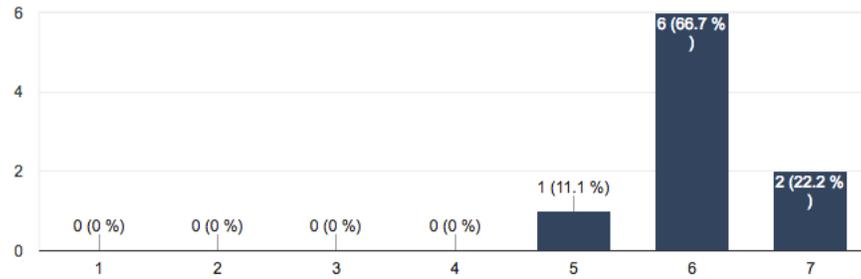


Figura B.5: Pregunta 3.

4. Me control sobre las actividades que realizaba anteriormente

9 respuestas

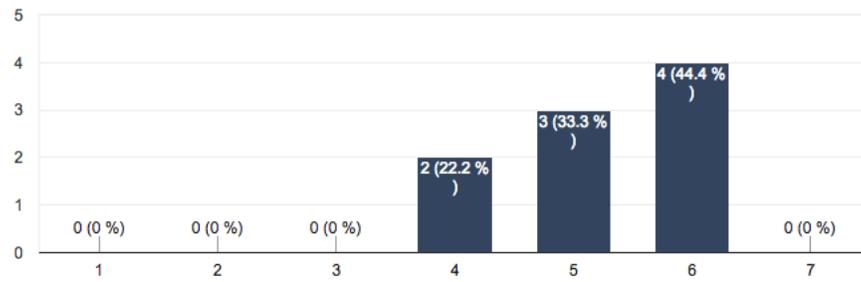


Figura B.6: Pregunta 4.

5. Hace que las cosas que quiero hacer sean más fáciles.

9 respuestas

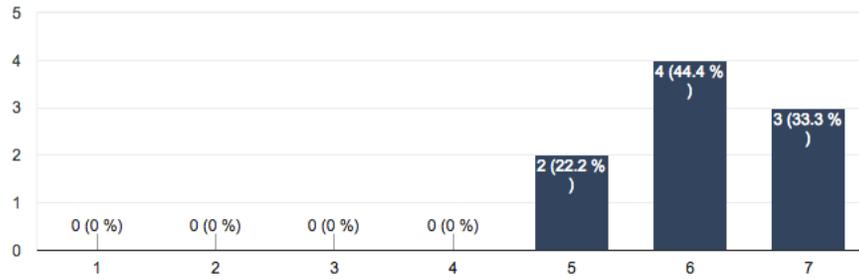


Figura B.7: Pregunta 5.

6. Hace que las cosas que quiero hacer sean más fáciles.

9 respuestas

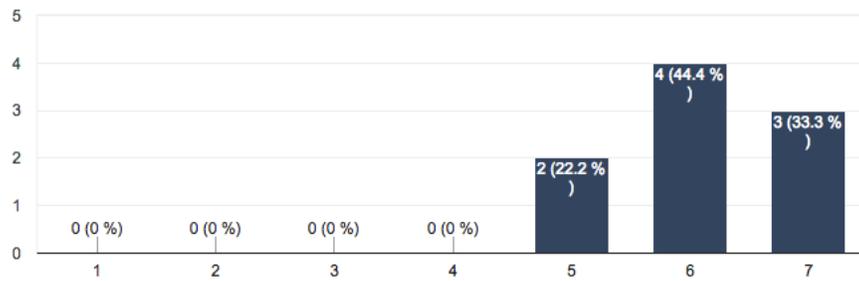


Figura B.8: Pregunta 6.

5. Hace que las cosas que quiero hacer sean más fáciles.

9 respuestas

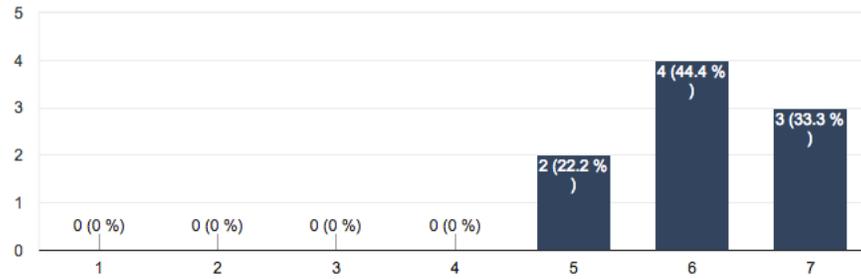


Figura B.9: Pregunta 5.

7. Me ahorra tiempo cuando lo uso.

9 respuestas

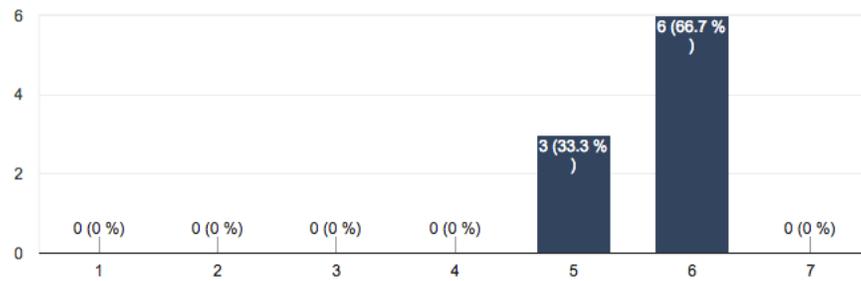


Figura B.10: Pregunta 7.

8. Satisface mis necesidades.

9 respuestas

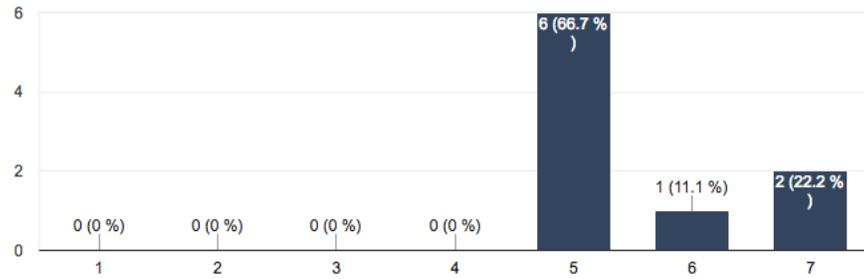


Figura B.11: Pregunta 8.

9. Hace todo lo que espero que haga.

9 respuestas

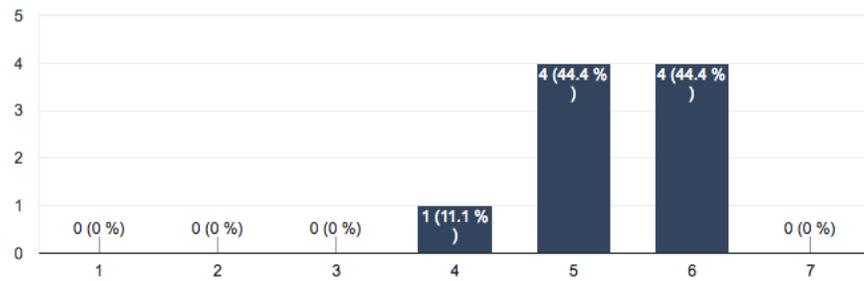


Figura B.12: Pregunta 9.

10. Es fácil de usar.

9 respuestas

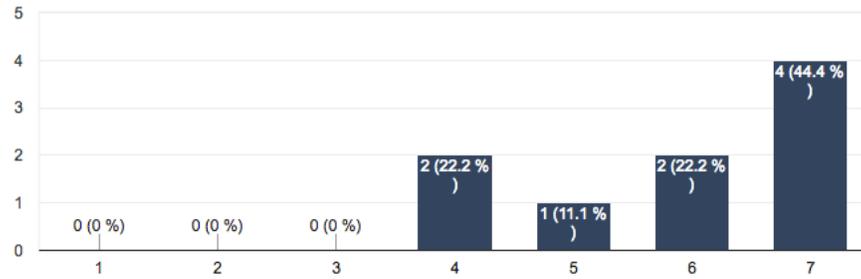


Figura B.13: Pregunta 10.

11. Es simple de usar.

9 respuestas

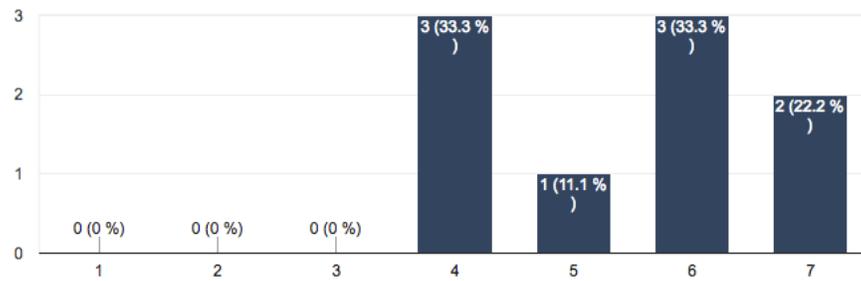


Figura B.14: Pregunta 11.

12. Es amigable.

9 respuestas

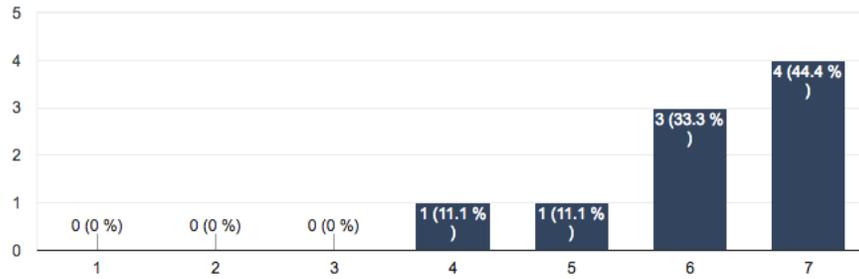


Figura B.15: Pregunta 12.

13. Requiere el menor número de pasos posibles para lograr lo que quiero hacer con él.

9 respuestas

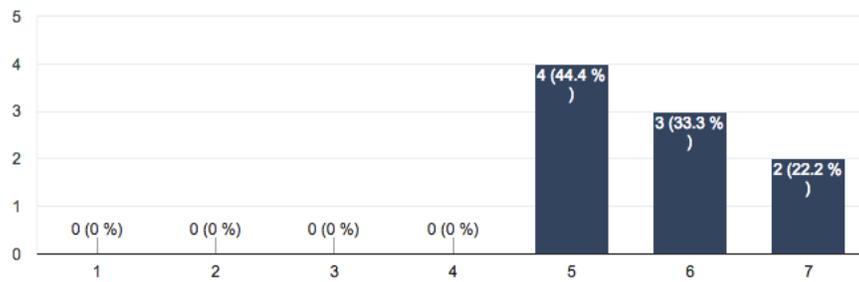


Figura B.16: Pregunta 13.

14. Es flexible.

9 respuestas

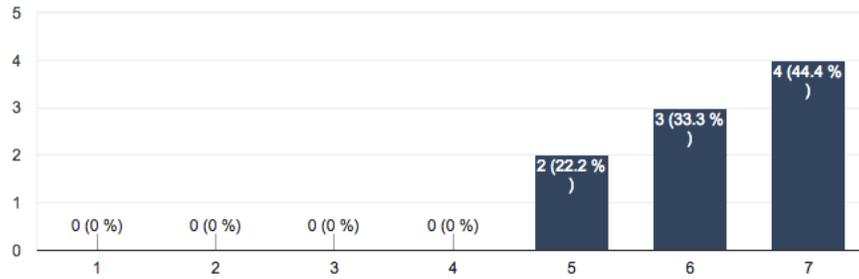


Figura B.17: Pregunta 14.

15. Usarlo no requiere esfuerzo.

9 respuestas

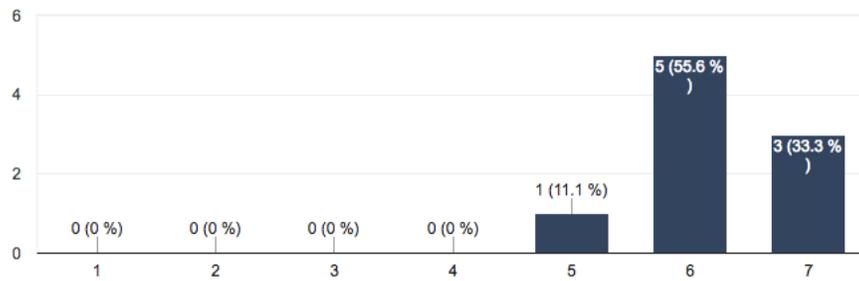


Figura B.18: Pregunta 15.

16. Puedo usarlo sin instrucciones.

9 respuestas

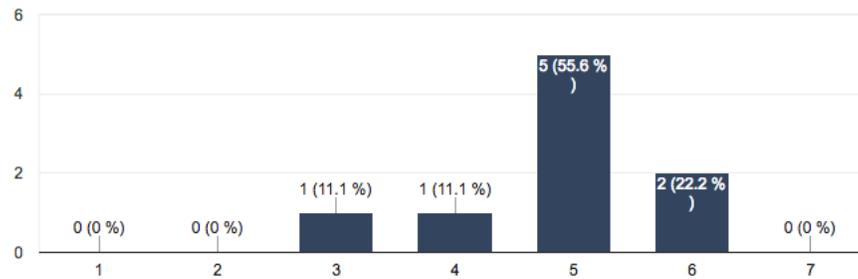


Figura B.19: Pregunta 16.

17. No noto incoherencias cuando la uso.

9 respuestas

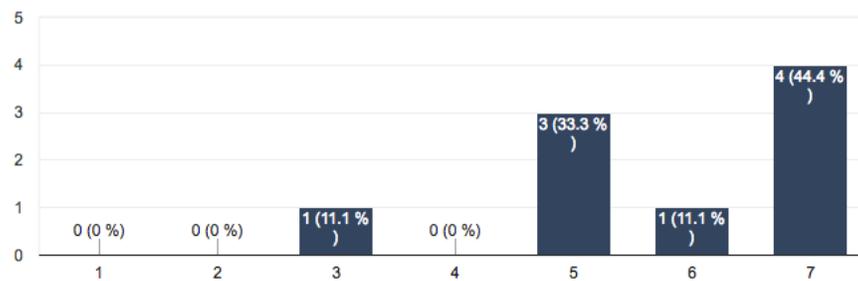


Figura B.20: Pregunta 17.

18. A los usuarios ocasionales y regulares les gustaría.

9 respuestas

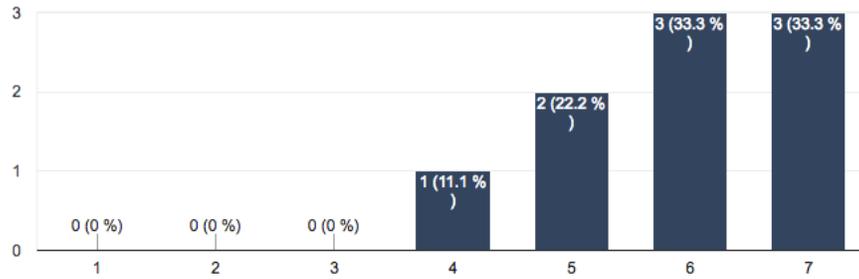


Figura B.21: Pregunta 18.

19. Puedo recuperarme de los errores en forma rápida y fácil.

9 respuestas

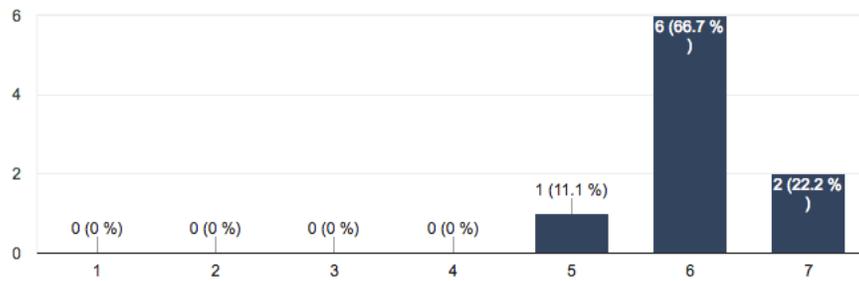


Figura B.22: Pregunta 19.

20. Puedo usarlo todo el tiempo satisfactoriamente.

9 respuestas

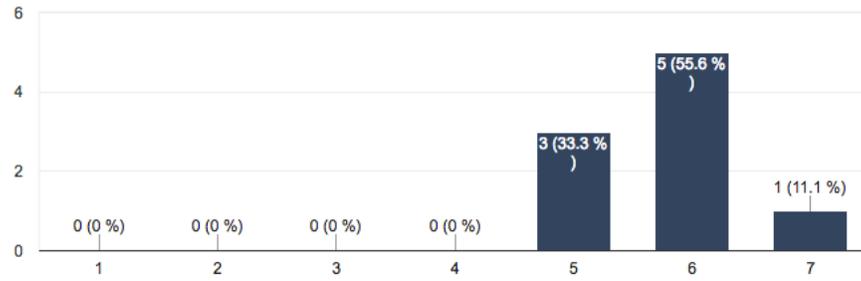


Figura B.23: Pregunta 20.

21. Aprendí a usarlo rápidamente.

9 respuestas

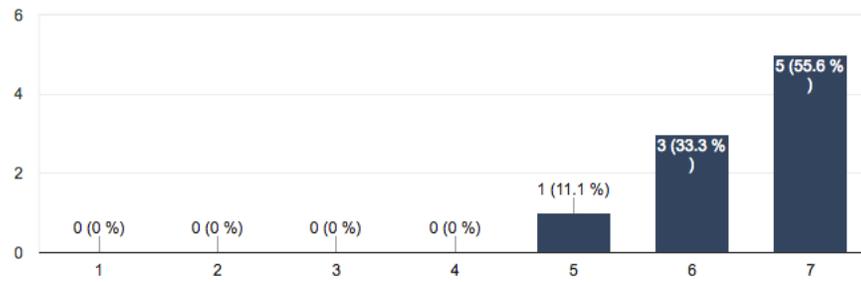


Figura B.24: Pregunta 21.

22. Recuerdo fácilmente cómo usarlo.

9 respuestas

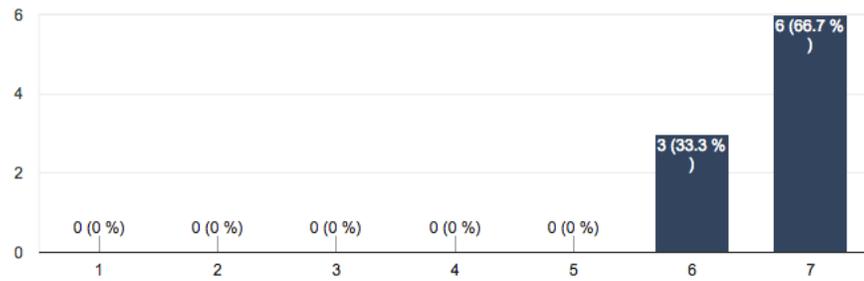


Figura B.25: Pregunta 22.

23. Rápidamente me volví hábil.

9 respuestas

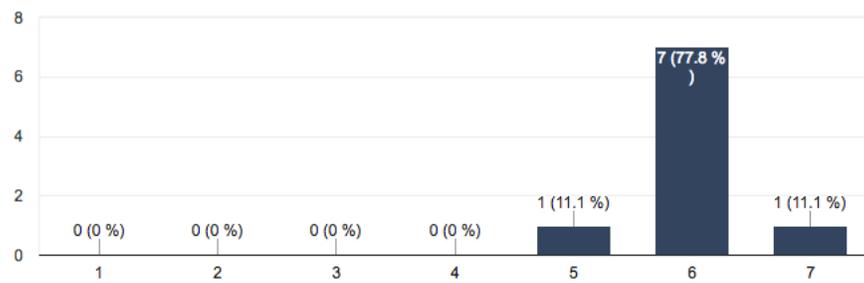


Figura B.26: Pregunta 23.

24. Estoy satisfecho con el uso.

9 respuestas

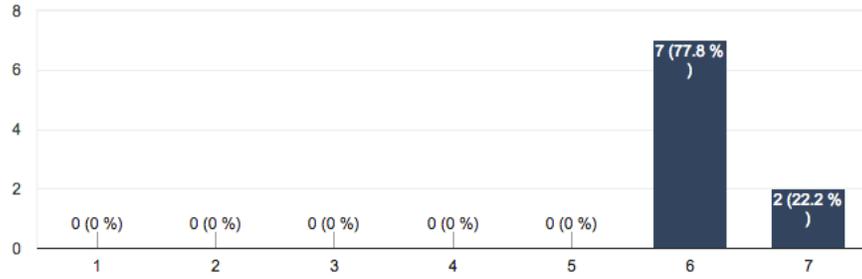


Figura B.27: Pregunta 24.

25. Se lo recomendaría a un amigo.

9 respuestas

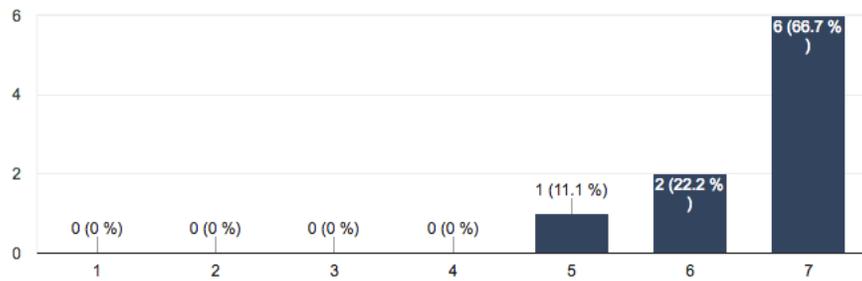


Figura B.28: Pregunta 25.

26. Es divertido de usar.

9 respuestas

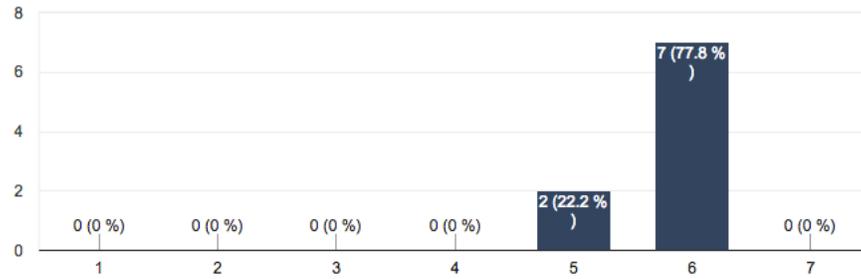


Figura B.29: Pregunta 26.

27. Funciona de la manera que yo quiero que funcione.

9 respuestas

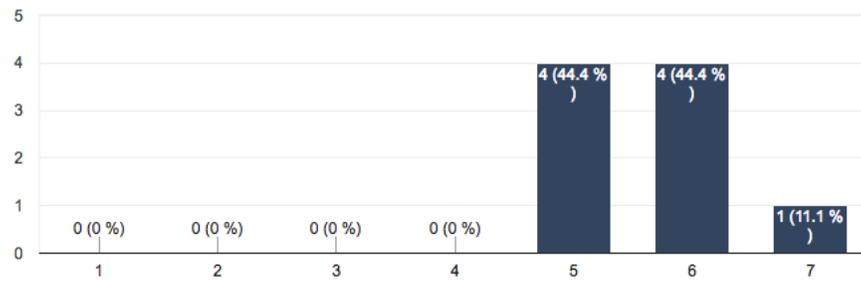


Figura B.30: Pregunta 27.

28. Es maravilloso.

9 respuestas

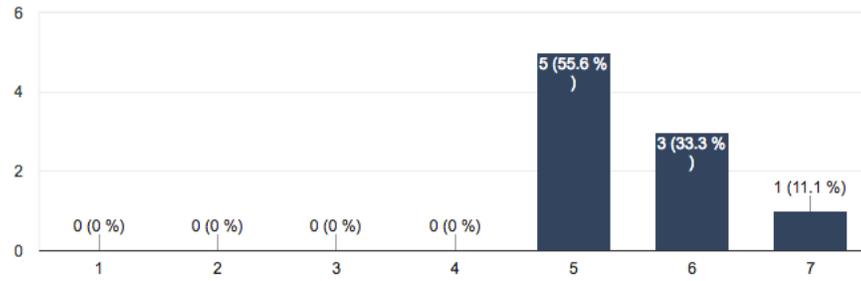


Figura B.31: Pregunta 28.

29. Siento que necesito tenerlo.

9 respuestas

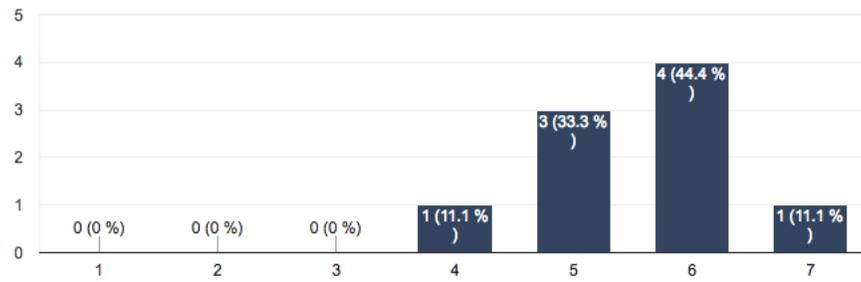


Figura B.32: Pregunta 29.

30. Es agradable de usar.

9 respuestas

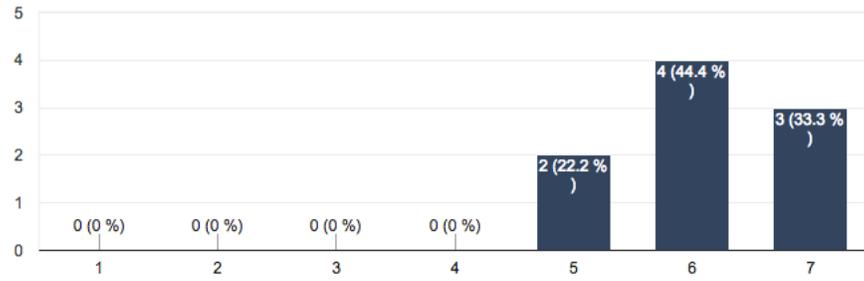


Figura B.33: Pregunta 30.

C. GNU General Public License

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular

programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without

conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining

whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a)* Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b)* Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c)* Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d)* Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e)* Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f)* Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying. If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its

coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL,

SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
```

```
Copyright (C) <textyear> <name of author>
```

```
This program is free software: you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.
```

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
```

```
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <https://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <https://www.gnu.org/licenses/why-not-lgpl.html>.