

MEC-SETEC

INSTITUTO FEDERAL DE MINAS GERAIS - *Campus* Formiga

Curso de Ciência da Computação

**DESENVOLVIMENTO DE UM GERENCIADOR INTELIGENTE PARA  
CAMPANHAS DE DUNGEON AND DRAGONS**

Samuel Gomes Duque

Orientador: Prof. Msc. Wallace de Almeida Rodrigues

Coorientador: Prof. Msc. Everthon Valadão dos Santos

Formiga - MG

2019

SAMUEL GOMES DUQUE

**DESENVOLVIMENTO DE UM GERENCIADOR INTELIGENTE PARA  
CAMPANHAS DE DUNGEON AND DRAGONS**

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Formiga, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Msc. Wallace de Almeida Rodrigues  
Coorientador: Prof. Msc. Everthon Valadão dos Santos

Formiga - MG

2019

004 Duque, Samuel Gomes.  
Desenvolvimento de um gerenciador inteligente para campanhas de  
Dungeon and Dragons / Samuel Gomes Duque. -- Formiga : IFMG, 2019.  
57p : il.

Orientador: Prof. Msc. Wallace de Almeida Rodrigues  
Coorientador Prof. Msc. Everthon Valadão dos Santos  
Trabalho de Conclusão de Curso – Instituto Federal de Educação,  
Ciência e Tecnologia de Minas Gerais – *Campus* Formiga.

1. Dungeon and Dragons. 2. Python. 3. Java. 4. Webservice. 5. Kivy.  
I. Título.

CDD 004

Samuel Gomes Duque

## Desenvolvimento de um gerenciador inteligente para campanhas de Dungeon and Dragons

Trabalho de Conclusão de Curso apresentado ao  
Instituto Federal de Minas Gerais-Campus  
Formiga, como Requisito parcial para obtenção do  
título de Bacharel em Ciência da Computação.

Aprovado em: 13 de junho de 2019.

BANCA EXAMINADORA



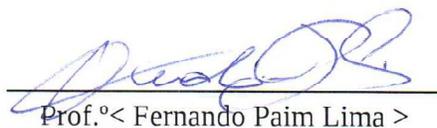
Prof.º <Wallace de Almeida Rodrigues>



Prof.º <Everthon Valadão dos Santos>



Prof.º < Denise Ferreira Garcia Rezende >



Prof.º < Fernando Paim Lima >

*Dedico este trabalho a minha família, por acreditar em mim. Mãe, seu cuidado e apoio foi o que me deu sustento para continuar minha caminhada. Pai, seu cuidado para comigo me impulsionou nas minhas tarefas e valores.*

# Agradecimentos

Primeiramente agradeço a Deus por me capacitar com diversos dons e em específico no desenvolvimento deste trabalho. À Nossa Senhora pela proteção e intercessão durante minha caminhada.

Agradeço aos meus pais Eunice e Enio, aos meus irmãos Lucas e Geovanna pelo apoio nas decisões e todo o amor recebido durante as fases de minha vida, vocês são os pilares que me sustentam. Agradeço também ao meu companheiro Edmilson, que com sua presença e cuidado me impulsionou em minhas decisões e me deu incentivo para que eu me tornasse um ótimo profissional.

Por fim agradeço aos meus professores e a todos os servidores do IFMG - Campus Formiga que se dedicam sempre para oferecer um ensino de qualidade a todos seus alunos. Ao meu orientador, Wallace, e meu coorientador Everthon, os quais aceitaram com carinho o convite para o auxílio neste trabalho e toda ajuda fornecida. Aos membros da banca, Profa. Denise e Prof. Fernando, que se dispuseram a participar deste momento tão importante em minha vida e na contribuição da evolução deste trabalho.

*“O caráter de um homem é sempre o seu destino.” (Demócrito)*

# Resumo

Este trabalho de conclusão de curso tem como objetivo desenvolver um simulador inteligente para campanhas de *Dungeon and Dragons*, o qual auxilia o mestre a realizar o gerenciamento eficiente e eficaz de suas atividades. No estudo para o desenvolvimento deste trabalho, percebe-se a grande possibilidade de aplicá-lo em áreas como educação, definição de estratégias de equipes e desenvolvimento da criatividade do usuário. Um grande estudo realizado sobre o universo dos *RPG's* (jogo de interpretação de papéis), orientou o desenvolvimento da estrutura básica da aplicação, acrescido do conhecimento obtido com o estudo de ferramentas que proporcionam o desenvolvimento rápido e eficiente como: Python, Kivy, JAVA, Webservice e MySQL. A obtenção de um sistema com o propósito de auxiliar o gerenciamento das campanhas é o principal foco deste trabalho, tendo como finalidade: comunicação entre jogadores e atratividade e dinamismo para criar os elementos de uma história. Conclui-se que existe grande mercado para desenvolvimento de jogos atualmente e novidades são esperadas pelos usuários.

**Palavras-chave:** *Dungeon and Dragons*, Python, Kivy, *Webservice*.

# ABSTRACT

This course conclusion work aims to develop a smart simulator for campaigns of Dungeon and Dragons, in which it assists the master to carry out the efficient and effective management of its activities. In the study for the development of this product, we perceive the great possibility of applying it in areas such as education, definition of team strategies and development of the user's creativity. A large study about the universe of RPG's (role-playing game), guided the development of the basic structure of the application, joined the knowledge obtained from the study of tools that provide the development fast and efficient as: Python, Kivy, Java, Webservice and MySQL. Obtaining a product with the purpose of assisting the management of the campaigns is the main focus of this work, having as purpose: communication between players and attractiveness and dynamism to create the elements of a story. It is concluded that there is a large market for game development today and news is expected by users.

**Keywords:** Dungeon and Dragons, Python, Kivy, Webservice.

Figura 1 – Tela Cadastro de Classes	
Figura 2 – Tela Cadastro de Raças	
Figura 3 – Tela Cadastro de Habilidades	
Figura 4 – Tela Cadastro de Perícia	
Figura 5 – Tela Cadastro de Armas	
Figura 6 – Tela Cadastro de Armadura	
Figura 7 – Tela Cadastro de Consumíveis	
Figura 8 – Tela Cadastro de Lojas	
Figura 9 – Tela Cadastro de Personagens	
Figura 10 – Tela Cadastro de Monstros	
Figura 11 – Tela Cria <i>Dungeons</i>	
Figura 12 – Tela Criação de Mapa-Mundi	
Figura 13 – Diagrama Comunicação	
Figura 14 – Diagrama de Casos de Uso	
Figura 15 – Tela Cria Sala	
Figura 16 – Tela Mapa-Múndi	
Figura 17 – Tela Encontro	
Figura 18 – Tela NPC	
Figura 19 – Autômato para Campanhas	
Figura 20 – Tela Campanha	
Figura 21 – Tela aba loja	
Figura 22 – Tela Inventário	
Figura 23 – Tela seleção encontro para batalhas	
Figura 24 – Tela batalha . . . . .	53

Tabela 1	–	Tabela de componentes de <i>Dungeons</i>	
Tabela 2	–	Tabela de componentes de Mapa Múndi	
Tabela 3	–	Tabela de componentes mapa-mundi	
Tabela 4	–	Tabela de componentes mapa-mundi: campanha . . . . .	45

# Lista de abreviaturas

- RPG** Jogo de interpretação
- D&D** Dungeon and Dragons
- MMORPG** Jogo multijogador massivo online
- GPU** Unidade de processamento gráfico
- WEB** World Wide Web
- IHC** Interface Humano-Computador
- CRPG** Jogo de interpretação eletrônico
- LAN** Rede área local
- WAN** Rede de longa distância
- SGB** Sistema Gerenciador de Banco de Dados
- DBA** Administrador de banco de dados
- SDL** *Simple DirectMedia Layer*
- NPC** Personagem não jogável

<b>1</b>	<b>INTRODUÇÃO</b>
1.1	<b>Justificativa</b> . . . . .
1.2	<b>Objetivos</b> . . . . .
1.2.1	Objetivo Geral . . . . .
1.2.2	Objetivos Específicos . . . . .
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>
2.1	<b>Surgimento do <i>RPG</i></b> . . . . .
2.2	<b><i>Dungeon and Dragons</i> aplicados na atualidade</b> . . . . .
2.3	<b>Linguagem de programação Python</b> . . . . .
2.4	<b>Linguagem Kivy</b> . . . . .
2.5	<b>Linguagem de programação Java</b> . . . . .
2.6	<b>Webservices</b> . . . . .
2.7	<b>JGroups</b> . . . . .
2.8	<b>Sistema Gerenciador de Base de Dados MySQL</b> . . . . .
2.8.1	Workbench . . . . .
<b>3</b>	<b>DESENVOLVIMENTO</b>
3.1	<b>Coleta dos dados do Universo <i>D&amp;D</i></b> . . . . .
3.1.1	O livro do Mestre . . . . .
3.1.2	O livro do Jogador . . . . .
3.1.3	O Manual do Monstro . . . . .
3.2	<b>Banco de Dados</b> . . . . .
3.2.1	Alimentação da Base de Dados . . . . .
3.2.2	Cadastro de Classes . . . . .
3.2.3	Cadastro de Raças . . . . .
3.2.4	Cadastro de Habilidades e Ações . . . . .
3.2.5	Cadastro de Perícias . . . . .
3.2.6	Cadastro de Itens . . . . .
3.2.7	Cadastro de Personagens . . . . .
3.2.8	Cadastro de Monstros . . . . .
3.3	<b>Gerenciador de Mapas</b> . . . . .
3.3.1	Codificação do Gerenciador de Mapas . . . . .
3.4	<b>Comunicação entre os jogadores</b> . . . . .
3.5	<b>Executando o Simulador</b> . . . . .
3.5.1	Tela seleção sala . . . . .

3.5.2	Tela mapa-mundi . . . . .
3.5.3	Encontros . . . . .
3.5.4	NPC's . . . . .
3.5.5	Campanha . . . . .
3.5.6	Loja . . . . .
3.5.7	Inventário . . . . .
3.5.8	Batalha . . . . .

**4 CONCLUSÃO**

**5 TRABALHOS FUTUROS**

<b>REFERÊNCIAS . . . . .</b>	<b>56</b>
------------------------------	-----------

# 1 Introdução

Os jogos eletrônicos estão presentes nas vidas de muitas pessoas, tanto do sexo masculino quanto do feminino, atraindo diferentes faixas etárias em todo o mundo. De Newzoo 2017), o Brasil é o 13º país do mundo em vendas no mercado de jogos eletrônicos, com uma média de 63.6 milhões de jogadores que em 2017 movimentaram uma receita de 1,3 bilhões de dólares.

O mercado dos jogos eletrônicos inclui diversos tipos de negócios, seja com a venda direta de jogos, comercializados tanto em mídia física quanto digital – cujos títulos mais famosos são anunciados nas mídias especializadas – seja com a venda de mensalidades e serviços vendidos dentro dos mundos virtuais dos jogos; seja com o comércio de microtransações realizadas dentro dos jogos; dentre outras possibilidades. A empresa NEWZOO 2017) que realiza diversas pesquisas sobre os avanços nas tecnologias e informações do mercado de jogos, publicou uma pesquisa em 2017 mostrando que nesse ano o comércio dos jogos digitais movimentou cerca de 116 bilhões de dólares. Contudo, com os grandes avanços na área, o mercado dos jogos continua sendo muito promissor e há ainda espaço para novos desenvolvedores no setor.

Passando ao largo das empresas que geram grandes lucros nas produções de jogos,<sup>1</sup> e afins, também existem benefícios para os usuários que utilizam tais ferramentas e serviços: além da realidade associada ao lúdico, a oportunidade para trabalhar a mente humana, desenvolvendo o raciocínio e melhorando a coordenação motora, dentre outras vantagens. Com a existência de jogos que permitem a comunicação entre mais de uma pessoa sem a necessidade de estarem em um mesmo local, foram desenvolvidos diversas formas de <sup>2</sup>, via voz e vídeo. Os jogos na atualidade tiveram um grande avanço tecnológico, absorvendo influência cultural e modificando a maneira como são apresentados ao jogador. Hoje os jogos digitais são uma realidade presente no cotidiano de muitas pessoas.

No decorrer do tempo, com a evolução dos jogos digitais foram surgindo diferentes estilos de jogos, como os de gênero de ação, estratégia, aventura, plataforma e vários outros. Um estilo que fez grande sucesso nos anos 90 e que ainda é muito presente nos dias atuais é o *Role-Playing Game*. O significado da sigla inglesa e o estilo remetem para um jogo de interpretação de personagens: jogo que reúne um grupo de pessoas para desenvolver uma história, seja seguindo um enredo pré-determinado ou criado no improviso, onde uma pessoa é o mestre (narrador) e as demais são personagens da história. Este estilo de jogo é diferente dos jogos convencionais, pois todo o desenrolar da história é dinâmico

---

<sup>1</sup> Microcomputador dedicado a executar jogos de eletrônicos de diversas naturezas.

<sup>2</sup> Aplicações de conversação em tempo real.

Grando e Tarouco 2008), os RPG's surgiram na década de 1970, inicialmente com jogos de tabuleiros e dados multifacetados e coloridos. No ano de 1974 surgiu o RPG *Dungeons and Dragons*, (GYGAX 1984), que servirá de base para o desenvolvimento desse trabalho de conclusão de curso.

Em uma partida de RPG participam dois tipos de jogadores: o mestre (narrador) e os personagens. O mestre tem como objetivo desenvolver uma história, criando um enredo que permite que os jogadores desenvolvam diversas ações, e dependendo das escolhas dos jogadores a história irá tomar um rumo diferente. Cada partida realizada pelo grupo é denominada uma “aventura”, e quando uma aventura se estende e se conecta em outras, com os mesmos personagens e com a sucessão da história, estas juntas formam uma “campanha”. Cada aventura possui um conjunto de regras próprias e a cada escolha de um jogador o mestre responderá com novas instruções que oferecem novas escolhas aos jogadores. Em uma situação onde as ações de um jogador podem resultar em falha, o narrador pode solicitar o uso de rolagem de dados para determinar se o personagem conseguirá ou não executar a ação desejada. O gênero de jogo RPG demanda tempo para ser jogado, pois é necessário definir o universo e as regras que compõem o ambiente onde vai se desenrolar a história, além das criações dos personagens e a montagem do enredo.

Toda a complexidade de um RPG pode confundir jogadores novatos e a computação pode ser uma ferramenta muito útil para auxiliar no desenvolvimento e gerenciamento desses jogos. Este trabalho de conclusão de curso tem por objetivo investigar o uso da computação nos jogos e desenvolver uma ferramenta para auxiliar o mestre no gerenciamento de uma campanha de *RPG*.

Existem diversos manuais descrevendo universos já criados para serem utilizados nas partidas, como *Forgotten Realms* (FANNON 1999); *Greyhawk*, (GYGAX 1984) e vários outros. Neste projeto utilizamos como base os livros: *Player's Handbook* (TEAM 2014b), *Monster Manual* (TEAM 2014c) e *Dungeon Master's Guide* (TEAM 2014a).

## 1.1 Justificativa

Os jogos RPG também vem sendo empregados em cursos de computação para o ensino de diversas disciplinas necessárias para a sua construção, pois favorecem a (SCHNEIDER 2015). Este gênero de jogo permite desenvolver histórias de maneira lúdica e também pode ser utilizado para abordar fatos históricos, permitindo sua aplicação nas escolas para trabalhar matérias (Grando e Tarouco 2008) afirma que os jogos de RPG possuem características que os tornam excelentes ferramentas educacionais, facilitando: socialização,

cooperação, criatividade, interatividade e interdisciplinaridade. O RPG tem um grande potencial para o ensino, pois permite sua utilização por professores no cotidiano das aulas.

(SCHNEIDER 2015), utilizou o RPG para criar um jogo educacional digital para o ensino dos conceitos do Scrum, uma metodologia ágil de desenvolvimento, assim é possível perceber as áreas que este jogo abrange, tanto em matérias de ensino fundamental, como geografia, história, português e outras como em matérias de curso superior, principalmente na área da computação. Um projeto desenvolvido

(Oliveira, Pierson e Zuin 2009) aplicou as características do RPG para avaliar a aprendizagem no ensino de química, neste projeto foram descritos diversas narrativas que apresentavam um problema para os alunos, assim permitia que eles desenvolvem-se ações para investigar as pistas informadas em busca de uma solução. O resultado gerado pelo (Schneider 2015) foi caracterizado de “grande pertinência”, pois os alunos demonstraram ganho de conhecimento ao decorrer do jogo.

Várias características dos jogos estilo RPG estão presentes também em sistemas que simulam um ambiente real e têm como objetivo treinamento, análise e previsão de (Jones 2013) este estilo de jogo exige o controle de um personagem cujas ações podem ou não impactar o resultado do jogo. Tarefas de simulação também estão presentes em jogos do estilo *War Games*, precursor dos jogos de RPG, jogos de negócios e outras variantes de RPG.

Com o grande avanço das tecnologias os jogos também tiveram uma transformação, os primeiros jogos criados permitiam a visualização de objetos e personagens de maneira mais simples, visualizando seus *pixels* na tela. Com o grande avanço da computação gráfica, os jogos atualmente buscam se aproximar da realidade o máximo possível, uma área bastante procurada não somente para desenvolvedores de jogos, mas também para desenvolvedores de animações e filmes. A computação gráfica teve seu sentido explicado (COMPUTER 1998).

No mercado competitivo dos jogos inovar é uma necessidade para o sucesso e os jogos buscam apresentar mecanismos diferenciados dos demais, não somente pelos gráficos, história, mas pela inteligência implementada no jogo. Desenvolver ações diferentes conforme o usuário avança no jogo é uma característica cada vez mais presente em jogos e principalmente nos que possuem o estilo RPG. Criar uma inteligência para uma máquina que permite analisar e aprender sobre o seres humanos não é trivial, pois o ser humano é imprevisível. O desenvolvimento de jogos abordam diversas áreas da computação, como: inteligência artificial, que permite a elaboração de uma ferramenta inteligente para controle do jogo; banco de dados, para controle dos dados inseridos e coletados conforme o decorrer do jogo; redes, para realizar a comunicação entre os diversos jogadores na rede; lógica, para controlar as diferentes ações que podem ocorrer; linguagens formais e autômatos, que possibilitam desenvolver uma máquina de estados conforme cada ação do jogador

é executada; interface humano-computador, para criação de interfaces interativas e que chamem atenção dos jogadores e várias outras disciplinas.

## 1.2 Objetivos

Esta seção descreve os objetivos desse trabalho de conclusão de curso.

### 1.2.1 Objetivo Geral

O projeto tem como objetivo geral desenvolver um sistema para gerenciar partidas *online* de RPG de mesa, dotado com automatização para auxiliar o mestre no gerenciamento de cada partida, além de permitir a comunicação entre os jogadores na rede com boa usabilidade para o desenrolar das aventuras.

### 1.2.2 Objetivos Específicos

Os objetivos específicos são apresentados nos seguintes itens:

- Estudar os *frameworks* disponíveis para o desenvolvimento das interfaces e selecionar aquele que será utilizado no projeto.
- Estudar os *frameworks* disponíveis de redes e sistemas distribuídos para desenvolvimento das aplicações cliente-servidor.
- Estudar inteligência artificial para o desenvolvimento de um módulo inteligente para auxiliar o mestre em seu trabalho.
- Estudar o jogo RPG, universo *Dungeon and Dragons*, para o entendimento das regras, funcionamento do jogo e possíveis ações do mestre.
- Criação e alimentação do banco de dados com as informações coletadas dos manuais do jogo.
- Desenvolvimento de uma interface.
- Estudo e desenvolvimento de geradores de mapas.
- Implementação das regras do jogo.
- Construir uma aplicação para o cliente e servidor (jogadores e o mestre respectivamente).

## 2 Fundamentação Teórica

### 2.1 Surgimento do *RPG*

O *Dungeon And Dragons* tem uma vasta história de desenvolvimento, em seus 40 anos de divulgação e existência, muitas transformações já ocorreram e o site oficial do (PETERSON 2017) enumera algumas importantes.

- **1971:**

1. *Chainmail*, escrito por Gygax e Jeff Perren, é publicado pela *Guidon Games*. Ele detalha a fantástica guerra medieval em miniatura, incluindo magos, heróis e dragões.
2. Arneson usa *Chainmail* como regras para sua campanha de fantasia *Blackmoor* situada no Grande Reino, incluindo incursões em uma masmorra de monstros e tesouros.

- **1973:**

1. Gygax e Arneson colaboram em rascunhos de *Dungeons And Dragons*.
2. Gygax e Kaye formam uma parceria chamada Regras de Estudos Táticos. Brian Blume junta-se às Regras de Estudos Táticos para ajudar a produzir o jogo *Dungeons And Dragons*.

- **1974:**

1. *Dungeons And Dragons* é publicado pela primeira vez em janeiro como três livretos enviados em uma caixa de papelão com textura de madeira: *Men & Magic*, *Monsters and Treasure* e *Underworld and Wilderness Adventures*. A tiragem de mil jogos esgotaram em onze meses.

- **1977:**

1. *Dungeons And Dragons* é dividido em versões Básica e Avançada. O Conjunto Básico de D&D editado por J. Eric Holmes se torna o primeiro conjunto de *Dungeons And Dragons* a ser lançado com dados incluídos.
2. A TSR Hobbies publica o Livro dos Monstros, o primeiro livro da linha de produtos *Advanced Dungeons And Dragons*. Seus 350 monstros exercem uma enorme influência sobre os jogos de fantasia subsequentes, tanto na mesa quanto nos computadores.

- **1978:**

1. *The Players Handbook*, o segundo livro de capa dura *Advanced Dungeons And Dragons* é lançado.

- **1979:**

1. A TSR Hobbies lança o *Dungeon Masters Guide*, o terceiro e último livro de regras de capa dura *Advanced Dungeons And Dragons*.

- **1980:**

1. Cerca de 250.000 unidades de produtos *Dungeons And Dragons* foram vendidas antes do início do ano. No verão, as vendas do conjunto básico atingem 12.000 cópias por mês.
2. O lançamento inicial da pasta *World of Greyhawk* marca o primeiro cenário de campanha produzido pela TSR, fornecendo um contexto para a ação em módulos que remonta ao original Grande Reino da *Castle & Crusade Society*.

- **1981:**

1. A *Role Playing Game Association* (RPGA) é formada para promover *role playing* de qualidade e unir os jogadores em todo o país, auxiliado por sua principal revista Polyhedron.

- **1982:**

1. A TSR lança seus primeiros produtos de *videogame*, incluindo uma versão do *Dungeon!* jogo de tabuleiro para o sistema de computador Apple II.

- **1988:**

1. A *Strategic Simulations Incorporated* (SSI) lança o jogo de computador pessoal *Pool of Radiance*, a primeira adaptação de computador licenciada do sistema *AD & D*, e o primeiro jogo de computador na configuração da campanha *Forgotten Realms*. Um número de jogos “Gold Box” baseados no mesmo motor *AD & D* seguem.

- **1991:**

1. O jogo RPG multijogador *online Neverwinter Nights* começa na rede AOL, em colaboração com SSI, Stormfront e TSR.

Após diversos anos grandes modificações foram realizadas e até os dias atuais novas atualizações vem sendo feitas. Diversos jogos aproveitam grande parte deste universo e suas evoluções, buscando trazer o máximo possível de realidade ao mundo dos jogos eletrônicos, este assunto será tratado no tópico a seguir.

## 2.2 *Dungeon and Dragons* aplicados na atualidade

Um termo criado para definir sobre os *RPGs* aplicados em computadores são os *CRPGs* ("*Computer Role Play Game*"). Definir o *CRPG* como precursor de um novo estilo de *RPG*, entretanto, não é algo certo a se dizer. Mesmo com pesquisas a fundo sobre o assunto não é possível afirmar qual jogo foi o ponto inicial para o desenvolvimento dos demais, pois cada jogo possui sua peculiaridade, tornando difícil classificá-los pelo (Marques e Ceoni 2007), os primeiros jogos desenvolvidos eram bastante limitados pela tecnologia da época. O ano de 1970 marcou o início da criação dos primeiros *CRPG*, com o desenvolvimento de animações 3D para fins didáticos, publicitários e do interesse da indústria cinematográfica nos avanços da computação. Esses eventos tornaram-se pontos fortes para os desenvolvimentos de jogos de computadores, que utilizam uma grande variedade de elementos visuais para prender a atenção de diferentes tipos de jogadores.

Durante muito tempo os *CRPGs* foram considerados *RPG*, apesar de apresentarem uma proposta um pouco diferente. A limitação da personalização do universo, como personagens, monstros e os demais elementos é maior, possibilitando ao usuário definir propriedades pré-definidas. Além da criação, toda a história que se segue, o jogador possui uma lista de ações para cada acontecimento e de acordo com a opção escolhida alguma reação acontece, trazendo a impressão ao jogador que ele pode tomar qualquer tipo de decisão. Um grande diferencial dos *CRPGs* sobre os *RPGs* é a possibilidade de realizar uma partida solo, algo que pode afastar ou não amantes do gênero. Muitos jogos desenvolvidos atualmente não possuem formas que permitem a interação de mais de um jogador, como reação contra isso foram desenvolvidos os *MMORPGs*.

Os *MMORPGs* ("Jogo multijogador massivo *online*") permitem que diferentes jogadores de diversas regiões se conectem ao mesmo universo, agregando novas características aos *RPGs* clássicos.

Com o avanço da computação, aumento da capacidade de processamento e renderização de imagens diferentes, muitos novos jogos surgiram com inovações. Um exemplo famoso que revolucionou o cenário de jogos *CRPG* é o jogo *The Witcher*, que permite uma personalização detalhada do personagem principal, além de possuir um vasto mundo com vegetações diferentes, climas, vasta biodiversidade e muitas outras novidades cativantes. Considerado por muitos críticos um dos melhores jogos de todos os tempos, aproxima-se

muito bem do que um *RPG* é capaz de proporcionar, mesmo não tendo a liberdade máxima (mas algo bem próximo).

## 2.3 Linguagem de programação Python

O jogo *Dungeon and Dragons* é um jogo dinâmico, tendo como grande foco a ambientação que permite o jogador se sinta imerso em seu universo. A linguagem Kivy juntamente com o Python permite que sejam desenvolvidos elementos gráficos eficientes para abordar o funcionamento do jogo.

O Python é uma linguagem de programação muito utilizada, aborda diversos paradigmas de linguagens, um dos fatores que a torna perfeita para ser utilizada em sistemas grandes. Permite também a interação com outras linguagens caso haja necessidade de realização de alguma tarefa. Existem diversos pacotes que facilitam a interação e o comportamento do sistema, sua facilidade em execução de *threads*, acionamentos de funções, estruturas de dados que realizam o armazenamento efetivo de diferentes tipos de

Borges 2014) afirma que o Python fornece legibilidade ao código-fonte, permitindo que o usuário desenvolva mais rápido suas aplicações. Oferece estruturas de alto nível que permitem controlar diversos tipos de operações e dados como: listas, dicionários, números complexos e outros. Uma das características dessa linguagem é o fato de possibilitar a inserção no projeto de módulos desenvolvidos por outros programadores além daqueles já disponibilizados pela linguagem. É uma linguagem muito utilizada como *script* em várias aplicações, automatizando tarefas sendo possível a integração com outros tipos de *softwares*.

## 2.4 Linguagem Kivy

O Kivy é uma plataforma, com uma linguagem própria, que oferece suporte gráfico para o Python. Apesar de ter surgido atualmente, vem sendo aprimorada para fornecer (Kivy 2011) os fatores do grande uso e sucesso da linguagem são: usar algoritmos inteligentes para minimizar operações dispendiosas; e utilizar a GPU sempre que ela fizer sentido no contexto. Além de ser uma linguagem altamente produtiva, ela é gratuita e com uma grande comunidade ativa.

O ambiente necessário para posicionamento de *tokens* como monstros, personagens, armadilhas e que realizam as batalhas é de grande importância, pois permite que o sistema de batalha seja implementado de acordo com o posicionamento dos personagens, além de permitir toda a interação com o mundo e com os objetos na tela. Com o uso desta

ferramenta, diferentes telas podem ser criadas e com as várias funcionalidades impostas pela mesma, um gerenciador de mapas foi criado, permitindo que diferentes mapas sejam criados aos usuários e estes carregados para as partidas, dispensando assim a não necessidade de uso de alguma ferramenta exterior.

## 2.5 Linguagem de programação Java

Java é considerada uma linguagem poderosa e portátil que implementa elegantemente o paradigma da orientação a objetos, permitindo o reuso e manutenção rápida e (INDRUSIAK 1996). Outra vantagem que ela oferece é a segurança que possibilita diminuir a quantidade de erros possíveis de programação, como exemplos: não é possível acessar informações diretamente da memória, quebrar o encapsulamento dos objetos ou violar o controle da tipagem.

## 2.6 Webservices

(Magri 2013), um *Webservice* é um sistema que oferece serviços para outros sistemas: um conjunto de métodos desenvolvidos para atender um objetivo comum, que é armazenado em local com acesso não necessariamente físico, para realizar operações específicas de forma rápida e eficiente. A capacidade de ser um ambiente padronizado sem vínculo com uma linguagem específica é um de seus pontos fortes e com isto permite que uma aplicação em Java consuma um *Webservice* criado por uma aplicação em *Python* e vice-versa. A forma que os valores são retornados aos consumidores é diversificada, proporcionando ao usuário uma ampla capacidade para realizar operações nas informações.

Para atender a comunicação entre os jogadores nesse trabalho, um *Webservice* foi desenvolvido para permitir a integração do Java com o Python, uma vez que o módulo encarregado de implementar a comunicação entre os usuários foi implementado em Java utilizando o pacote de *JGroups*.

## 2.7 JGroups

Para um sistema onde é necessário suportar mais de um cliente simultaneamente, é preciso que seja desenvolvida uma forma de comunicação que consiga suportar a quantidade de clientes conectados, a confiabilidade no transporte das informações entre todos, para que nenhum usuário fique desatualizado perante os demais. Um sistema distribuído necessita uma validação rígida, se aplicado em um sistema que não seja tolerante a falhas – qualquer informação errada ou em falta pode prejudicar os jogadores causando problemas

irreversíveis. O JGroups é um *kit* de ferramentas em Java para realizar a comunicação distribuída entre diversos clientes em um mesmo *cluster*.

Algumas características do JGroups:

“Criação e exclusão de *cluster*; nós de *cluster* podem ser distribuídos por LANs ou WANs; juntando e partindo de agrupamentos; detecção de membros e notificação sobre nós do *cluster* conectados, desconectados e com falhas; detecção e remoção de nós com falha; envio e recebimento de mensagens de nó para *cluster* (ponto a multiponto); envio e recebimento  
JGROUPS 2002)

A pilha de protocolos do JGroups é bastante flexível e permite adaptar cada protocolo conforme as necessidades da aplicação, bem como utilizar apenas os protocolos mais adequados para os requisitos do problema

## 2.8 Sistema Gerenciador de Base de Dados MySQL

Além do controle de tráfego dos dados entre os usuários, é necessário uma forma destes dados serem armazenados e difundidos. Nesse trabalho foram utilizadas as ferramentas MySQL e Workbench para desenvolver a base de dados utilizada para armazenar os dados presentes do jogo.

“O MySQL é o banco de dados de código aberto mais conhecido no mundo. Com comprovado desempenho, confiabilidade e facilidade de uso, o MySQL tornou-se a principal opção de banco de dados para aplicativos  
MYSQL 2019).

MySQL 2019), o MySQL é considerada a base de dados em código aberto número um em quantidade de usuários, implementando funcionalidades para aplicações web e sistemas embarcados, sendo uma ferramenta presente em diversos sites conhecidos e utilizados mundialmente como: Facebook, Youtube e Twitter.

Milani 2007), o MySQL Server contém todas as características presentes de um SGBD (Sistema Gerenciador de Banco de Dados). Além das características já mencionadas, ele possui outras como: gerenciamento de acesso, integridade dos dados, transações e diversas outras, as quais o tornam uma ferramenta poderosa para gerenciamento de muitos dados. É uma aplicação de alto grau de portabilidade, sendo assim possível o uso em linguagens como: Python e Java, as linguagens que são utilizadas

Milani 2007) afirma também que o MySQL tem enorme velocidade de processamento, pelo uso de muitas *threads* facilitando a integração com computadores com mais de um processador. É possível definir no banco de dados criado, tabelas com fins específicos para armazenar uma massa maior de dados, ou priorizar consultas com o

Milani 2007) confirma que é uma ferramenta com alta segurança, permitindo habilitar um *firewall* específico ou uma criptografia para as senhas.

### 2.8.1 Workbench

O MySQL Workbench funciona como uma alternativa visual que permite a criação de banco de dados de forma interativa, assim o usuário consegue realizar a criação de tabelas possuindo uma melhor organização e praticidade. Uma vez criadas as tabelas, a ferramenta permite ao usuário gerenciá-las, realizando conversão de dados, criação de dicionário de dados e muitas outras funcionalidades, além daquelas implementadas como *plugins*.

“O MySQL Workbench é uma ferramenta visual unificada para arquitetos de banco de dados, desenvolvedores e DBAs. Fornece modelagem de dados, desenvolvimento de SQL e ferramentas abrangentes de administração para configuração de servidores, administração de usuários, *backup* e  
WORKBENCH 2019)

## 3 Desenvolvimento

Para o desenvolvimento deste projeto foi feito um estudo do universo de *RPG*, com suas regras e procedimentos mais comuns do mestre e dos jogadores. Também foi realizado o estudo das ferramentas utilizadas para atender os requisitos necessários para a execução de uma partida. A ideia inicial era utilizar a linguagem Python, pois suas características permitem uma construção eficaz, rápida, além de disponibilizar diversas bibliotecas que facilitam seu uso. Uma base de dados foi implementada e alimentada com informações, coletadas do estudo dos manuais do jogo, necessárias para auxiliar a execução da partida. Para prover boa usabilidade, foram desenvolvidas interfaces para a interação usuário-computador. A utilização dos métodos de Interface Humano-Computador (IHC)

Alves e Padovani (2006): “os modelos e métodos de avaliação ergonômica tradicionalmente em uso mostram-se pouco eficazes no desenvolvimento e avaliação de jogos digitais”. Assim, para analisar um jogo digital é necessário seguir outras etapas visando a satisfação do usuário, e algumas verificações podem ser realizadas para obter um bom resultado como:

- Evitar a exaustão do jogador.
- Obter uma inteligência artificial que agrada o jogador, tornando o jogo mais dinâmico.
- Desenvolver um jogo divertido aos jogadores.

Toda a comunicação dos jogadores foi realizada por meio de *Webservice*, o qual está presente em todos os computadores, funcionando como interface entre a aplicação em Python (interface gráfica e regras de negócios) e Java (comunicação entre nós distribuídos no *cluster*) utilizando *JGroups*. Todas estas etapas do desenvolvimento seguiram a forma incremental, sendo cada módulo responsável por uma tarefa.

### 3.1 Coleta dos dados do Universo *D&D*

A primeira etapa foi o pilar para o desenvolvimento da aplicação. As informações relevantes identificadas, coletadas, analisadas e adaptadas para atender o sistema. Nessa etapa foram consultados três livros essenciais que descrevem todos os dados relacionados à criação do universo de *Dungeon and Dragons*.

### 3.1.1 O livro do Mestre

(Team 2014a), contém diversas informações sobre o ambiente, configuração de batalha, definição de eventos e missões.

A definição da campanha se dá pelo detalhamento da história e os eventos que a precedem, pela definição de quando e como cada evento ocorre e as ligações que permitem que o jogador se envolva com o universo criado pelo mestre. Diversas configurações são importantes para auxiliar o mestre: quais dados e a forma como são jogados durante os eventos, como inicia-se uma batalha, as ações possíveis dos jogadores e monstros, etc.

### 3.1.2 O livro do Jogador

Definir os personagens é algo que exige criatividade, grande conhecimento sobre as classes e raças, e o que mais causa impacto durante a criação: o tempo.

Criar um personagem é o momento mais demorado na pré-criação de uma campanha. Cada personagem é único e com uma ampla quantidade de características possíveis. Cada um possui raça, classe, equipamentos e habilidades em seus diversos níveis. Com o uso do (TEAM 2014b), foram inseridos os elementos essenciais que permitem tornar o processo de criação mais rápido.

### 3.1.3 O Manual do Monstro

Além dos personagens, do mestre e o que ele gerencia, existem também outros elementos muito importantes no universo de D&D, a saber, os monstros.

(TEAM 2014c) traz diversas informações sobre os seres existentes no mundo de *Dungeon and Dragons*. Tais seres são controlados pelo mestre, que escolhe o momento em que eles aparecem para os jogadores. Como os personagens, os monstros são seres bastantes complexos, com características únicas que precisam de grande atenção. Além dos seres existentes no livro, que facilitam a criação da campanha, nada impede que os jogadores desenvolvam seus próprios monstros.

## 3.2 Banco de Dados

Toda base de dados foi desenvolvida a partir do estudo dos livros: foram ao todo 42 tabelas criadas com atributos específicos que se aproximam o mais próximo do *RPG Dungeons And Dragons*. Os dados iniciais estão presentes no banco de dados do mestre e logo após a conexão de outros jogadores, esses dados são difundidos entre os jogadores na medida que o mestre for liberando o acesso. O acesso para o modelo lógico do banco de dados está disponível em: <https://github.com/SamuelGDuque/bancoDeDadosRPG>

### 3.2.1 Alimentação da Base de Dados

Cada tela desenvolvida nesta etapa alimenta o banco de dados do mestre, cadastrando monstros, habilidades, classes, raças, equipamentos e vários outros elementos. Com a base de dados alimentada, realizar qualquer ação durante o jogo torna-se mais simples e rápido, pois não é necessário realizar todos os cálculos, verificar todas as fichas e etc. Tudo se torna mais rápido com a automatização das consultas, trazendo mais praticidade aos jogadores e principalmente ao mestre.

(DUSTMAN 2012) que permite realizar operações com o MySQL de forma bastante intuitiva. Assim, durante a execução em tempo real do jogo podem ser realizadas consultas, inserções e todo o gerenciamento de dados no banco.

### 3.2.2 Cadastro de Classes

A tela de Cadastro de Classes permite ao usuário criar classes e foi inspirada nas classes já existentes no universo *D&D*, entretanto, o usuário também pode criar sua própria.

1 representa a tela de criação de uma classe. Os campos apresentados nesta tela estão presentes na base de dados, sendo que cada um contém as seguintes características:

#### 1. Dados principais:

- Nome: Nome dado a Classe
- Vida: Quantidade de vida que a classe possui como base

#### 2. Nível de resistência da classe aos atributos dos ataques dos inimigos.

- FOR: Resistência ao atributo força
- DES: Resistência ao atributo destreza
- INT: Resistência ao atributo inteligência
- SAB: Resistência ao atributo sabedoria
- CAR: Resistência ao atributo carisma
- CON: Resistência ao atributo constituição

### 3. Equipamentos que a classe poderá utilizar

- Tipos de Armas: Espadas, arcos, adagas e cajados
- Tipos de Armadura: Armadura leve, média ou pesada

Figura 1 – Tela Cadastro de Classes

Simulador Ultros

Cadastro Classe    Cadastro Habilidade    Cadastro Perícia

Cadastro de Classes

Nome:

Vida:

Teste de Resistência

FOR     DES     INT     SAB     CAR     CON

Equipamentos

Tipos Armas:

Tipos Armadura:

Espada     Armadura Leve   
Arco     Armadura Média   
Adaga     Armadura Pesada   
Cajado

Cadastrar    Voltar

Fonte: Elaborado pelo autor.

#### 3.2.3 Cadastro de Raças

Existem diversas raças, com variações que dependem da espécie ou do ambiente familiar ao personagem (exemplo: elfo negro, elfo da floresta, elfo da planície, etc). Cada raça define alguns atributos especiais que, juntamente com a classe, podem criar personagens

2 representa a tela de cadastro de raças e os seus respectivos campos.

1. **Nome:** Nome dado a raça
2. **Deslocamento:** Quantidade de movimento que a raça permite no turno.
3. **Descrição:** Características da raça e informações relevantes para desenvolver uma história.

Figura 2 – Tela Cadastro de Raças



The image shows a software window titled "Simulador Ultros" with a sub-header "Cadastro de Raças". The interface is set against a parchment-textured background. It features three text input fields: "Nome:", "Deslocamento:", and "Descrição:". At the bottom, there is a dark grey bar containing two buttons: "Cadastrar" and "Voltar".

Fonte: Elaborado pelo autor.

### 3.2.4 Cadastro de Habilidades e Ações

A tela de habilidades e a tela de ações permitem a criação de qualquer habilidade utilizada pelos personagens e monstros respectivamente. Cada habilidade ou ação possui efeitos diferentes e pré-requisitos para serem utilizadas. As habilidades são usadas durante a tela de habilidades e também os seus campos.

1. **Nome:** Nome da Habilidade
2. **Descrição:** Descrição da habilidade
3. **Alcance:** Alcance mínimo entre o personagem e o alvo para que a habilidade possa acertar
4. **Acerto:** Chance de acerto ao alvo (dados)
5. **Dano:** Valor que será retirado da vida do alvo
6. **Conjuração:** Tempo necessário (em turnos) para executar a habilidade
7. **Penalidades:**

- **Cegueira:** Se aplicado ao alvo, diminui a taxa de acerto do mesmo
  - **Sangramento:** Alvo perderá um pouco de vida por turno
  - **Debilitação Movimentação:** Diminui a taxa de esquiva do alvo
  - **Debilitação de Ataque:** Diminui o dano que o alvo causa aos outros jogadores
  - **Envenenamento:** Alvo perderá um pouco de vida por turno
8. **Tipo Dano:** Tipo de dano causado ao usuário, diferencia a forma que o ataque é realizado
9. **Tipo Mágico:** Permite inserir dano mágico ao ataque, aplicando efeitos diversos de acordo com o elemento selecionado

Figura 3 – Tela Cadastro de Habilidades

Fonte: Elaborado pelo autor.

### 3.2.5 Cadastro de Perícias

Uma perícia é aplicada a classe do personagem, permitindo que o mesmo tenha alguma habilidade diferente dos demais. Nem todas as perícias podem ser utilizadas durante batalhas, algumas podem ser habilidades que são utilizadas na exploração do mundo. A Figura 4 representa a tela de cadastro de perícia e os campos existentes.

1. **Nome:** Nome da perícia
2. **Descrição:** Descrição da perícia (características e efeitos que causam ao mundo ou alvo)

Figura 4 – Tela Cadastro de Perícia



A imagem mostra uma janela de software intitulada "Simulador Ultros" com o título "Cadastro de Perícia". O fundo da janela tem uma textura de papel amarelado. Há dois campos de texto: "Nome" e "Descrição". Na base da janela, há dois botões: "Cadastrar" e "Voltar".

Fonte: Elaborado pelo autor.

### 3.2.6 Cadastro de Itens

Nesta tela pode-se criar diversos itens e equipamentos, também é possível definir em quais lojas tais objetos criados podem ser vendidos para os jogadores. Na criação dos três tipos de itens, alguns campos possuem a mesma finalidade, estes são:

1. **Nome:** Define o nome do objeto
2. **Descrição:** Define ao objeto suas características
3. **Valor do Item:** Valor de ouro, prata e bronze (dinheiro do universo criado)
4. **Peso:** Peso do item (peso que ocupará do inventário)

Na aba de criação de armas e armaduras, temos o campo denominado “tipo” que permite designar se os equipamentos são: leves, médios ou pesados.

Para criar uma arma é necessário que seja definido a distância máxima entre o alvo e o personagem, desta forma calcula-se a chance de acerto; define também se o equipamento precisa de munição; o campo designado *rank* baseia-se na escolha da qualidade do equipamento; na opção “dano” determina o valor do ataque causado ao alvo; e por fim a "acuidade" que evidencia a capacidade e agilidade de cada arma.

Na criação de uma armadura é necessário informar o CA (capacidade de defesa que a armadura proporciona ao personagem) e os modificadores de destreza e força (bônus

5 6, representam as telas de cadastro de armas e armaduras respectivamente.

Figura 5 – Tela Cadastro de Armas

The screenshot shows a software interface titled "Simulador Ultros" with a navigation menu containing "Cadastro Armas", "Cadastro de Armaduras", "Cadastro de Itens", and "Lojas". The active screen is "Cadastro de Armas", which has a parchment-style background. It contains the following elements:

- Fields for "Nome:" and "Dano:".
- A large text area for "Descrição:".
- A "Tipo:" dropdown menu with "Tipo Arma" selected, and a "Rank:" field.
- Fields for "Peso:" and "Distância:".
- Buttons for "Acuidade:" (Sim, Não) and "Munição:" (Sim, Não).
- Fields for "Valor do item:" with sub-fields for "Ouro:", "Prata:", and "Bronze:".
- Two large buttons at the bottom: "Cadastrar" and "Voltar".

Fonte: Elaborado pelo autor.

Figura 6 – Tela Cadastro de Armadura

Simulador Ultros

Cadastro Armas   Cadastro de Armaduras   Cadastro de Itens   Lojas

Cadastro de Armadura

Nome:  CA:

Descrição:

Tipo:  Peso:

Modificador Des:

Modificador For:

Valor do item:

Ouro:  Prata:  Bronze:

Cadastrar   Voltar

Fonte: Elaborado pelo autor.

Do contrário dos equipamentos mencionados anteriormente, a tela de consumíveis permite criar itens que podem ser utilizados como forma de suporte ao jogador, como poções e outros. Há o campo de efeito, que define a consequência que será causada ao

7 representa a tela de cadastro de consumíveis.

Figura 7 – Tela Cadastro de Consumíveis

The image shows a software interface for 'Simulador Ultros'. At the top, there is a navigation bar with four tabs: 'Cadastro Armas', 'Cadastro de Armaduras', 'Cadastro de Itens', and 'Lojas'. The 'Cadastro de Itens' tab is currently selected. Below the navigation bar, the main content area has a parchment-like background and is titled 'Cadastro de Consumíveis'. The form includes the following fields:

- Nome:** A single-line text input field.
- Peso:** A single-line text input field.
- Descrição:** A large multi-line text area.
- Efeito:** A single-line text input field.
- Valor do item:** Three separate single-line text input fields labeled 'Ouro', 'Prata', and 'Bronze'.

At the bottom of the form, there are two large buttons: 'Cadastrar' on the left and 'Voltar' on the right.

Fonte: Elaborado pelo autor.

Na tela corrente, a aba de opção “Lojas” proporciona o jogador a seleção dos itens criados anteriormente e permite os designar as suas respectivas lojas no universo. Para isto, o mestre deve primeiro criar as lojas ou selecionar uma previamente criada e atribuir 8 representa a tela de cadastro de lojas.

Figura 8 – Tela Cadastro de Lojas



Fonte: Elaborado pelo autor.

### 3.2.7 Cadastro de Personagens

A tela de cadastro de personagens é onde se encontra a maior parte dos dados, pois, são umas das informações mais importantes da campanha. É nesta tela que o jogador realiza a criação de seu personagem e seleciona as raças e classes existentes, compondo sua ficha. Todos os personagens criados nesta tela, são inseridos ao banco de dados e armazenados em um arquivo local caso queira utilizar o personagem em uma outra campanha posteriormente. A lista a seguir informa os campos presentes em uma ficha de um personagem.

1. **Nome:** Nome do personagem;
2. **Classe:** Escolha da classe do personagem;
3. **Raça:** Escolha da raça do personagem;
4. **Imagem do Personagem:** Permite o usuário escolher uma imagem para o personagem;
5. **Atributos**
  - **Vida:** Vida do personagem (ao chegar a 0 em batalha, o personagem é derrotado e retirado da mesma;)

- **Deslocamento:** Quantidade de movimentação que um personagem pode fazer;
  - **Armadura:** Defesa do jogador contra ataques inimigos;
  - **Teste Resistência:** Quais as resistências do personagem;
  - **Imunidade a Dano:** Qual tipo de dano o personagem é imune (cortante, perfurante...);
  - **Imunidade a Condição:** Tipo de imunidade a feitos como (sangramento, envenenamento e outros);
  - **Sentido:** Bônus em algumas áreas (como visibilidade no escuro, florestas e outros) ;
  - **Idioma:** Linguagens que o personagem tem domínio;
6. **Atributos:** Quantidade de cada atributo inicial do personagem (Força, Inteligência, Destreza, Sabedoria, Constituição e Carisma)

9 apresenta a tela de personagens.

Figura 9 – Tela Cadastro de Personagens

A interface de usuário para o 'Simulador Ultros' apresenta a tela de 'Cadastro de Personagem'. O formulário é dividido em seções: 'Nome' com um campo de texto; 'Classe' e 'Raça' com botões de seleção; 'Imagem Personagem' com uma pré-visualização de um personagem em silhueta; e 'Cadastro Atributos' com uma lista de campos para: Vida, Deslocamento, Armadura, Teste Resistencia, Resistência Dano, Imunidade Dano, Imunidade Condição, Sentido, Idioma, e atributos FOR, INT, DES, SAB, CON, CAR. Na base, há botões 'Cadastrar' e 'Voltar'.

Fonte: Elaborado pelo autor.

### 3.2.8 Cadastro de Monstros

Nesta tela é criado a ficha dos monstros que serão utilizados contra os jogadores nas batalhas. Estes, serão controlados pelo mestre e aparecerão em grupos. A ficha do

monstro é bastante parecida com uma ficha de personagem com exceções de algumas características. A lista a seguir demonstra os atributos da ficha de monstro.

1. **Nome:** Nome do monstro;
2. **Teste de Resistência:** Resistência sobre os tipos de ataque que utiliza algum atributo como: Força, Destreza, Inteligência, Sabedoria, Constituição e Carisma;
3. **Imagem do Monstro:** Permite o usuário escolher uma viagem para o monstro;
4. **Nível:** Nível do monstro (pode considerar como a dificuldade do mesmo);
5. **XP:** Quantidade de experiência que o monstro dará aos jogadores que o derrotarem;
6. **Atributos**
  - **Vida:** Vida do monstro (ao chegar a 0 em batalha, o monstro é derrotado e retirado da mesma);
  - **Deslocamento:** Quantidade de movimentação que o monstro pode fazer;
  - **Armadura:** Defesa do monstro contra ataques inimigos;
  - **Teste Resistência:** Quais as resistências do monstro;
  - **Imunidade a Dano:** Qual tipo de dano o monstro é imune (cortante, perfurante...);
  - **Imunidade a Condição:** Tipo de imunidade a feitos como (sangramento, envenenamento e outros);
  - **Sentido:** Bônus em algumas áreas (como visibilidade no escuro, florestas e outros);
  - **Idioma:** Linguagens que o monstro tem domínio;
7. **Atributos:** Quantidade de cada atributo inicial do monstro (Força, Inteligência, Destreza, Sabedoria, Constituição e Carisma);

10 apresenta a tela de monstros.

Figura 10 – Tela Cadastro de Monstros

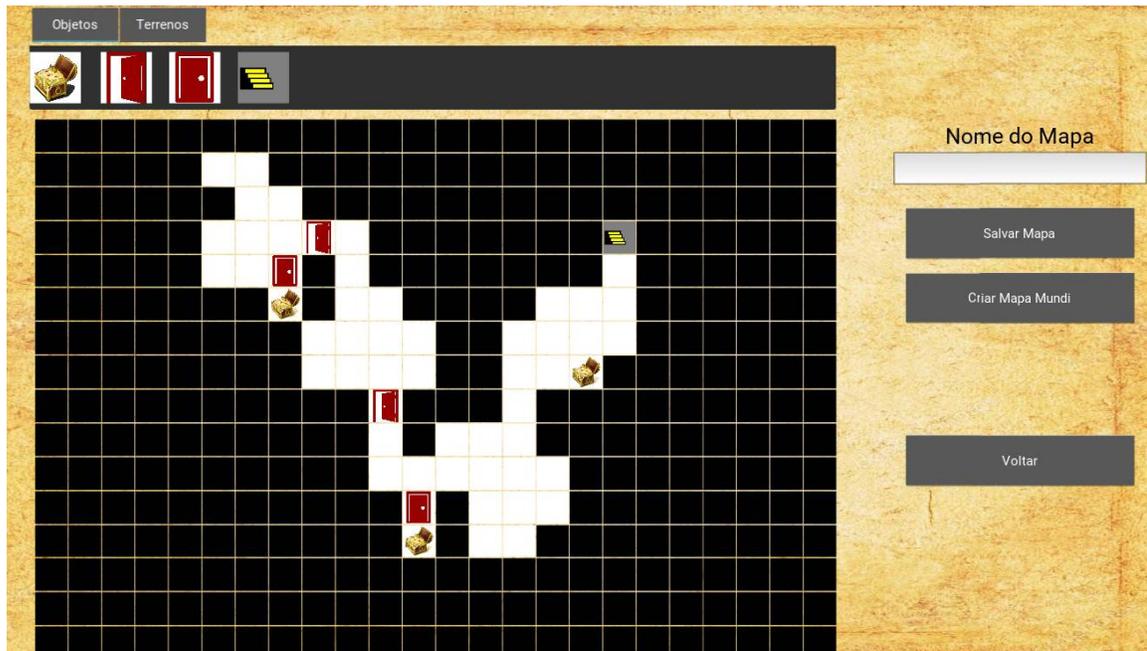
Fonte: Elaborado pelo autor.

### 3.3 Gerenciador de Mapas

Um dos pontos que mais atrai a atenção dos jogadores é a presença de diversos mapas, com ambientações diferentes. Estes mapas podem incluir cavernas, florestas, pântanos, dentre outros locais. Viabilizando a criação destes locais uma pesquisa sobre geradores de mapas foi realizada. A ideia inicial era utilizar o *website* Inkarnate, que constrói diversos tipos de mapas para *RPG's* e permite criar cidades, regiões com montanhas e florestas com biomas diferentes. Por conseguinte, a ferramenta não foi compatível com o sistema desenvolvido, pois, o resultado retornado é uma imagem monolítica e adicionar novos componentes ao mapa seria difícil em tempo de execução.

Assim, foi desenvolvido uma aplicação que está integrada ao simulador, que permite criar tanto regiões quanto cavernas (as famosas *dungeons* 11 representa um exemplo de criação de uma *dungeon*.

Figura 11 – Tela Cria *Dungeons*



Fonte: Elaborado pelo autor.

A linguagem *Kivy* proporciona controle de tamanho e posicionamento das imagens, durante a execução do gerenciador de mapas as mesmas podem ser alteradas sem a  
 1 observa-se componentes com suas características.

Tabela 1 – Tabela de componentes de *Dungeons*

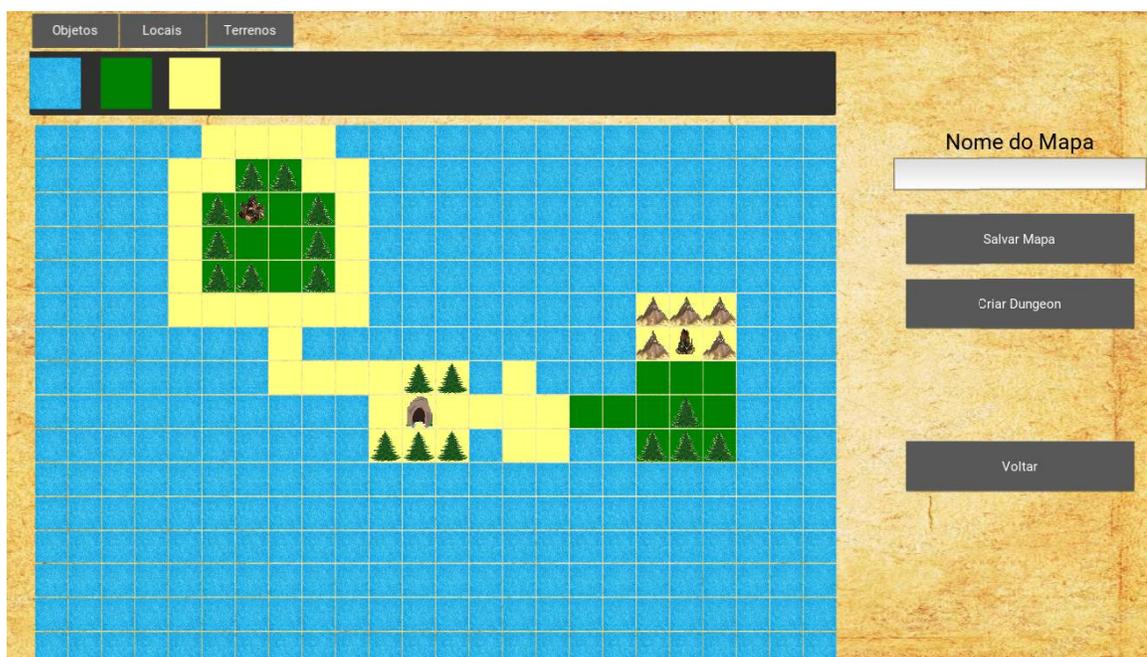
Imagem	Funcionalidade
	Parede ou local que não pode ser acessado pelos jogadores
	Chão para a passagem dos jogadores
	Porta que permite a passagem dos jogadores
	Porta que não permite a passagem dos jogadores
	Baú que contém algum item ou equipamento para os jogadores
	Permite aos usuários acessarem outro andar da <i>dungeon</i>

Fonte: Elaborado pelo autor.

Nas masmorras (*dungeons*) é onde diversos monstros residem e onde os personagens podem encontrar diversas riquezas. É neste local que o mestre pode criar uma aventura altamente desafiadora para os jogadores e definir diversos elementos novos a sua história.

Para adentrar as masmorras é necessário a criação prévia do mapa-mundi, através da ferramenta geradora de mapas, facilitando assim o acesso. A ferramenta foi desenvolvida para que o usuário crie inúmeros mapas até mesmo durante a campanha, a troca de mapas 12 demonstra a criação de um mapa-mundi.

Figura 12 – Tela Criação de Mapa-Mundi



Fonte: Elaborado pelo autor.

A construção do mapa-mundi não possui diferença em relação a criação de *dungeons*, somente os elementos visuais são trocados permitindo o sentimento de estar em um local 2 explica da elemento posicionado na tela.

Tabela 2 – Tabela de componentes de Mapa Múndi

Imagem	Funcionalidade
	Água, não pode ser acessado pelos jogadores
	Chão para a passagem dos jogadores com cor de grama
	Chão para a passagem dos jogadores com cor de areia
	Define naquele local um castelo
	Define naquele local uma cidade ou vila
	Define naquele local entrada de uma masmorra
	Montanha, não permite passagem de jogadores
	Árvore, não permite a passagem de jogadores

Fonte: Elaborado pelo autor.

### 3.3.1 Codificação do Gerenciador de Mapas

Para a criação do gerenciador de mapas, foram posicionados diversas cédulas de *Image Buttons* (Componente do Kivy que permite atribuir imagens a botões). Foram inseridas as abas que contém as imagens de cada elemento presente nas masmorras e mapas-mundi. Ao selecionar uma das imagens presentes nas abas e logo em seguida selecionar um botão, a imagem do mesmo será trocada pela imagem selecionada, criando o mapa desejado.

Com o mapa já construído, no campo “*edit*” será necessário digitar o nome do mapa, para finalizar a criação do mesmo pressiona-se o botão "Salvar Mapa", após um arquivo local será criado com a especificação da imagem de cada célula. Este arquivo será utilizado no momento que o mestre desejar realizar a troca de mapas durante a execução do jogo.

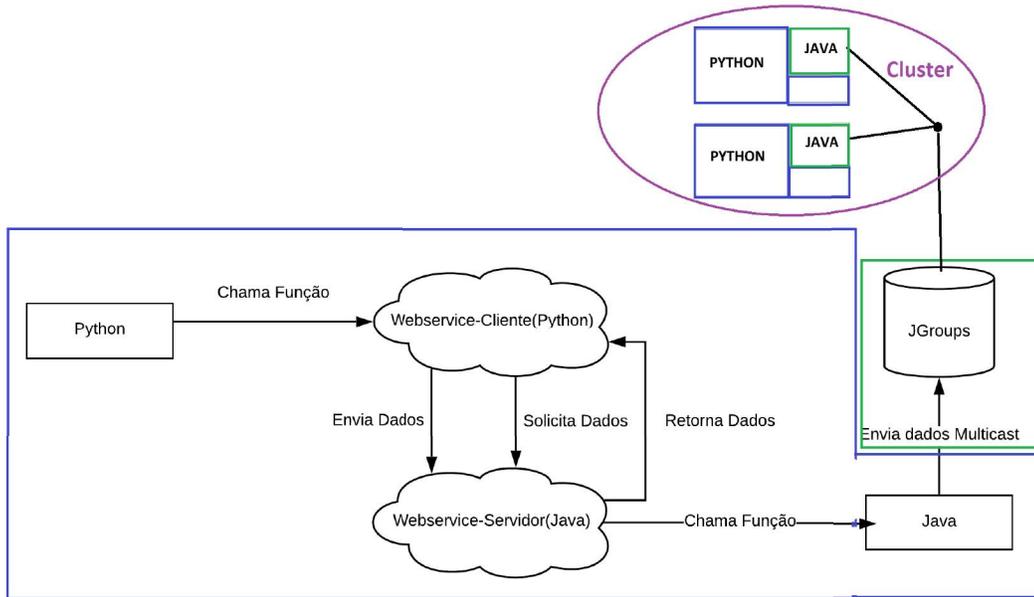
### 3.4 Comunicação entre os jogadores

Para que o jogo contenha fluidez e também a possibilidade de permitir diversos jogadores em uma mesma campanha, é preciso desenvolver um sistema de comunicação consistente e distribuído. Para isto foram utilizados diversos conceitos de sistemas distribuídos para tornar possível a troca de dados dos usuários (localizados em computadores diferentes, acessíveis pela rede). Cada usuário terá em seu computador duas aplicações em execução: O Python é responsável pela interface gráfica, regra de negócios e o Java responsável pela comunicação entre os jogadores.

Utilizando o *JGroups* como principal elemento da comunicação, pode-se definir *clusters* criados pelo mestre e estes são apresentados como salas para os demais jogadores. Os usuários conectados a estes *clusters* tem acesso a toda informação importante da partida corrente e de envio/recebimento de dados. O mestre possui todos os dados iniciais presentes da campanha salvos em seu banco de dados, assim logo que os jogadores conectam a partida a aplicação cliente (de cada jogador) solicita à aplicação servidora (do mestre) os primeiros dados necessários para o início do jogo. A cada alteração realizada em dados dos personagens, monstros e no mundo, são enviados via *multicast* para os jogadores, ou seja, todos que estão conectados naquela sala irão receber os dados. A linguagem predominante neste projeto é o *Python*, contudo a comunicação entre os processos clientes (jogadores) e o processo servidor (mestre) é toda realizada pelo Java com a utilização do *JGroups*. Assim para realizar a comunicação das aplicações é utilizado *Webservice*-servidor em Java e um *Webservice*-cliente em *Python*.

O *Webservice*-servidor funciona como uma porta para acesso às funcionalidades  
13 abaixo demonstra como é feita a  
comunicação.

Figura 13 – Diagrama Comunicação



Fonte: Elaborado pelo autor.

A aplicação em *Python*, a qual os jogadores tem acesso visualmente realizando chamadas locais a procedimentos, sendo que estas podem ser de envio e solicitações de dados. Caso seja para envio de dados, é feita uma chamada no *webservice* do *Java* e este aciona a função alvo do *JGroups* que implementa o serviço de comunicação desejado. Posteriormente a informação enviada é disseminada em todos os usuários conectados na sala por meio *multicast*. Entretanto se o usuário realizar uma solicitação de dados, a função correspondente será acionada no *webservice* do *Java* e o dado salvo será retornado para o *webservice* do *Python*.

Os dados enviados entre os processos do sistema distribuído ficam salvos na aplicação *Java* que está executando em segundo plano, assim são replicados igualmente a todos os jogadores, então para acessá-los necessita somente chamar a função correspondente do *webservice*. Para melhor explicação de envio e solicitações de dados é necessário o uso de exemplos, assim nas seções a seguir será explicado com mais detalhes o funcionamento do jogo, juntamente com os envios de dados.

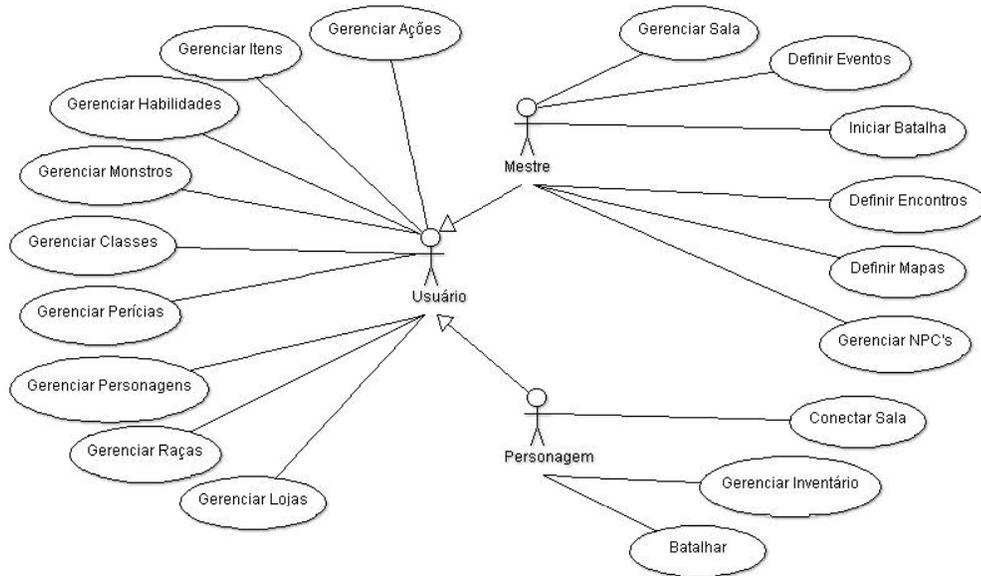
### 3.5 Executando o Simulador

Para facilitar e organizar a estrutura do jogo, a alimentação da base de dados inicial segue uma organização diferente, tornando possível somente criar certos tipos de dados antes de iniciar uma partida, como a criação de: ações, classes, habilidades, itens, lojas, monstros, perícias, personagens e raças. Ao dar início em uma partida, os dados

previamente cadastrados são carregados para a estrutura de dados do mestre e logo em seguida de acordo com a conexão de jogadores, disseminados entre eles.

14 demonstra como cada usuário (mestre e jogador) pode interagir com as funcionalidades do sistema.

Figura 14 – Diagrama de Casos de Uso



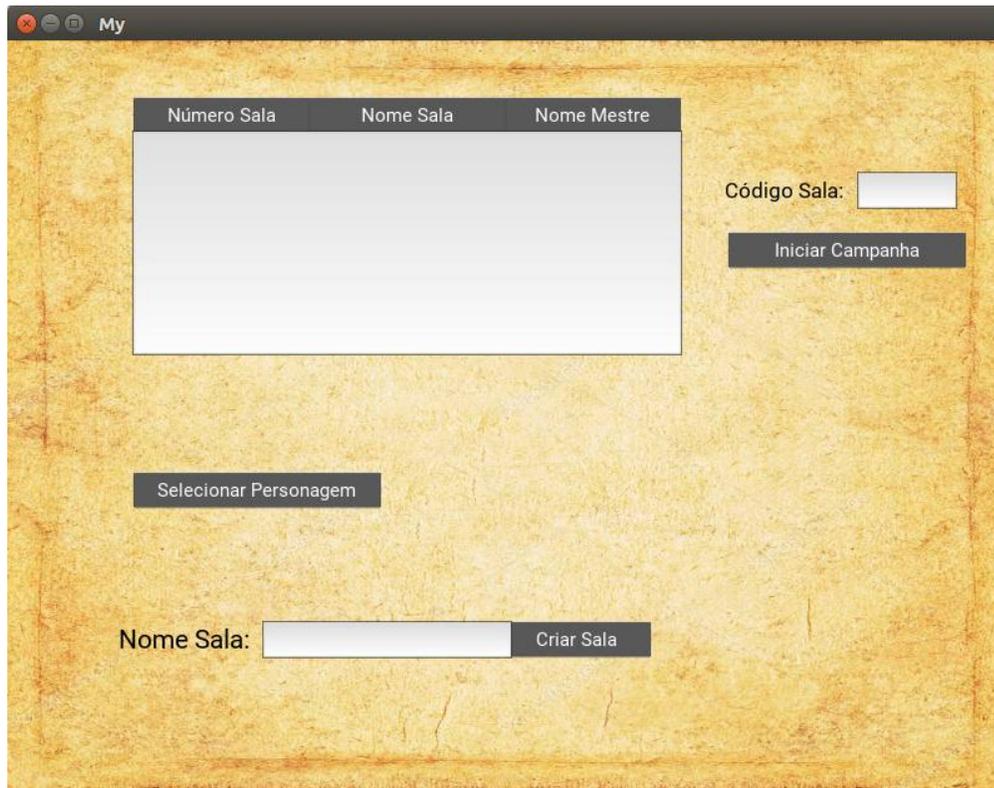
Fonte: Elaborado pelo autor.

### 3.5.1 Tela seleção sala

O primeiro contato com o universo *RPG* deste simulador é no momento de escolha da sala para partida. Os jogadores conectados ao *cluster* conseguem criar novas salas e visualizar todas as salas já criadas pelos jogadores. Quando um usuário cria uma sala, a mesma é enviada para o *JGroups* correspondente deste usuário, utilizando as funções de criações de sala disponíveis no *Webservice*. Este usuário torna-se automaticamente mestre e é conectado a sala criada, aguardando conexões de novos usuários. Quando um jogador executa o simulador a lista de sala é preenchida via *JGroups + Webservice*. Ao abrir a tela de salas o usuário aciona o *Webservice* que solicita ao *JGroups* todas as salas criadas, assim por meio de *multicast* as salas são enviadas a todos os novos usuários, permitindo

15 demonstra a tela de criação de salas.

Figura 15 – Tela Cria Sala



Fonte: Elaborado pelo autor.

Algumas regras são estabelecidas para que não ocorra conflito entre os nós conectados:

- Existe somente um mestre em cada sala;
- O número máximo de jogadores presentes em uma sala são 6 (5 personagens e 1 mestre);
- Os participantes só podem estar conectados a uma sala por vez;
- Cada jogador poderá controlar somente um personagem;
- Um jogador personagem não pode ser um jogador mestre, da mesma forma que um jogador mestre não pode ser um jogador personagem;

Para selecionar um personagem, o mesmo deve ser criado utilizando a ferramenta de criação disponível, assim selecionando a opção “Selecionar Personagem” é necessário selecionar um arquivo no disco com as informações do personagem e este será carregado para a estrutura adequada.

Ao iniciar a campanha todos os personagens criados são solicitados via *Webservice* e difundidos entre as aplicações via *JGroups*.

### 3.5.2 Tela mapa-mundi

Ao seleccionar e conectar em uma sala, os jogadores serão apresentados a tela do mapa-mundi. A tela apresenta o mapa criada pelo gerador de mapas, as funcionalidades do sistema e o acesso as demais telas.

Na aba cliente estão localizadas três funcionalidades, nas quais os jogadores podem 3 as evidenciam.

Tabela 3 – Tabela de componentes mapa-mundi

Imagem	Funcionalidade
	Realiza o movimento dos jogadores
	Acessa a tela inventário
	Acessa a tela de loja

Fonte: Elaborado pelo autor.

- Movimento: Permite que o grupo de jogadores se desloque no mapa-mundi, possibilitando acesso a cavernas ou cidades. Caso o grupo realize movimentações é possível que batalhas com monstros se iniciem;
- Inventário: Realiza acesso ao inventário do jogador. Cada personagem tem acesso único ao seu inventário;
- Loja: Realiza acesso a loja para a compra de itens;

Logo abaixo da aba cliente encontra-se o chat para os usuários, no qual permite a troca de mensagens entre os jogadores e informações importantes do sistema. Esses dados são enviados via *JGroups* e *Webservice*, possibilitando que todos os usuários recebam via *multicast*.

Na aba superior estão presentes a funcionalidades para o mestre. Na aba “Troca Mapa” permite que o mestre escolha um novo mapa-mundi, assim este arquivo é carregado do sistema e enviado a todos os usuários conectados ao *cluster*.

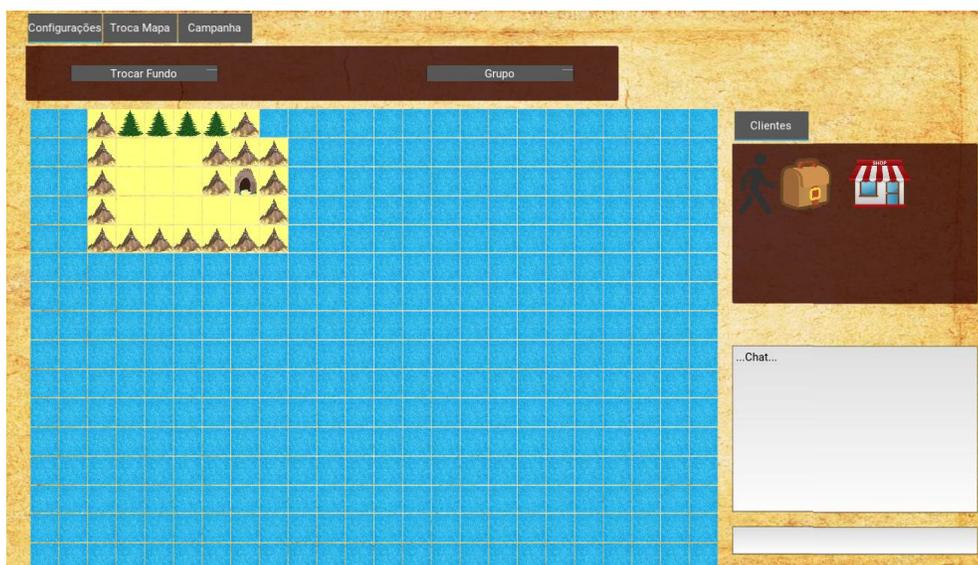
Na aba campanha estão presentes três funcionalidades importantes para a aplicação: mostra determinadas funções.

Imagem	Funcionalidade
	Define missões para os jogadores
	Acessa a tela para definição de grupos de monstros usado em batalhas
	Acessa a tela de batalha

Tabela 4 – Tabela de componentes mapa-mundi: campanha

16 ilustra a tela mapa-mundi.

Figura 16 – Tela Mapa-Múndi



Fonte: Elaborado pelo autor.

### 3.5.3 Encontros

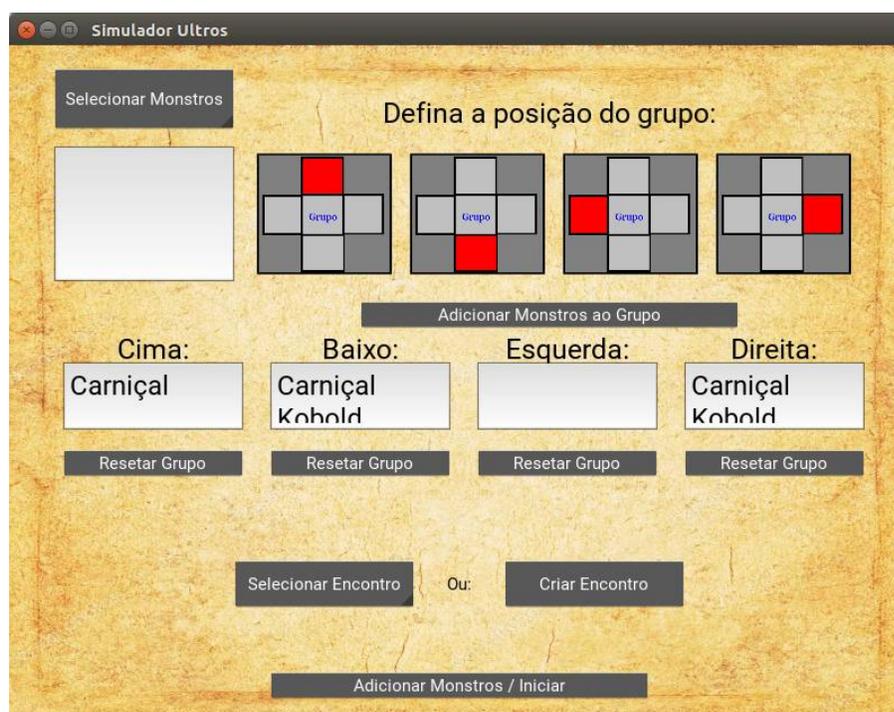
Na tela de gerenciamento de campanha – criação de encontros, é possível definir grupos de monstros para participarem das batalhas. Todos os monstros previamente criados

pelos mestres estarão presentes para a montagem de cada grupo. Um encontro é dividido em quatro grupos: cima, baixo, esquerda e direita, que são respectivamente as posições: frente, costas, esquerda e direita do grupo de personagens. Essa divisão é realizada objetivando posicionar monstros de forma que tais possuam mais vantagens ou desvantagens sobre os jogadores. Cada posição permite no mínimo nenhum monstro e no máximo cinco.

Cada encontro criado é salvo no banco de dados do mestre (Servidor) possibilitando seu uso em batalhas. A criação desta funcionalidade permite que o sistema crie encontros aleatórios durante as ações dos jogadores fazendo que o jogo se torne mais dinâmico, além de facilitar a criação de batalhas para o mestre.

17 demonstra a tela de gerenciamento de encontros.

Figura 17 – Tela Encontro



Fonte: Elaborado pelo autor.

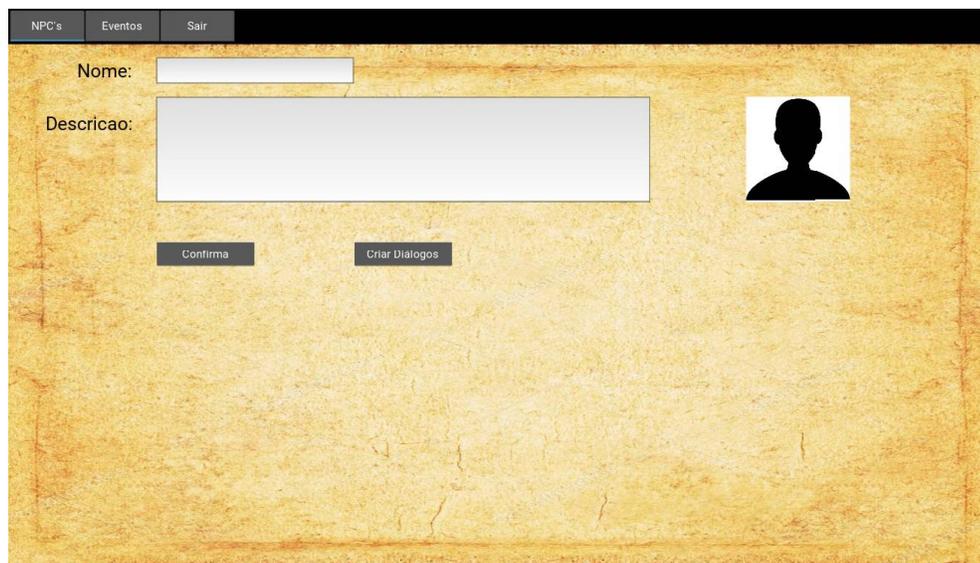
### 3.5.4 NPC's

Além dos jogadores e dos monstros controlados pelo mestre, existem também os NPC's, “personagens não jogáveis”, estes personagens geralmente possuem a função de entregar aos jogadores informações importantes, missões e contar alguma história. O mestre tem a opção de cadastrar falas para estes personagens e estes estão presentes em todo o universo criado pelo narrador, podendo estar em locais de difícil acesso ou não. Quando um NPC é criado, os usuários recebem pela ferramenta *JGroups* tais informações,

assim todos os jogadores terão acesso ao NPC.

18 apresenta o gerenciamento de de NPC's juntamente com suas falas.

Figura 18 – Tela NPC



Fonte: Elaborado pelo autor.

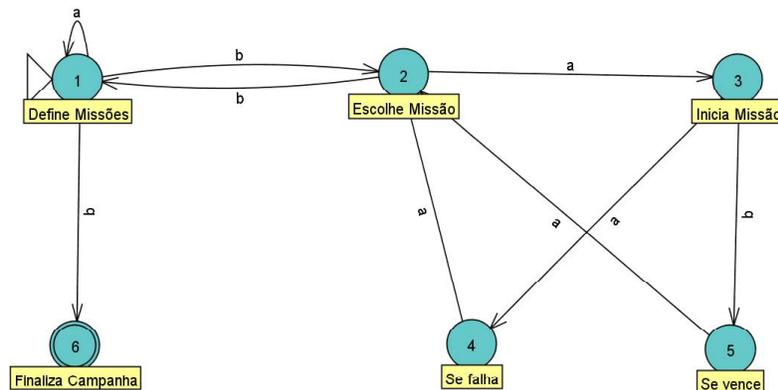
### 3.5.5 Campanha

Desenvolver uma campanha de uma partida de D&D é algo que precisa de muito tempo e criatividade, tendo em vista esta dificuldade a tela campanha permite que o mestre crie missões para os jogadores. As missões são etapas que o grupo de jogadores devem seguir para completar um objetivo maior, no qual finaliza a campanha. As missões são classificadas em dois tipos:

- Conversar com NPC;
- Derrotar algum monstro;

O conjunto de missões possuem informações e uma história, que unidas constitui todo o enredo da partida criada. As missões são enviadas para todos os jogadores, assim todos conseguem visualizar os objetivos já concluídos, os correntes e os que ainda não foram completados. O sistema de campanha funciona seguindo um autômato, que é demonstrado

Figura 19 – Autômato para Campanhas

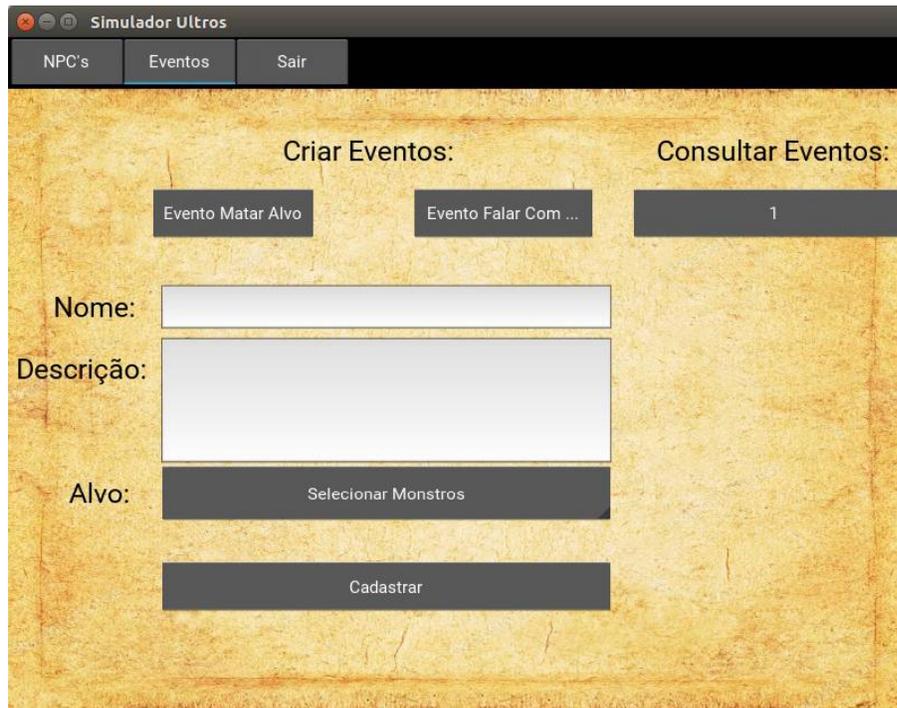


Fonte: Elaborado pelo autor.

- Estado 1: Define as missões com suas descrições e seus alvos, caso crie missão ativa estado 2, do contrário ativa estado 6;
- Estado 2: Escolhe a missão a ser passada aos jogadores e ativa estado 3, caso não exista mais missões para seleção ativa estado 1;
- Estado 3: Inicia a missão selecionada, quando completada com sucesso ativa estado 4, caso falhe ativa o estado 5.
- Estado 4: Define as penalidades aos jogadores e ativa estado 2;
- Estado 5: Define as recompensas aos jogadores e ativa estado 2;
- Estado 6: Atribui recompensas geral da campanha e a finaliza;

20 mostra o resultado da tela de criação de campanhas.

Figura 20 – Tela Campanha



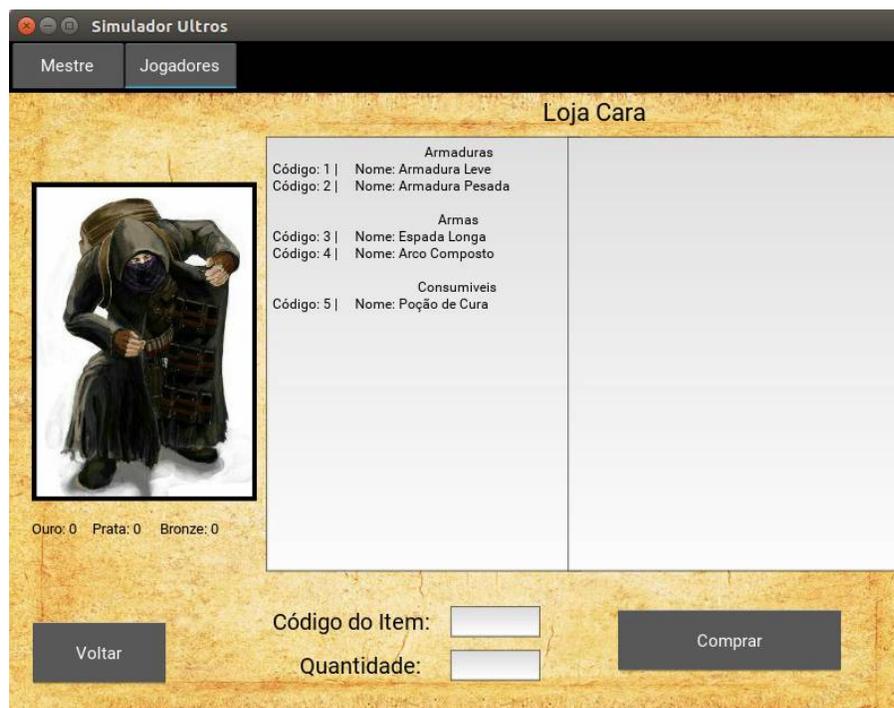
Fonte: Elaborado pelo autor.

### 3.5.6 Loja

Ao acessar a tela de loja, as listas de itens disponíveis serão preenchidas para todos. Cada compra realizada na loja é de maneira individual, ou seja, todos os itens comprados serão enviados para o inventário do seu respectivo comprador. Toda compra de item é realizada gastando as moedas do jogo: ouro, prata e bronze, caso o usuário não tenha os valores necessários para realizar a compra, o mesmo não será enviado para o inventário do comprador, o envio dos itens comprados é realizado de maneira *unicast*, ou seja, somente aquele usuário receberá as informações difundidas no *JGroups* e *Webservice*.

21 demonstra a tela de compra de itens para os jogadores.

Figura 21 – Tela aba loja



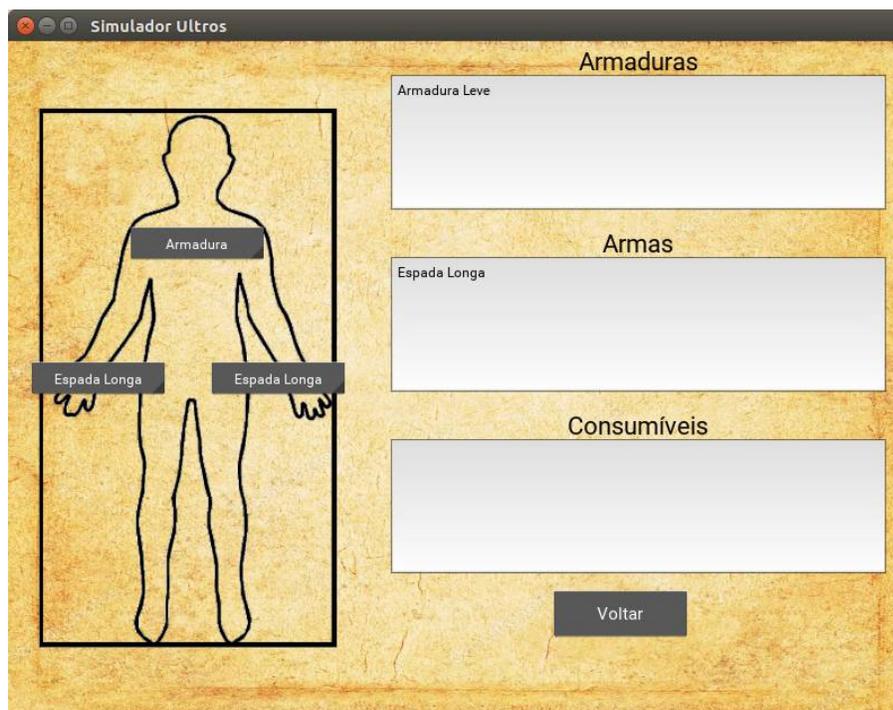
Fonte: Elaborado pelo autor.

### 3.5.7 Inventário

Permite que o usuário verifique todos os itens coletados durante a campanha, além de permitir que sejam equipadas suas armas e armaduras. Cada usuário possui a lista de seus itens separados e não é possível realizar a troca entre os participantes.

22 demonstra o resultado da criação desta funcionalidade.

Figura 22 – Tela Inventário



Fonte: Elaborado pelo autor.

### 3.5.8 Batalha

A tela de batalha é dividida em duas partes: o gerenciador de batalhas e a tela de batalha propriamente dita. O gerenciador de batalha pode ser acessado somente pelo mestre e é nesta tela a qual o mestre escolhe o encontro desejado para os jogadores. A 23 demonstra a tela de seleção de encontros para batalha.

Figura 23 – Tela seleção encontro para batalhas



Fonte: Elaborado pelo autor.

Quando os jogadores iniciam uma partida uma *thread* assíncrona é criada, sua funcionalidade é verificar quais solicitações o mestre e os jogadores realizam. Quando o mestre inicia a batalha, uma solicitação é enviada ao *Webservice*, e este avisa ao sistema que uma batalha irá se iniciar, logo em seguida os dados são enviados via *JGroups* a todos os jogadores conectados na partida, iniciando a mesma em todas as aplicações que estejam rodando o simulador. Quando a batalha é definida e é iniciada todos os jogadores conectados recebem as informações do mestre, nas quais contém: dados dos monstros e a ordem de cada jogador. Toda a batalha acontece em turnos, sendo assim cada jogador terá de esperar a ação dos demais para realizar a sua. Para definir qual turno é de cada jogador, a cada ação o turno do próximo é chamado e estes dados são enviados de forma *multicast*. Os monstros são controlados pelo mestre, tendo seus turnos organizados juntamente com os personagens.

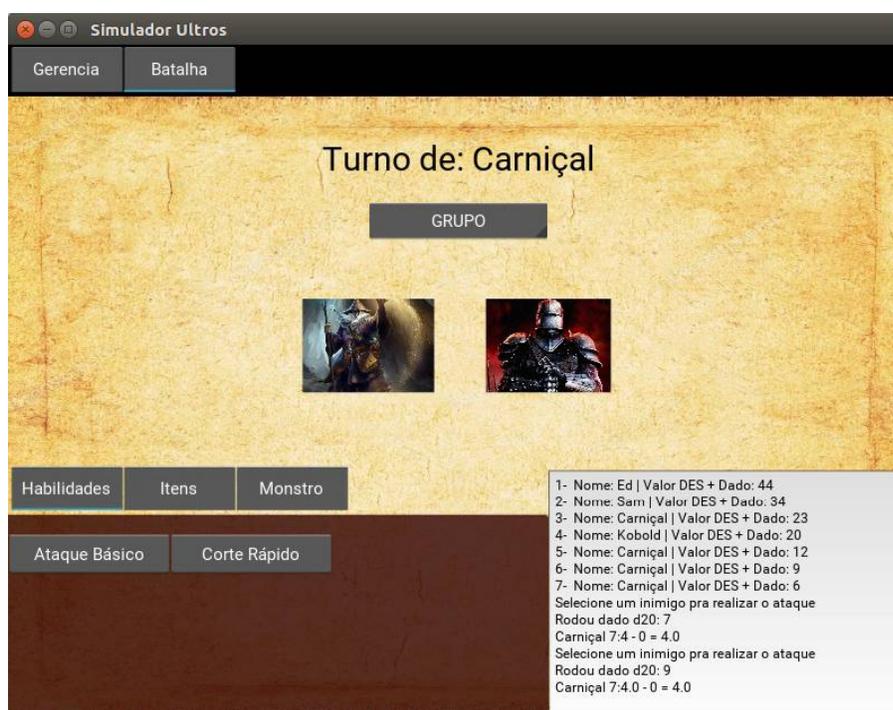
Cada personagem possui suas próprias habilidades e estas não podem ser utilizadas em membros do próprio grupo. Para realizar alguma ação, dados precisam ser jogados e a conclusão da ação ou não depende do resultado. Todos os cálculos de rolamento de dados são feitos automaticamente, não necessitando que os usuários realizem alguma ação. Os valores de acerto, erro e crítico são baseados na literatura dos livros de RPG mencionados nesta monografia. As habilidades possuem atributos especiais, estes causam efeitos aos alvos selecionados. Caso o usuário seja atingido e entre em estado de sangramento

e envenenamento, a cada turno deste jogador sofrerá perda da vida, caso o jogador ou monstro seja atingido por estado de cegueira, a chance de acerto é diminuída drasticamente, afetando suas ações.

Para selecionar os alvos há um campo que possui todos os tipos de visões: cima, baixo, esquerda, direita e grupo. Na área grupo estão presentes os personagens e nas demais estão presentes os grupos de monstros definidos pelo encontro.

24 demonstra a formatação da tela e o funcionamento dos posições dos grupos.

Figura 24 – Tela batalha



Fonte: Elaborado pelo autor.

A batalha só irá finalizar caso todo o grupo de personagens ou de monstros estejam com a vida zerada, no primeiro caso ocorre vitória para os jogadores, no segundo derrota. Se o grupo ganhar a partida todos os jogadores são recompensados com o dinheiro do jogo, entretanto se o grupo não sobreviver a batalha se encerra a partida.

## 4 Conclusão

O desenvolvimento do presente trabalho possibilitou a criação de um *software* que<sup>1</sup> de campanhas de *Dungeon and Dragons*, implementando diversas funcionalidades essenciais que auxiliam o jogador mestre na elaboração e realização das mesmas. Este trabalho pode ser aplicado em diversas áreas com finalidades diferentes; com a liberdade fornecida é possível que o mestre desenvolva suas histórias abordando temas aplicados em aulas de ensino médio, principalmente nas áreas de história, geografia, as quais abordam sobre diferentes tipos de biomas e eventos históricos.

Durante todo o processo de desenvolvimento, as ferramentas utilizadas foram de grande importância para toda a criação, quando integralizadas apresentaram um bom desempenho, não afetando a jogabilidade dos jogadores. Um dos pontos fortes deste trabalho está na parte da comunicação entre os usuários, na qual foram utilizadas duas ferramentas que comunicam entre si: *Webservice* e *JGroups*. O uso destas tornou a comunicação mais leve, com escalabilidade e disponibilidade, possibilitou o uso de duas linguagens em uma só aplicação.

Ao focar no auxílio do mestre, é possível abordar diversos fatores que facilitam a utilização do *software* pelos jogadores durante todo o processo: as fichas de personagens e monstros possuem somente os campos principais e ainda assim permitem criar seus personagens com mais autonomia. Determina a criação de mapas, uma campanha com a história de acordo com o que o mestre deseja e com a disponibilidade dos recursos presentes na ferramenta. Facilita o cálculo e a organização das batalhas, tanto quanto o seu início, pois este momento é um dos que mais exigem atenção dos jogadores e do mestre.

Com a abordagem deste tema no projeto, foi adquirido um vasto conhecimento sobre as tecnologias que nele foram utilizadas, como programação e criação de interfaces gráficas com Python, desenvolvimento de *Webservices* servidores e consumidores, a geração de um banco de dados que armazena uma grande massa de dados e a utilização do *JGroups*.

Assim, é possível observar que grande parte das disciplinas do curso de ciência da computação foi empregada neste projeto, através do conhecimento adquirido dentro da sala de aula, o que proporcionou um aprimoramento e amadurecimento para minha atuação profissional.

---

<sup>1</sup> Permite que o gerenciador controle os dados de forma mais automatizada, calculando os valores das fichas dos jogadores e monstros de forma mais rápida, sem a necessidade de envolver o mestre nos cálculos, na rodagem dos dados e definições de batalhas.

## 5 Trabalhos futuros

Para um melhor aproveitamento e dinamismo procurado pelos usuários, uma das funcionalidades para trabalhos futuros seria o aprimoramento do gerenciador de mapas com novas características e a criação de novas formas de comunicação entre os jogadores (envio de áudio e figuras). A confecção de um sistema de batalha visual com o emprego de elementos mais atrativos e impactantes. A determinação da hora virtual corrente do jogo que facilitaria a alteração de dia e noite, afetando de forma direta a jogabilidade dos personagens. Criação de um módulo inteligente no qual proporcionaria ao mestre a possibilidade de desenvolver combates mais difíceis caso os jogadores não siga a campanha principal. Por fim o desenvolvimento de uma funcionalidade que permita a criação de personagens pré-existentes, agilizando todo o processo inicial do jogo.

## Referências

- ALVES, D. M.; PADOVANI, S. Estabelecendo relações entre critérios de avaliação ergonômica em hci e recomendações de game design. *Recife: Universidade Federal de Pernambuco* 23.
- BORGES, L. E. *Python para desenvolvedores: aborda Python 3.3*. [S.l.]: Novatec Editora, 19.
- COMPUTER, C. U. P. of. *What is Computer Graphics?* <<http://www.graphics.cornell.edu/online/tutorial/>> 14.
- DUSTMAN, A. *MySQLdb User's Guide* <<http://mysql-python.sourceforge.net/MySQLdb.html>> 25.
- FANNON, S. P. *The Fantasy Roleplaying Gamer's Bible (2nd ed.)*. [S.l.]: Obsidian Studios 13.
- GRANDO, A.; TAROUCO, L. M. R. O uso de jogos educacionais do tipo rpg na educação. *RENOTE* 13.
- GYGAX, G. *The World of Greyhawk* 13.
- INDRUSIAK, L. S. Linguagem java. *Grupo JavaRS JUG Rio Grande do Sul*, 1996. 20.
- JGROUPS. *JGroups - A Toolkit for Reliable Messaging* <<http://www.jgroups.org/>>. 21.
- JONES, K. *Simulations: A Handbook for Teachers and Trainers*. [S.l.]: Routledge, 2013. 14.
- KIVY. *Kivy - Open source Python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps*. <<https://kivy.org/#home>> 19.
- MAGRI, J. A. Criando e usando web service. *Augusto Guzzo Revista Acadêmica*, n. 11, p. 20.
- MARQUES, J. A.; CEONI, K. T. O histórico e a importância da mídia digital nos jogos on-line: dos rpgs tradicionais para os crpgs e os on-line. In: *Intercom. São Paulo: Sociedade Brasileira de Estudos Interdisciplinares da Comunicação. V Congresso Nacional de História da Mídia* 18.
- MILANI, A. *MySQL-guia do programador*. [S.l.]: Novatec Editora, 2007. Citado na 21.
- MYSQL. *MySQL Enterprise Edition* <<https://www.mysql.com/products/enterprise/>> 21.

- NEWZOO. *The Global Games Market Will Reach \$108.9 Billion in 2017 With Mobile Taking 42%* <<https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/>>. 12.
- OLIVEIRA, R. C. de; PIERSON, A. H. C.; ZUIN, V. G. O uso do role playing game (rpg) como estratégia de avaliação da aprendizagem no ensino de química. 2009. Citado na 14.
- PETERSON, J. *Forty Years of Adventure* <<http://dnd.wizards.com/dungeons-and-dragons/what-dd/history/history-forty-years-adventure>>. Accessed: 16.
- SCHNEIDER, M. F. Scrum'ed: um jogo de rpg para ensinar scrum. *Undergraduate thesis. Federal University of Santa Catarina, Florianópolis, SC, Brazil*, 2015. Citado 2 vezes nas 13 14.
- TEAM, W. R. *Guia do Mestre* 13 24.
- TEAM, W. R. *Livro do Jogador* 13 24.
- TEAM, W. R. *Manual dos Monstros* 13 24.
- WORKBENCH, M. *MySQL Workbench* <<https://www.mysql.com/products/workbench/>> 22.