

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**ORC – ONLINE RPG CREATOR: UM PROTÓTIPO PARA  
AUXÍLIO NA CRIAÇÃO DE NARRATIVAS DE JOGOS DE  
INTERPRETAÇÃO RPG ONLINE**

**DIRCEU PIANA JUNIOR**

**BLUMENAU**  
**2016**

**DIRCEU PIANA JURNIO**

**ORC – ONLINE RPG CREATOR: UM PROTÓTIPO PARA  
AUXÍLIO NA CRIAÇÃO DE NARRATIVAS DE JOGOS DE  
INTERPRETAÇÃO RPG ONLINE**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Aurélio Faustino Hoppe, Mestre - Orientador

**BLUMENAU  
2016**

**ORC – ONLINE RPG CREATOR: UM PROTÓTIPO PARA  
AUXÍLIO NA CRIAÇÃO DE NARRATIVAS DE JOGOS DE  
INTERPRETAÇÃO RPG ONLINE**

Por

**DIRCEU PIANA JUNIOR**

Trabalho de Conclusão de Curso aprovado  
para obtenção dos créditos na disciplina de  
Trabalho de Conclusão de Curso II pela banca  
examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Aurélio Faustino Hoppe, Mestre – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Dalton Solano dos Reis, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Mauro Marcelo Mattos, Doutor – FURB

Blumenau, 8 de julho de 2016

Dedico este trabalho à minha família e meus amigos, que me deram apoio e sempre estiveram ao meu lado.

## **AGRADECIMENTOS**

A minha família que me apoiou incondicionalmente, mais do que eu me julgo digno.

Aos meus amigos, que me deram um suporte inestimável para que eu realizasse o trabalho.

Ao meu orientador, Aurélio Faustino Hoppe, por me apoiar, por me direcionar, e por acreditar na minha capacidade e na minha ideia.

Um livro é a prova de que os homens são capazes de fazer magia.

Carl Sagan

## RESUMO

Este trabalho apresenta o desenvolvimento de um protótipo para criar jogos narrativos de RPG num ambiente *online*. O ORC, como foi chamado, disponibiliza um ambiente *online* que simula um tabuleiro de RPG, permitindo aos usuários criem e personalizem seus personagens. Para auxiliar os jogadores nesta tarefa, ele disponibiliza modelos 3D pré-fabricados, o tabuleiro, a ficha de personagem, assim como, dois *chats* de comunicação: escrito e por voz. A partir dos experimentos realizados com usuários, foi verificado que o ORC pode ser utilizado como ferramenta de auxílio ao RPG, pois conseguiu simular bem uma mesa de jogo.

Palavras-chave: RPG. Online. Personagens. Modelo 3D. Mesa de jogo.

## **ABSTRACT**

This work presents the development of a prototype to create narrative RPG games in an online environment. The ORC, as it was called, offers an online environment that simulates an RPG board, allowing users to create and customize their characters. To help the players in this task, it provides pre-made 3D models, the board, the character sheet, as well as two communication chats: writing and voice. From the experiments with users, it was verified that the ORC can be used as a tool to RPG, it could well simulate a game table.

Key-words: RPG. Online. Characters. 3D Model. Game table.



## LISTA DE FIGURAS

Figura 1 – Conjunto de dados de RPG .....	16
Figura 2 – Ficha de um monstro.....	18
Figura 3 – Ficha de personagem.....	19
Figura 4 – Miniaturas de D&D.....	20
Figura 5 – Jogo Elder Scrolls .....	21
Figura 6 – Tela principal do <i>Tabletop Forge</i> .....	23
Figura 7 – Tela principal do Taulukko .....	24
Figura 8 – Tela principal do RRPg Firecast .....	25
Figura 9 – Diagrama de casos de uso .....	28
Figura 10 – Diagrama de classes dos objetos de personagens .....	29
Figura 11 – Diagrama de classes da criação de personagens .....	30
Figura 12 – Diagrama de classes da conexão com o Photon.....	31
Figura 13 – Diagrama de classes sobre o controle e criação de mapas.....	32
Figura 14 – Diagrama de atividades .....	33
Figura 15 – Exemplo de uma imagem referência para modelagem .....	36
Figura 16 – Exemplo de malha básica com modificadores aplicados.....	37
Figura 17 – Exemplo de modelo detalhado .....	38
Figura 18 – Exemplo de modelo com esqueleto .....	39
Figura 19 – Exemplo de modelo com esqueleto .....	40
Figura 20 – Fluxo de atividades .....	41
Figura 21 – Lista de usuários <i>offline</i> .....	41
Figura 22 – Menu principal .....	42
Figura 23 – Definição de Perícias .....	43
Figura 24 – Edição de modelo de PdM .....	46
Figura 25 – Configurações de Cenário .....	46
Figura 26 – Criação de mapa.....	47
Figura 27– Edição de elemento do mapa .....	48
Figura 28 – Clonagem de elemento do mapa .....	48
Figura 29 – Tela de Servidores.....	50
Figura 30 – Tela principal do ORC .....	52

## LISTA DE QUADROS

Quadro 1 – Características dos trabalhos correlatos.....	26
Quadro 2 – Atividades por módulos.....	41
Quadro 3 – Carregamento de itens editáveis .....	44
Quadro 4 – Cálculo de deformação do osso .....	45
Quadro 5 – Controle de item selecionado .....	49
Quadro 6 – Posicionamento do item .....	49
Quadro 7 – Comunicação voip .....	51
Quadro 8 – Conexão com o Photon.....	53
Quadro 9 – Criação de salas .....	53
Quadro 10 – Perfis dos usuários .....	54
Quadro 11 – Respostas do questionário sobre a avaliação dos instrumentos do ORC .....	55
Quadro 12 – Respostas do questionário de usabilidade da criação de PdJ.....	55
Quadro 13 – Respostas do questionário de usabilidade da tela de jogo (Jogador).....	56
Quadro 14 – Respostas do questionário de usabilidade da criação de PdM.....	56
Quadro 15 – Respostas do questionário de usabilidade da criação de mapas .....	56
Quadro 16 – Respostas do questionário de usabilidade da tela de jogo (Mestre) .....	57
Quadro 17 – Respostas do questionário de usabilidade geral do ORC .....	57
Quadro 18 – Comparação entre os trabalhos correlatos e o ORC.....	58
Quadro 19 – Caso de uso UC01 - Cadastro de usuários.....	62
Quadro 20 – Caso de uso UC02 - Implementar ficha de PdJs.....	62
Quadro 21 – Caso de uso UC03 - Customizar PdJs.....	63
Quadro 22 – Caso de uso UC04 - Cadastrar PdMs.....	63
Quadro 23 – Caso de uso UC05 - Customizar PdMs.....	64
Quadro 24 – Caso de uso UC06 - Criar mapas.....	64
Quadro 25 – Caso de uso UC07 - Criar narrativa.....	65
Quadro 26 – Caso de uso UC08 - Gerar rolagem de dados .....	65
Quadro 27 – Caso de uso UC09 - Restringir jogadas.....	65
Quadro 28 – Caso de uso UC10 - Controlar pontos de experiência.....	66
Quadro 29 – Caso de uso UC11 - Controle de vida e fichas.....	66
Quadro 30 – Caso de uso UC12 - Comunicação por chat.....	67

Quadro 31 – caso de uso UC13 - Disponibilizar prefabs .....	67
Quadro 32 – Questionário de perfil de usuário.....	68
Quadro 33 – Lista de tarefas a serem realizadas pelo jogador .....	69
Quadro 34 – Lista de tarefas a serem realizadas pelo mestre.....	69
Quadro 35 – Questionário geral sobre o ORC.....	70

## LISTA DE ABREVIATURAS E SIGLAS

D&D – *Dungeons & Dragons*

IDE – *Integrated Development Environment*

MMORPG – *Massively Multiplayer RolePlaying Game*

MUD - *Multi User Dimension*

NPC - *Non-player Character*

ORC – *Online RPG Creator*

PdJ – *Personagem do Jogador*

PdM – *Personagem do Mestre*

RF- *Requisito Funcional*

RNF- *Requisito Não-Funcional*

RPG – *RolePlaying Game*

SDK – *Software Development Kit*

UML – *Unified Modeling Language*

UV - *Unplesant Visual*

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>14</b>
1.1 OBJETIVOS.....	15
1.2 ESTRUTURA.....	15
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>16</b>
2.1 RPG DE MESA.....	16
2.2 TRABALHOS CORRELATOS .....	22
2.2.1 Tabletop Forge .....	22
2.2.2 Taulukko .....	24
2.2.3 RRPg Firecast.....	25
2.2.4 Comparativo entre trabalhos correlatos .....	26
<b>3 DESENVOLVIMENTO.....</b>	<b>27</b>
3.1 LEVANTAMENTO DE INFORMAÇÕES .....	27
3.2 ESPECIFICAÇÃO .....	27
3.2.1 Diagrama de casos de uso .....	27
3.2.2 Diagrama de classes .....	29
3.2.3 Diagrama de atividades .....	33
3.3 IMPLEMENTAÇÃO .....	34
3.3.1 Técnicas e ferramentas utilizadas.....	35
3.3.2 Modelagem 3D.....	35
3.3.3 Implementação do ORC.....	40
3.3.4 Seleção de usuário.....	41
3.3.5 Criação de personagens.....	42
3.3.6 Criação de mapas .....	46
3.3.7 Durante o jogo.....	50
3.3.8 Conexão com o Photon PUN .....	52
3.4 RESULTADOS E DISCUSSÕES.....	53
3.4.1 Metodologia .....	54
3.4.2 Cenário de teste .....	54
3.4.3 Comparação entre os trabalhos correlatos e o trabalho proposto.....	57
<b>4 CONCLUSÕES .....</b>	<b>59</b>
4.1 EXTENSÕES .....	59

<b>APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO .....</b>	<b>62</b>
<b>APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO DO ORC .....</b>	<b>68</b>

## 1 INTRODUÇÃO

Os jogos fazem parte da história da evolução humana, sendo fundamentais na cultura, desde os tempos primitivos. Entre as principais características dos jogos, pode-se destacar a prática do “fazer de conta”, muito presente nos jogos atuais, que os torna em ambientes interativos, mesmo que limitados até certo ponto pela tecnologia. Os jogos eletrônicos, principalmente os jogos online vem ganhando notoriedade na sociedade e na vida de seus usuários, pois proporcionam comunicação entre as pessoas, influenciando assim a sociedade e a cultura, além de ter impacto direto sobre o desenvolvimento científico e a tecnologia (RODRIGUES; LOPES; MUSTARO, [2008?]).

O RolePlaying Game (RPG) é uma modalidade de jogo abrangente, o qual o jogador pode ser quem ele quiser, ou seja, o limite do possível é determinado pela imaginação. Dessa maneira, o RPG influenciou o surgimento de inúmeros jogos, filmes, séries, além de servir como parte integrante na formação humana e profissional de várias pessoas (ANDRADE, 2000). Dentre as modalidades de RPGs destacam-se o de mesa, o de mesa *online*, o *live action*, o *Massively Multiplayer Online RolePlaying Game* (MMORPG), entre outros.

Dentre os adeptos do RPG, a grande maioria dos jogadores preferem o jogo de mesa tradicional, seguidos pela grande massa de usuários dos MMORPGs. Outra modalidade de RPG muito difundida é a partida *live action* (ZAPT MÍDIA, 2009). Como bem define Riyis (2004), os jogadores personificam teatralmente os personagens, construindo uma história coletivamente. Para tanto, os sujeitos envolvidos buscam inspiração em experiências próprias, baseadas em leituras e o envolvimento com diversos tipos de mídia.

Segundo ZAPT Mídia (2009), o mercado de RPG é razoavelmente amplo. O preço pode variar de R\$50,00 até mais de R\$500,00, resumindo-se em livros e suplementos às histórias, como dados, escudos de narrador, miniaturas de personagens, entre outros. Dessa forma, pode-se analisar que o mercado gira em torno de um produto consideravelmente caro, sendo obtido por jogadores já adeptos aos jogos de RPG. Ainda de acordo com a pesquisa realizada pela empresa ZAPT Mídia (2009), 25% dos jogadores que ficaram mais de 12 meses sem jogar, justificam a ausência por falta de grupo. Ainda nesta linha, 60% dos questionados afirmam que é difícil encontrar membros de grupos para jogar. A pesquisa afirma também que existe atualmente, uma dificuldade por parte dos jogadores em encontrar outros adeptos. Há que se observar que existe uma parcela representativa desses jogadores que se ausentam das mesas por falta de grupo, muitas vezes não tendo mais possibilidade de voltar a jogar por não ter com quem se reunir. Dessa maneira, o dia-a-dia corrido da sociedade atual, juntamente

com as responsabilidades que o ser humano adquire com o passar do tempo, o RPG, como atividade que proporciona diversão, desenvolvimento intelectual e da criatividade, perde espaço na vida de muitas pessoas.

De toda forma, conclui-se que o mercado de RPG encontra-se em crescimento, com a evolução do hábito da leitura e constante valorização da criatividade na sociedade atual. Sendo assim, este trabalho apresenta o desenvolvimento de um protótipo para construção de RPG que possa ser usada de forma que ajude a agilizar os processos do jogo e suas regras, tendo como intuito disponibilizar um ambiente online para que a partida se assemelhe o máximo possível com as tradicionais e que além disso, disponha de recursos visuais para que a imersão no jogo seja mais realista e dinâmica.

## 1.1 OBJETIVOS

Este trabalho tem como objetivo desenvolver um protótipo para auxiliar a criação e o desenvolvimento de narrativas de jogos de RPG de mesa *online*.

Os objetivos específicos são:

- a) disponibilizar um ambiente *online* para os jogadores de RPG de mesa;
- b) disponibilizar modelos de cenários, personagens 3D e objetos para simular um tabuleiro de RPG;
- c) disponibilizar um mecanismo para que os usuários possam alterar os aspectos e características dos objetos, facilitando a customização do personagens;
- d) disponibilizar fichas para que os usuários controlem as estatísticas de seus personagens;
- e) disponibilizar um mecanismo de comunicação para dinamizar o contato entre usuários.

## 1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos. O primeiro capítulo é composto pela introdução e os objetivos do trabalho, além da apresentação de sua estrutura. O segundo capítulo apresenta a fundamentação teórica, esclarecendo os conceitos e considerações sobre o RPG e sobre o *Dungeons & Dragons* (D&D). Ainda nesse capítulo são descritos os trabalhos correlatos, que apresentam propostas similares ao ORC. No terceiro capítulo é explanado sobre o desenvolvimento do ORC através dos requisitos, especificação, implementação, operacionalidade e os resultados. Por fim, no quarto capítulo é apresentada a conclusão do trabalho, e também as possíveis extensões para o ORC.



## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo explorar os principais assuntos necessários para a realização deste trabalho. A seção 2.1 descreve o que é RPG, regras do D&D e a situação atual dos RPG digitais. E, na seção 2.2, são apresentados os trabalhos correlatos.

### 2.1 RPG DE MESA

RPG é a sigla, em inglês, para RolePlaying Game, ou jogos de interpretação de papéis, numa tradução livre. O jogo surgiu na metade dos anos setenta, nos EUA, como uma derivação dos chamados jogos de guerra, War Games em inglês, muito comuns à época, assim como por influência da literatura fantástica de autores como J.R.R. Tolkien. O jogo de RPG, muitas vezes chamado de RPG de mesa, para se diferenciar de dos jogos digitais, se constitui de um momento narrativo lúdico em que várias pessoas assumem o papel de uma personagem com vistas à criação de uma narrativa que apresenta um desafio a ser enfrentado pelos personagens dos jogadores, as pessoas que participam da narrativa. Um dos jogadores é chamado de mestre do jogo, ou simplesmente, mestre. O mestre é aquele que coordena a narrativa, o jogo em si, e todas as demais personagens que não as dos jogadores. O mestre é aquele que apresenta o desafio a ser enfrentado pelos jogadores, é o responsável pela criação inicial do cenário de campanha ou o “mundo” de jogo e pelo conhecimento aprofundado das regras e do sistema que serão utilizados (PRIETTO, 2013).

Em geral todo RPG de mesa consiste em criar personagens dentro das limitações impostas pelo sistema e interpretá-los dentro de uma aventura desenvolvida pelo narrador. Dessa maneira, cria-se uma história interativa onde ninguém sabe como a narrativa será desenvolvida, pois todos têm papel decisivo dentro da trama. Quando um personagem precisa realizar uma tarefa da qual não se tem certeza do resultado, deve-se rolar um dado específico com modificadores da situação e da habilidade que o personagem possui para realizar tal ato. A Figura 1 mostra um conjunto de dados.

Figura 1 – Conjunto de dados de RPG



Fonte: Kevin Owens Enterprises (2012).

Veras e Santos (2004) caracterizam o RPG como um gênero de jogo, um amplo universo lúdico que abriga dezenas de jogos diferentes - todos unidos por um elemento em

comum, a interpretação de uma personagem. Cada tipo de RPG funciona com uma mecânica de regras diferentes, isso é chamado de sistema de RPG. Existem diversos sistemas de jogo, sendo alguns exemplos: D20, Daemon, Storyteller, 3D&T e GURPS. O Dungeons & Dragons (D&D) é o mais difundido jogo de RPG, sendo o primeiro do gênero. Ele possui forma e estrutura definidas, que geraram narrativas coerentes e com inúmeras possibilidades.

Utilizando o sistema D20, o D&D, como também é conhecido é um jogo de aventuras de fantasia medieval, onde os jogadores assumem papéis de heróis, e por meio de uma história narrada pelo Mestre de jogo, deve enfrentar perigos e desafios para conquistar experiência além de outras recompensas (COLLINS et al., 2009). O D&D 4ª edição ou D&D 4.0, é uma reformulação das edições antigas do jogo, baseada nos pontos fortes de seus antecessores, como o mundo medieval repleto de monstros e cenários de campanha, tais como *Forgotten Realms* e *Dragonlance*. Com regras simplificadas, e com mais dinamismo, o Dungeons & Dragons da quarta edição se tornou muito boa para jogadores iniciantes do RPG (COLLINS et al., 2009).

Segundo Sales (2014), o D&D, surgiu no ano de 1974 nos Estados Unidos. Criado por Gary Gygax e Dave Arneson. Inicialmente era apenas um complemento para outro jogo de miniaturas chamado *Chainmail*, também criado por Gary Gygax. Os criadores do jogo eram jogadores de jogos de guerra, que era um passatempo comum na época. Nesses jogos de guerra cada jogador controlava um exército inteiro. Com o passar do tempo surgiu a ideia inicial de que jogassem com apenas um personagem por jogador, ao invés de tropas. Essa modalidade, foi o precursor do RPG que temos hoje.

O D&D é um jogo de imaginação que gira em todo de um mundo com criaturas fantásticas, magia aventura. Basta encontrar um local confortável onde possa espalhar seus livros, mapas e dados, se reunir com seus amigos e experimentar uma história em grupo. É como num filme de ação fantástica, onde seus personagens são as estrelas. A história se desdobra conforme os personagens tomam decisões e atitudes – o que acontece em seguida depende de você (WYATT, 2008).

A partida de RPG começa antes do jogo em si. Ele tem início quando os jogadores decidem entre uma variedade infinita de realidades, onde o jogo terá ambientação. Entre eles estão por exemplo, o velho oeste, o espaço futurista, a Europa medieval, um mundo com super-heróis, tudo é possível conforme afirma José Filho (2014).

O mestre de jogo, ou mestre, é o responsável por dar forma ao jogo. Ele é a pessoa que apresenta a aventura e os desafios a serem enfrentados pelos personagens. Além disso, o mestre possui funções distintas dentro do jogo, como criar aventuras, escolhendo quais desafios os jogadores devem enfrentar, narrando o transcorrer da história, controlando os monstros e vilões do jogo. O mestre também possui o título de árbitro de jogo, quando não

estiver claro o que deve acontecer, o mestre decide como aplicar as regras e determinar o rumo da história (COLLINS et al., 2009). A Figura 2 mostra um exemplo da ficha de um monstro.

Figura 2 – Ficha de um monstro

<b>Kobold Dragonwrought</b>		<b>Level 4 Lurker</b>
Small natural humanoid (reptile)		XP 175
HP 42; Bloodied 21		Initiative +10
AC 19; Fortitude 12; Reflex 16; Will 14		Perception +8
Speed 6, fly 8 Planar		Darkvision
<b>Traits</b>		
<b>Combat Advantage</b>		
The kobold dragonwrought deals an extra 1d4 damage on melee attacks against any target it has combat advantage against.		
<b>Trap Sense</b>		
The kobold gains a +2 bonus to all defenses against traps.		
<b>Standard Actions</b>		
① <b>Short Sword (weapon) • At-Will</b>		
Attack: +9 vs. AC Hit: 0 damage.		
† <b>Ataque do ar (weapon) • At-Will</b>		
Attack: +9 vs. AC; Voa na distância do movimento e faz um ataque contra qualquer adversário Hit: 1d6 + 5 e o alvo fica derrubado.		
← <b>Breath Weapon (radiant) • Recharge ☹ ☹ ☹</b>		
Attack: Close blast 3; +8 vs. Reflex Hit: 2d6 + 3 Dano radiante + 1d4 quando em vantagem de combate.		
<b>Minor Actions</b>		
<b>Shifty • At-Will</b>		
Effect: The kobold shifts 1 square as a minor action.		
<b>Skills</b> Acrobatics +11, Stealth +13, Thievery +13, Bluff +9		
Str 9 (+1)	Dex 18 (+6)	Wis 12 (+3)
Con 12 (+3)	Int 9 (+1)	Cha 14 (+4)
<b>Alignment</b> evil <b>Languages</b> Common, Draconic		
<b>Equipment</b> leather armor, short sword x2		

Fonte: Wyatt (2008).

Apesar de controlar os inimigos e vilões, o mestre não é um adversário dos jogadores. A função do mestre é oferecer a estrutura para que o grupo todo se divirta com a aventura. O papel do mestre é propor desafios, como combates, enigmas e outros tipos de tarefas (COLLINS et al., 2009).

Wyatt (2008) mostra que além de impor os desafios o mestre também é responsável por recompensar os jogadores, dando pontos de experiência e tesouros. Além de também ser o responsável por interpretar todas as outras criaturas da aventura, como animais ou até personagens que possam vir a ajudar os jogadores em suas missões.

O jogador é a pessoa que interpreta um personagem único dentro do jogo de RPG. Cada jogador precisa preencher um formulário sobre seu personagem, onde tudo que o

personagem sabe fazer está anotado. Essas habilidades geralmente são descritas através de um valor numérico, sendo que estes mesmos valores representam o quão fácil é realizar tal ação, ou ainda se esta ação é possível. Este formulário é conhecido como ficha do personagem (SALES, 2014). A Figura 3 apresenta uma ficha de personagem em branco.

Figura 3 – Ficha de personagem

Fonte: Collins et al. (2009).

Os jogadores precisam dispor de alguns materiais, para jogar D&D. Dentre eles, os mais importantes são os dados de RPG, a ficha de cada personagem, os livros de regra e uma miniatura que represente o personagem dentro do tabuleiro de jogo. A Figura 4 mostra algumas miniaturas de Dungeons & Dragons.

Figura 4 – Miniaturas de D&D



Fonte: Jones (2014).

Os elementos de RPG trabalham em conjunto para dar ao jogador e ao mestre a sensação de imersão no jogo. Um exemplo é ao realizar uma ação, que não um ataque, onde o jogador diz ao mestre o que deseja fazer. Em seguida o mestre fala sobre qual perícia se baseia a ação, e informa a dificuldade. O jogador ou o mestre joga o dado designado e soma com o modificador apresentada em sua ficha (COLLINS et al., 2009). Para que a ação seja visualizada de certa forma, são usadas matrizes quadriculadas juntamente com miniaturas. Assim os jogadores e o mestre conseguem ter dimensão do posicionamento em um campo de batalha, traçar estratégias baseadas no terreno, entre outras utilidades.

Os valores de atributo definem um modificador, que é somado nas jogadas de ataque, nos testes e nas defesas. Por exemplo, realizar um ataque básico corpo a corpo com um machado de batalha usa a força, portanto o modificador desse atributo é sempre somado nas jogadas de ataque e dano com essa arma. Se a força do personagem for igual a 17, esse bônus é de +3 (COLLINS et al., 2009).

Leon (2004), mostra que no início da década de 1980, os computadores começaram a ficar populares, com isso os usuais jogadores de RPG viram a possibilidade de poder criar jogos similares, como os jogos de guerra e tabuleiro que eram jogados por e-mail. O sucesso nesse tipo de jogo e o mercado de RPG ávido por material *online*, fez com surgisse o *Multi*

*User Dimension* (MUD), que incorporava a base do RPG, que é a interpretação de um único personagem durante o jogo. Com o número de usuários crescendo aos milhares o MUD ficou conhecido pelo termo de MMORPG. Existem RPGs eletrônicos que não são *online* e ainda assim são muito famosos, por assimilarem melhor a interpretação durante os jogos, exemplos desses jogos são *Elder Scrolls* e *Final Fantasy*, que introduzem a customização do personagem e a evolução dele ao concluir desafios. RPGs de fantasia geralmente possuem complexos sistemas de magias e técnicas de combate, possibilitando ao jogador manter uma estratégia para melhor evoluir seu personagem (MALLMANN, 2012). A Figura 5 mostra a tela do jogo *Elder Scrolls*.

Figura 5 – Jogo *Elder Scrolls*



Fonte: Mallmann (2012).

Segundo Machado et al. (2011), a indústria de jogos tem feito esforços para melhorar os gráficos de seus produtos objetivando aumentar o nível de imersão do jogador. Mas também é afirmado que a imersão não é completa se quando há por exemplo personagens sem reação ou sem interação durante o jogo. Os jogos de RPG tendem a ser uma exceção no que diz respeito a isso, pois no caso do MMORPGs, o jogador interage em sua maioria com outros personagens controlados por jogadores. Entretanto, o mesmo não acontece quando a interação é realizada para com um *non-player character* (NPC), pois este precisa ter uma programação de como agir.

Juul (2003) nos faz refletir sobre a diferença entre o RPG digital e o RPG de mesa tradicional. Destacando que o digital possui objetivos a serem realizados, mas ele sempre está

acompanhando jogo. No entanto Juul (2003) não considera o RPG tradicional como um jogo propriamente dito, e sim um “quase-jogo”, pois seus objetivos não são implícitos, e nem mesmo as regras precisam ser respeitadas se o mestre do jogo assim definir.

## 2.2 TRABALHOS CORRELATOS

Existem poucas ferramentas que proporcionam uma adaptação eletrônica para o RPG de mesa, devido à complexidade de cada jogo e partida. Entretanto, vale a pena citar algumas que conseguem fazer o que se propõem. Esta seção tem por finalidade apresentar três ferramentas relacionadas. São elas: *Tabletop Forge* (2012), Taulukko (2008) e *RRPG Firecast* (2009).

### 2.2.1 Tabletop Forge

O sistema *Tabletop Forge* (2012), produzido em 2012, inicialmente como aplicativo secundário para o Google Hangouts, era um aplicativo simples que incorporava os elementos básicos para a dita “mesa de RPG“. Os elementos gráficos eram limitados pelos aplicativos ou o usuário poderia importar alguma imagem previamente produzida. A dinâmica do programa deixava a cargo do narrador a edição da história e permitia total liberdade de criação da narrativa. O jogador poderia criar seu personagem da mesma maneira em que criaria em uma partida com papel e caneta, definindo características, 7 habilidades, atributos, etc.

Embora o sistema proporcionasse elementos gráficos para se usar, como NPC, personagens e mapas, ele não disponibilizava nenhuma ferramenta de edição ou criação, com exceção de mapas. O sistema de rolagem de dados era simples, mas continha tudo o que era necessário para que os usuários pudessem alterar o número de dados jogados, quantas faces eles deveriam ter e os modificadores que se aplicavam a cada rolagem.

Para se jogar eram criadas salas, que poderiam ser públicas ou privadas, além de dispor de um *chat* para comunicação e sessões onde os usuários poderiam fazer anotações, caso preciso. A Figura 6 mostra a tela principal do *Tabletop Forge*.

Figura 6 – Tela principal do *Tabletop Forge*

Fonte: Tabletop Forge (2012).

Em 2013 o *Tabletop Forge* (2012) deixou de ser um aplicativo do Google Hangouts e passou a funcionar em seu próprio site, onde a maioria dos elementos permaneceram, com a exceção da vídeo conferência. Entretanto, ainda é possível disponibilizar o jogo usando o *Hangouts*, pois a *Tabletop Forge* (2012) libera isso através de outro aplicativo, o Roll20, que é apenas uma ponte entre o site do programa e o *Hangouts*.

O sistema utiliza elementos gráficos simples e sem animações, e dispõe de *tokens* para definir atributos com intuito que esses fiquem visíveis ao jogador e ao narrador, para que os mesmos possam interagir de forma dinâmica e distinguir entre os personagens. O sistema ficou bem conhecido entre os jogadores, pois resolvia em parte o maior problema, a disponibilidade para se jogar, pois partidas de RPG costumam demorar de 2 a 4 horas variando de sistema, quantidade de jogadores e história proposta. Além de aproximar usuários de várias partes do mundo, o aplicativo facilita o encontro destes nos horários condizentes com suas disponibilidades. O site ainda possui facilidades para aquele jogador que não possui grupo, como fóruns e uma ferramenta de busca, onde o usuário pode definir o sistema que deseja jogar, ordenar por relevância ou ordem de data e ainda incluir palavras chave para a conveniência, entre outras opções.



### 2.2.2 Taulukko

O Taulukko (2008) é um site brasileiro que fornece um conjunto de funcionalidades básicas para se jogar RPG de mesa. Sua simplicidade talvez seja sua melhor qualidade, pois isso permite uma personalização da narrativa muito ampla. Ele possui pouco conteúdo gráfico, que se limita a um *Grid* (Tabuleiro), em que se pode preencher as células com uma cor da preferência do narrador, e avatares padrões, sem especificações gráficas.

O site ainda disponibiliza diversas fichas de vários sistemas, para que a narrativa se adapte a qualquer uma, além de uma ferramenta em que o próprio usuário pode desenvolver uma ficha personalizada. Possui um sistema de rolagem de dados singular, onde é disponibilizado um atalho para as rolagens mais comuns dos sistemas mais conhecidos.

Porém, também é permitido fazer a rolagem com quaisquer outros tipos de dados, alocando essas rolagens em atalhos para facilitar a personalização da narrativa. A Figura 7 exibe a tela principal do Taulukko.

Figura 7 – Tela principal do Taulukko



Fonte: Taulukko (2008).

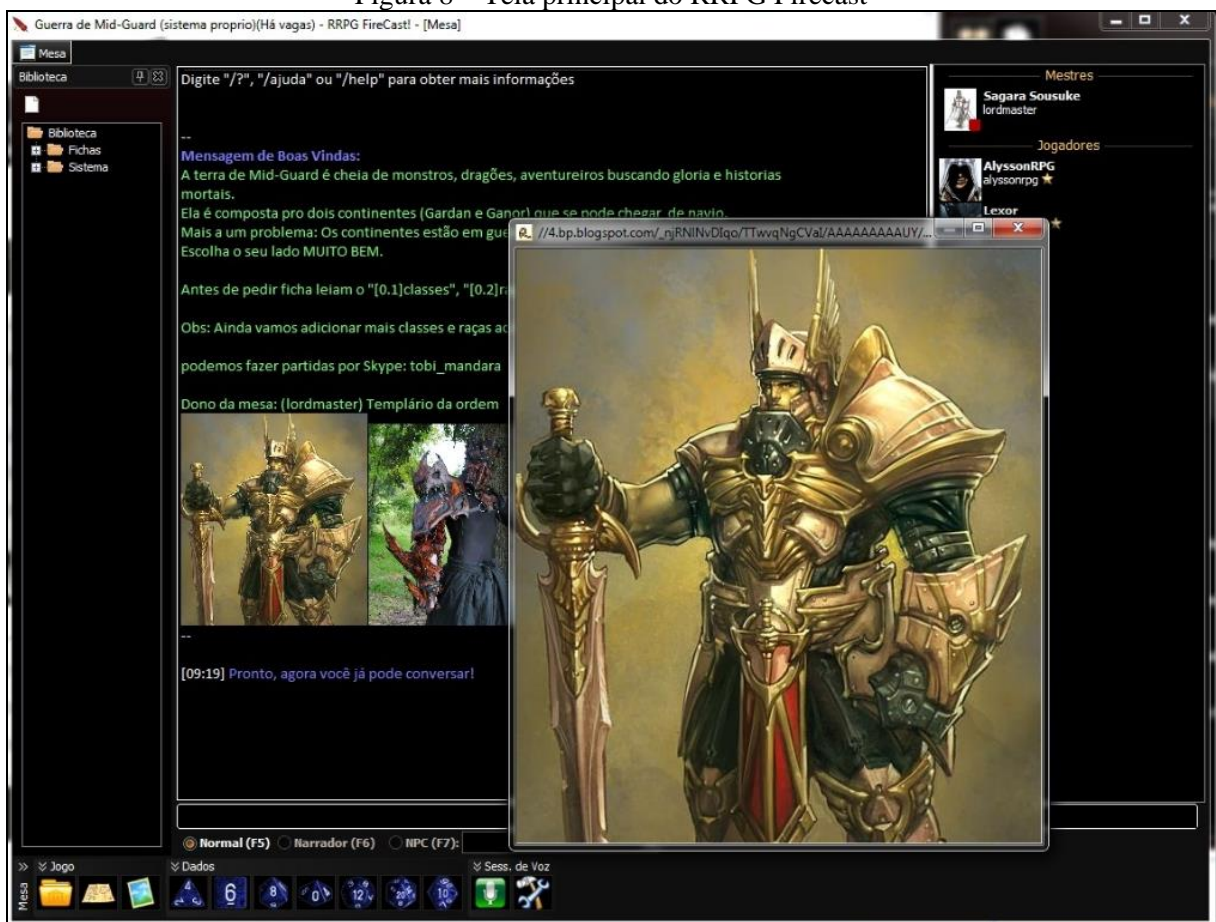
Para que se facilite o entendimento da história em relação aos comentários dispersos, foi disponibilizado dois tipos de *chat* para cada jogo, o *chat* convencional e o *off topic*, o qual serve para fazer comentários ou manter conversas dispersas ao jogo sem atrapalhar o dinamismo da narrativa. Além disso, o *chat* mantém um histórico, para que não se perca

detalhes que podem ter sido esquecidos, e um campo de anotações, importante para muitas narrativas. O site não possui *chat* de voz nativo, mas disponibiliza um programa com acesso ao seu servidor de *Team Speak*, que serve justamente para isso.

### 2.2.3 RRPg Firecast

O RRPg Firecast (2009) é um simulador de mesas de RPG brasileiro que conta com mais de 5 mil usuários. Com dinâmica clara e simples, suporta vários sistemas de RPG diferentes e fornece um *Software Development Kit* (SDK) para que seus usuários desenvolvam fichas personalizadas caso já não existam para *download*. Possui *chat* escrito e *chat* de voz, que são suportados na versão gratuita e paga. A dinâmica do *chat* é bem útil para a velocidade e entendimento do jogo. Mesmo no modo escrito, há opções para que o narrador possa falar pelo Non-Player Character (NPC), permitindo que os jogadores vejam claramente que o diálogo veio de um personagem e ainda possam distinguir de qual personagem. Além de varias funcionalidades que ajudam a diferenciar as ações, sussurros, falas, pensamentos, gritos, etc. A Figura 8 apresenta a tela principal do RRPg Firecast.

Figura 8 – Tela principal do RRPg Firecast



Fonte: RRPg Firecast (2009).

O programa não possui tabuleiro gráfico para usuários gratuitos. Todas as ações são feitas através da descrição e cada usuário preferencialmente deve manter um tabuleiro físico onde está jogando, para que possa acompanhar o movimento das ações. No caso de um usuário “*Gold*”, existem *plugins* liberados pelo site para que se possa usá-los no modo gráfico, que como quase toda ferramenta do gênero é bidimensional. O usuário pagante ainda tem a capacidade de exportar imagens, criar pastas para organização, criar cenas e itens para as narrativas. Um dos diferenciais do programa, é que permite que o usuário associe o alcance dos instrumentos usados por eles, como armas, poderes, lanternas, tochas, etc. Mas essas ações devem ser cadastradas antes que o personagem ou NPC possa usá-las.

#### 2.2.4 Comparativo entre trabalhos correlatos

O Quadro 1 apresenta um comparativo entre os trabalhos correlatos, com suas principais características.

Quadro 1 – Características dos trabalhos correlatos

Características/ Trabalhos	<i>Tabletop Forge</i> (2012)	Taulukko (2008)	RRPG <i>Firecast</i> (2009)
Gráficos 3D	Não	Não	Não
Modificação gráfica de personagens	Parcial	Não	Parcial
Dinâmica de luz	Sim	Não	Parcial
Facilitação de encontro de mesa	Sim	Sim	Não
Rolagem de dados	Sim	Sim	Sim
Chat <i>voip</i>	Sim	Sim	Não
Chat escrito	Sim	Sim	Sim

A partir do Quadro 1, pode-se perceber que as ferramentas conhecidas de RPG de mesa *online* partilham de muitos elementos em comum como o *chat* e a rolagem e dados. Também pode-se perceber, que nenhum dos trabalhos possui recursos 3D ou opções de personalizações gráficas dos personagens. Isso acontece devido à complexidade e numerosidade dos personagens e histórias narradas. Embora seja útil, não é absolutamente necessário para jogar RPG, sendo que este é um jogo de interpretação e não gráfico. Entretanto, esta característica proporcionaria melhor interpretação por parte dos jogadores.

### 3 DESENVOLVIMENTO

Este capítulo apresenta as etapas para o desenvolvimento do ORC. A seção 3.1 apresenta os principais requisitos do ORC. Na seção 3.2 está a especificação contendo os diagramas de casos de uso, de classes e de atividades. Na seção 3.3 está descrito a implementação das principais partes do ORC, assim como as ferramentas utilizadas. Por fim, na seção 3.4 estão apresentados os testes realizados e os resultados obtidos.

#### 3.1 LEVANTAMENTO DE INFORMAÇÕES

O ORC foi desenvolvido para atender os seguintes requisitos:

- a) permitir o cadastro de usuários (Requisito Funcional - RF);
- b) permitir ao mestre a criação de mapas (RF);
- c) permitir ao jogador implementar a ficha do personagem (RF);
- d) permitir ao jogador customizar a aparência dos personagens (RF);
- e) permitir ao mestre cadastrar o personagem do mestre (PdMs) (RF);
- f) permitir ao mestre a customização de PdMs (RF);
- g) dispor de um mecanismo comunicação por *chat* escrito e voip (RF);
- h) dispor de *prefabs* para a criação dos elementos necessários para o jogo (RF);
- i) permitir ao mestre controlar personagens do jogo (RF);
- j) permitir ao mestre restringir e recriar jogadas (RF);
- k) utilizar o banco de dados SQLite (Requisito Não Funcional - RNF);
- l) ser desenvolvido utilizando a linguagem C# (RNF);
- m) ser desenvolvido utilizando a ferramenta gráfica Unity (RNF);
- n) ser compatível com os sistemas operacionais Windows 7 ou superior (RNF);
- o) utilizar Photon PUN como servidor (RNF).

#### 3.2 ESPECIFICAÇÃO

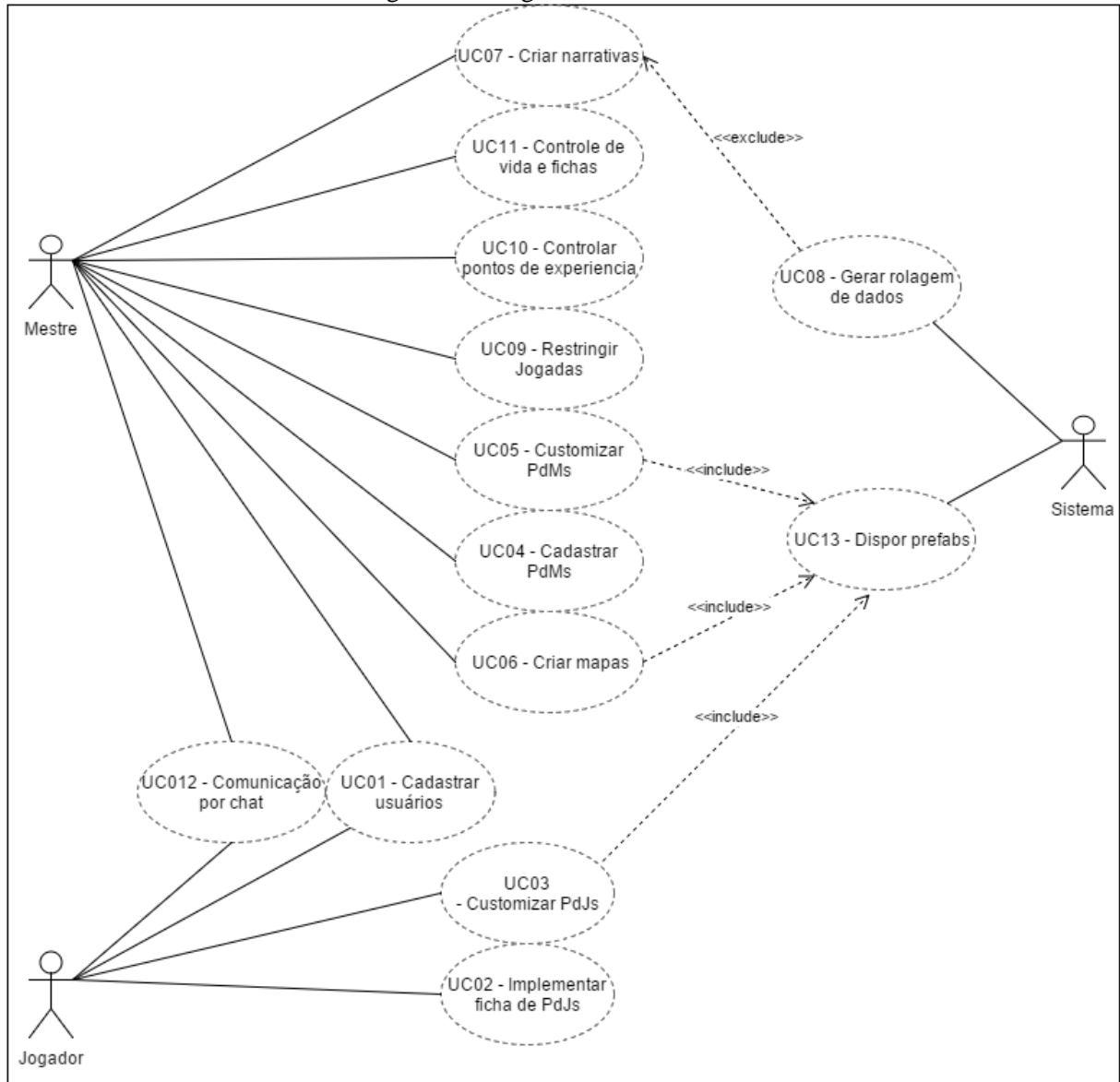
Na seção de especificação, o ORC é representado através dos diagramas da *Unified Modeling Languages* (UML). Foram elaborados os diagramas de casos de uso, de classes e de atividades.

##### 3.2.1 Diagrama de casos de uso

Esta seção apresenta o diagrama de casos de uso do ORC, composto de 13 casos de uso. O diagrama possui três atores distintos: o *mestre*, que tem acessos diferenciados dentre os outros usuários, o *jogador* que pode agir com a liberdade imposta pelo mestre, e o

sistema que age como ator secundário. A Figura 9 apresenta o diagrama de casos de uso da engine ORC.

Figura 9 – Diagrama de casos de uso



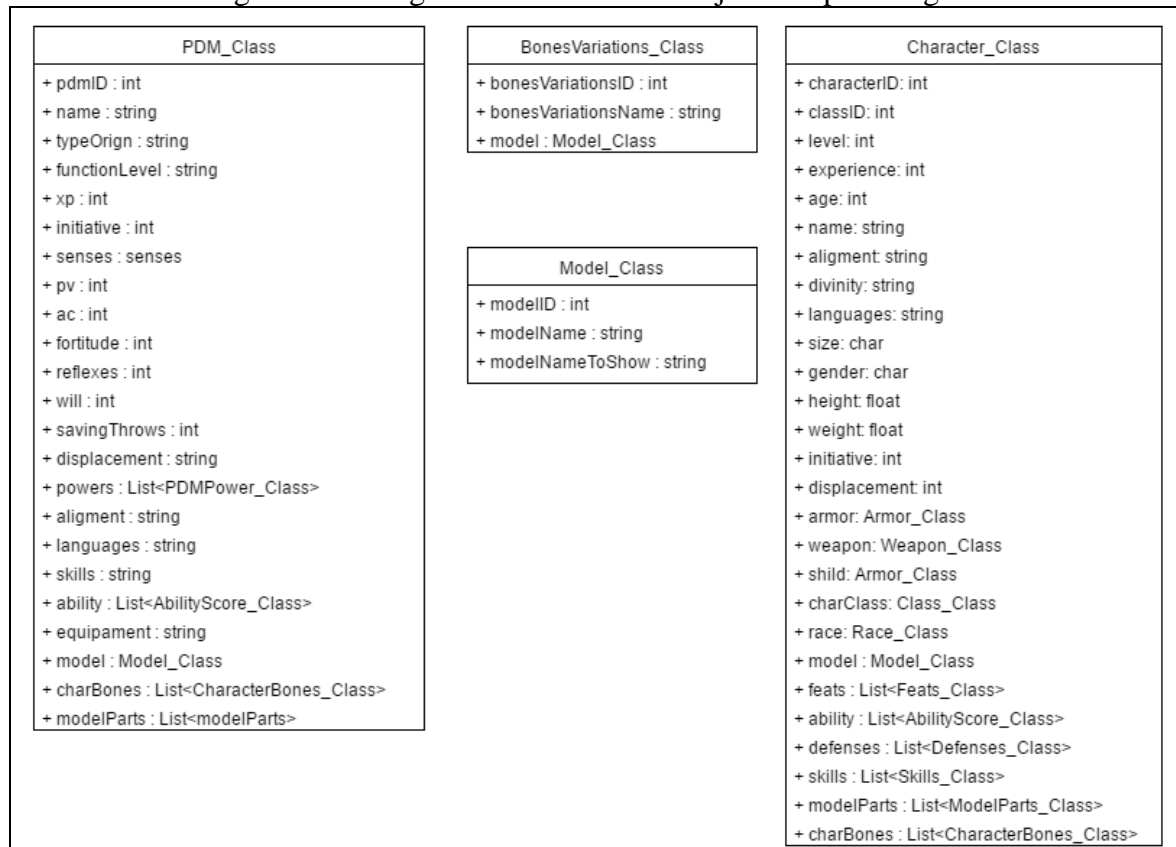
No diagrama de casos de uso apresentado, o mestre irá criar uma narrativa para que o jogador atue. No caso de uso UC01 - Cadastrar usuários, ambos os atores, mestre e jogador devem criar um usuário para que possam usar os módulos do ORC. No caso de uso UC02 - Implementar ficha de PdJs, o jogador implementa a ficha do personagem para usar durante o jogo. No caso de uso UC03 - Customizar PdJs, o jogador customiza o modelo 3D do personagem. No caso de uso UC04 - Cadastrar PdMs, o mestre cria e cadastra todos os PdMs para inseri-los durante o jogo. No caso de uso UC05 - Customizar PdMs, o mestre customiza o modelo 3D dos PdMs. No caso de uso UC06 - Criar mapas, o mestre cria o mapa para que se desenvolva o jogo. No caso de uso UC07 - Criar

narrativa, o mestre cria sua narrativa para que o jogo se desenvolva conforme ela avança. No caso de uso UC08 - Gerar rolagem e dados, durante as narrativas são geradas rolagem de dados, que são usadas em todos os jogos de RPG. No caso de uso UC09 - Restringir jogadas, o mestre pode controlar todos os personagens, voltando posições e recriando ações. No caso de uso UC10 - Controlar pontos de experiência, o mestre controla os pontos de experiência. UC11 - Controle de vida e fichas, o mestre controla as fichas de todos os personagens, além dos pontos de vida. UC12 - Comunicação por chat, ambos os personagens, mestres e jogadores, comunicam-se por *chat* escrito e de voz. No caso de uso UC13 - Dispor prefabs, para cada modelo 3D, o ORC dispõe modelos como *prefabs*. O detalhamento dos casos de uso do ORC encontra-se no Apêndice A.

### 3.2.2 Diagrama de classes

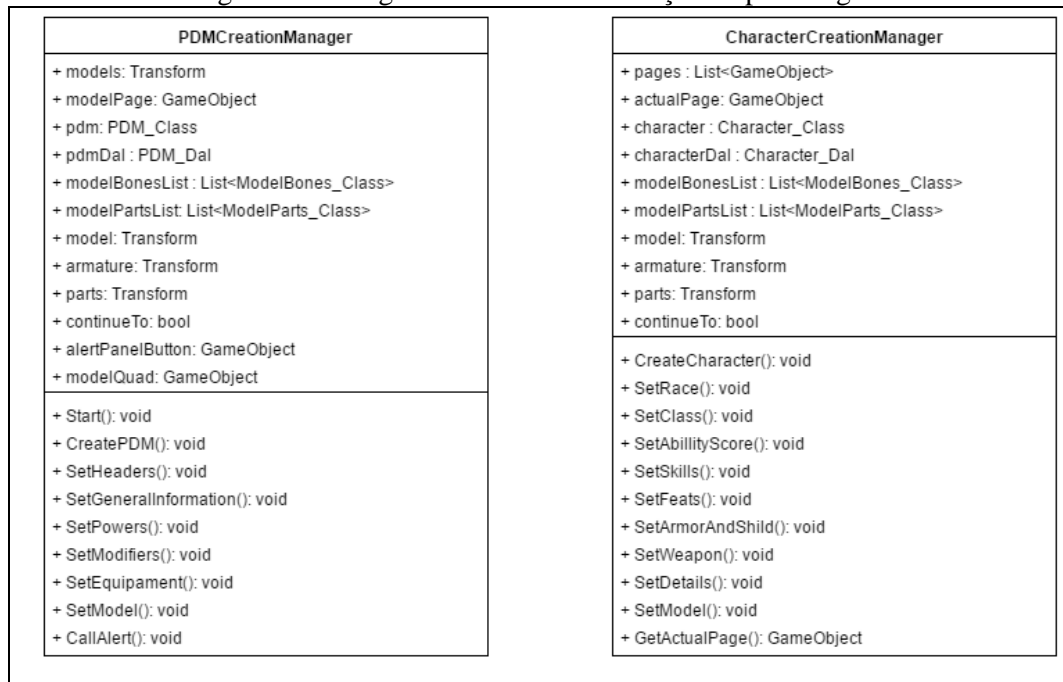
Nesta seção constam os diagramas das principais classes do protótipo desenvolvido. A Figura 10 mostra as classes `PDM_Class`, `BonesVariations_Class`, `Model_Class` e `Character_Class` que são referentes aos objetos que representam os personagens e suas formas 3D.

Figura 10 – Diagrama de classes dos objetos de personagens



As classes apresentadas acima são referentes aos objetos de personagens. As classes `BonesVariations_Class` e `Model_Class` são responsáveis por manter a referência durante a alteração das malhas 3D. A criação dos personagens, assim como a personalização de seus modelos são realizados nas classes representadas na Figura 11.

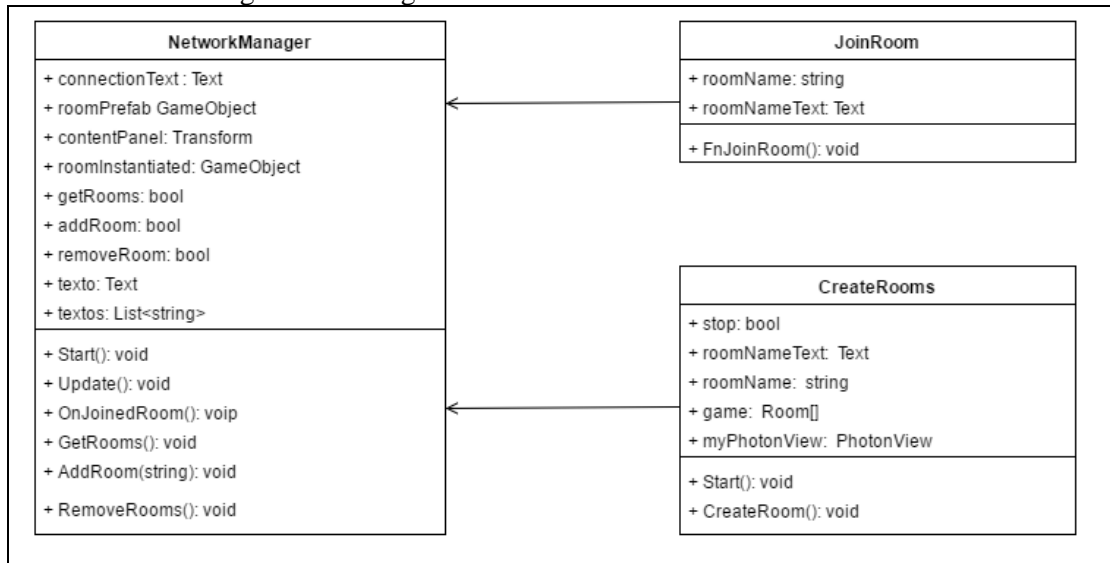
Figura 11 – Diagrama de classes da criação de personagens



A classe `PDMCreationManager` é responsável por cadastrar e definir como o modelo 3D de um novo personagem para que o mestre interprete em jogo. Os métodos `CreatePDM` e `SetModel` podem ser considerados os mais importantes da classe. O primeiro é o responsável por controlar a criação do PdM, ordenando os métodos e validando-os. Se houver um erro no cadastro, o cadastro é interrompido e uma mensagem de erro é mostrada ao usuário. Já o método `SetModel` é o responsável tanto pela escolha do modelo 3D, quando pela modificação de sua malha, assim como a adição de partes.

A classe `CharacterCrationManager` realiza a gravação e customização do personagem do jogador. As duas classes acima apresentadas possuem instancias que se comunicam com o banco de dados. Vale ressaltar que o método `CreateCharacter`, que a sequência de verificação dos passos de criação e em seguida efetua o cadastro. Além do método, outro método importante é o `SetModel`, que possui finalidade similar ao de PdM. Entretanto, não permite a seleção de um outro modelo, que por sua vez, é definido pela raça do personagem. Na Figura 12 são apresentadas as principais classes para conexão ao Photon.

Figura 12 – Diagrama de classes da conexão com o Photon



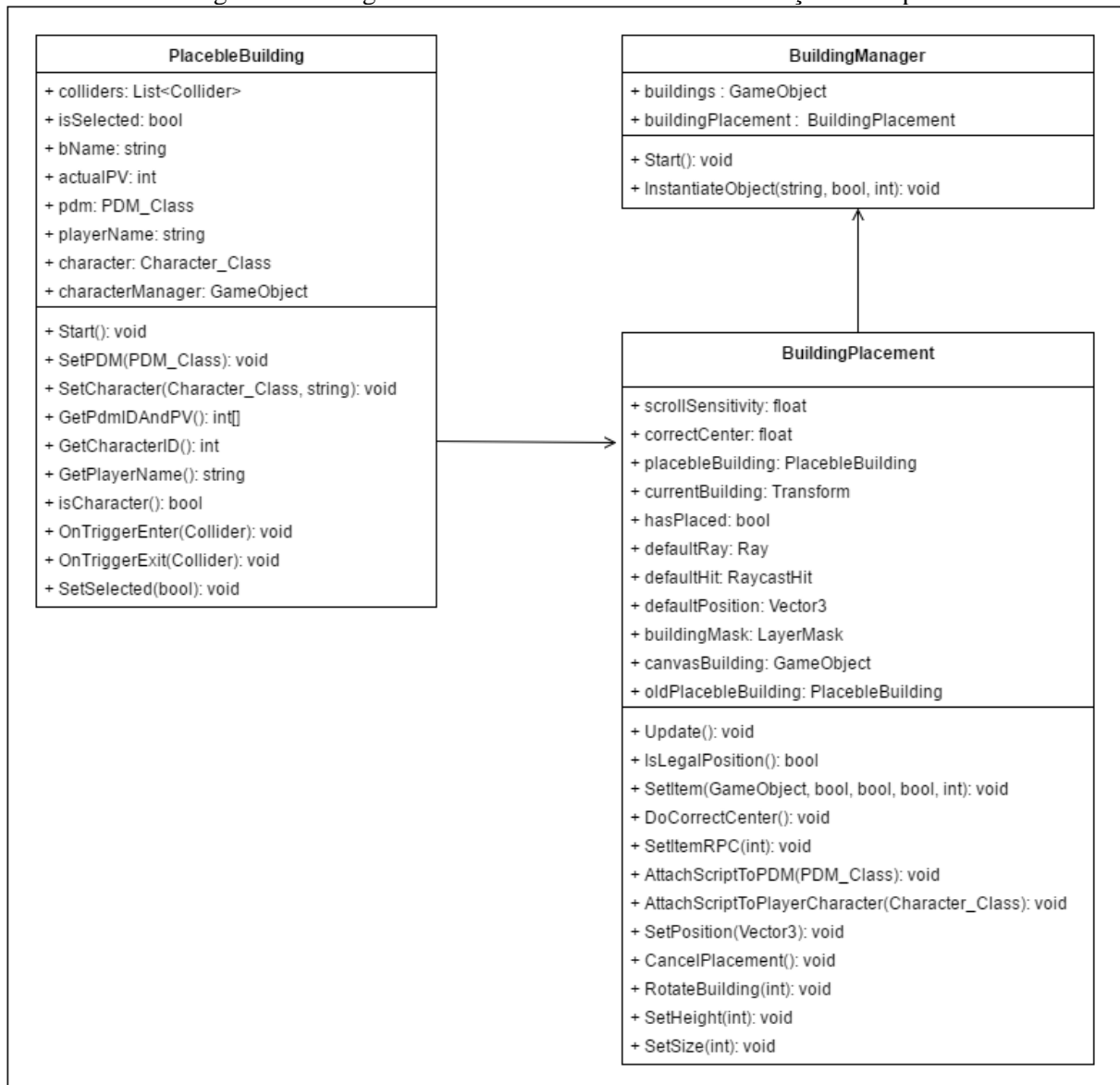
A classe `NetworkManager` mantém a lista de salas atualizadas, estabelece a conexão com o *lobby* realizada através do método `Start`. Para atualizar as salas disponíveis, o método `Update` busca constantemente novas salas, contanto que o usuário esteja conectado ao *lobby*. Ainda na classe `NetworkManager` existe o método `OnJoinedRoom`, ele é responsável por direcionar os usuários para a tela de jogo ao entrar na sala.

A classe `CreateRooms` recebe as salas criadas através do usuário em questão e envia a sala criada para o *lobby*. Nesta classe vale citar o método `CreateRoom`, que informa ao servidor que existe uma nova sala de jogo, além de enviar as propriedades e as informações sobre o usuário que a criou.

Por fim, a classe `JoinRoom` estabelece a sala criada como sala de jogo através do método `FnJoinRoom`. A Figura 13 apresenta algumas das principais classes para criação de mapas e controle de cenário durante o jogo.



Figura 13 – Diagrama de classes sobre o controle e criação de mapas



A classe `PlacebleBuilding` é responsável por simular o comportamento de cada objeto dentro do jogo simulando as funções das peças de RPG, sendo que cada objeto possui uma instancia da classe. Os métodos mais relevantes desta classe são `OnTriggerEnter` e `OnTriggerExit`, que mostram se a posição de cada objeto é legal dentro do tabuleiro. Não permitindo que as malhas de objetos sejam posicionadas de forma a não ocuparem o mesmo local dentro do tabuleiro.

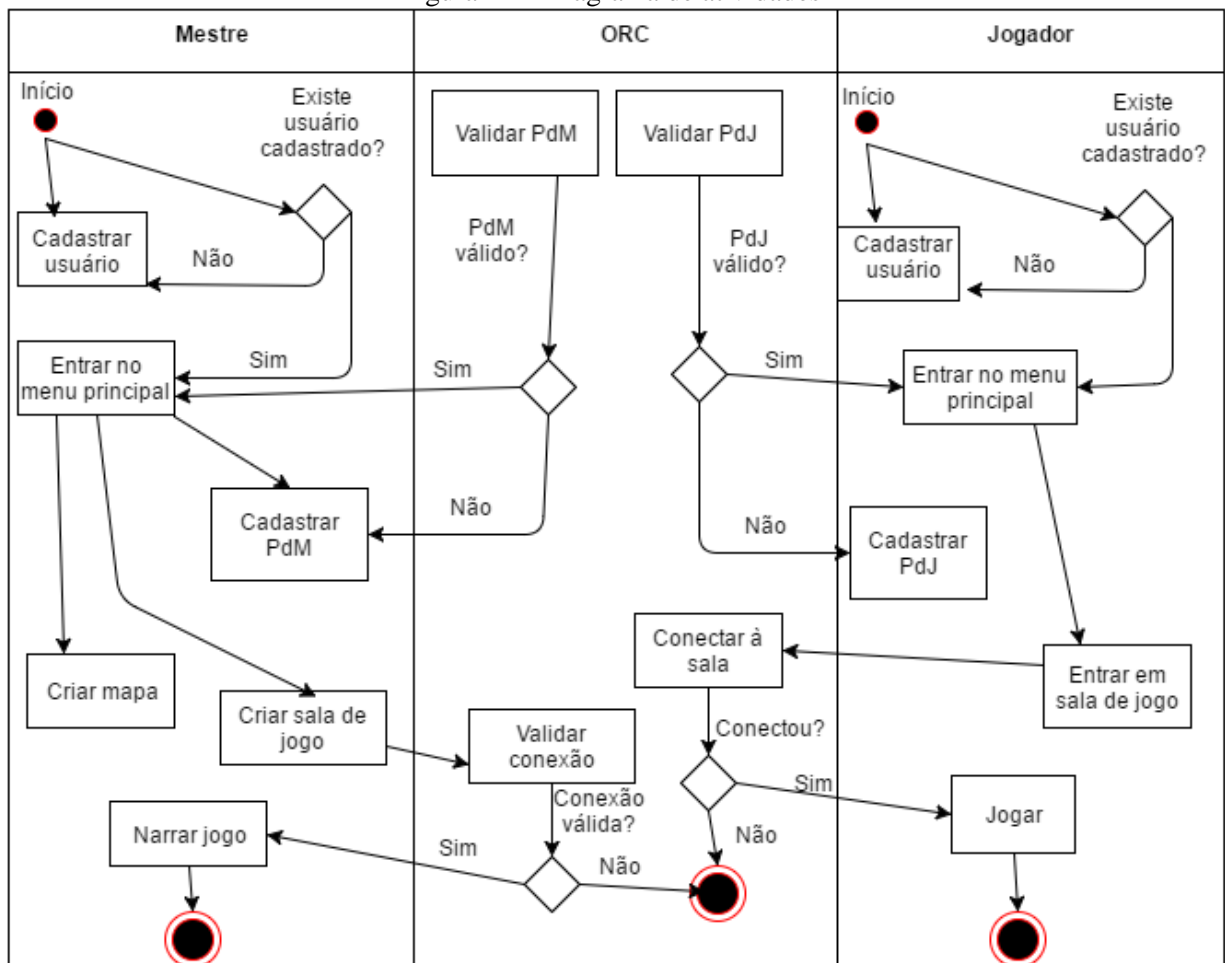
A classe `BuildingPlacement` é a classe mais importante do módulo em relação à criação de mapas, é ela que recebe, trata e valida todas as ações do usuário no que diz respeito aos itens de jogo através do método `SetItemRPC`. Esta classe também é responsável por vincular as fichas de personagens aos seus respectivos modelos, através dos métodos `AttachScriptToPlayerCharacter` e `AttachScriptToPDM`.

Já a classe `BuildingManager` é a classe que controla quais *prefabs* são passíveis de serem usados como objetos de jogo. O método `InstantiateObject`, informa qual objeto é clicado, além da ação requerida para ele como edição, clonagem e exclusão.

### 3.2.3 Diagrama de atividades

A Figura 14 apresenta o diagrama de atividades com as etapas que devem ser executadas pelo mestre, ORC e jogador para poderem jogar uma partida de RPG.

Figura 14 – Diagrama de atividades



Para o mestre, inicialmente deverá ser feito um cadastro de usuário. Para realizar o cadastro, é necessário apenas o nome do usuário, que não pode ser igual a um já existente e não pode estar em branco. Em seguida o mestre é direcionado ao menu principal do ORC. A partir dele, o mestre pode cadastrar um PdM, criar um mapa ou criar uma sala de jogo. Se cadastrar um PdM, ele deve informar o nome, nível, atributos, defesas perícias, equipamentos, poderes, define o modelo 3D e em seguida customiza-o. O PdM é enviado à validação. Na validação de PdM, o ORC verifica se todos os campos estão preenchidos corretamente, caso

estejam, o mestre é direcionado ao menu, caso contrário, o mestre pode então, corrigir o que está em desacordo com o ORC.

Criando um mapa, o mestre pode planejar como será o jogo, primeiramente decidindo as medidas para o mapa, em seguida posicionando os itens da maneira que desejar. O mestre também pode modificar a rotação, a escala e a posição vertical dos objetos se assim desejar.

Se o mestre criar uma sala, a conexão é validada, se houver resposta, a sala é aberta para que o mestre possa narrar o jogo. Durante a narrativa o jogo, o mestre dispõe dos personagens que ele criou, além dos personagens criados por seus jogadores. A narração também conta com um sistema de rolagem de dados e de comunicação, tanto escrita quando por voz. Dentro do jogo o mestre também tem liberdade para editar o mapa e controlar as fichas dos personagens.

Para o jogador, a criação de usuários se repete, sendo apenas necessário um nome único que não esteja em branco. Assim que o usuário entra no jogo como jogador, ele é enviado ao menu principal do ORC, que é o mesmo para os dois tipos de usuário (mestre e jogador).

A partir do menu, o jogador pode cadastrar um PdJ, se o fizer, ele então é enviado para a tela de criação, onde deve escolher como será seu personagem, passando pelas etapas descritas no livro do jogador. Após escolher a raça, classe, poderes, talentos, perícias treinas, valores de atributo, itens e personalizar o modelo o personagem é enviado para a validação. O sistema então verifica se o personagem segue as regras de criação. Se for válido, o jogador é direcionado ao menu principal, caso contrário, ele pode rever o cadastro e se ajustar conforme as exigências do ORC.

Por fim, o jogador se conecta a sala de jogo que achar melhor. Se for estabelecida uma conexão, o jogo será iniciado. Durante o jogo o papel do jogador é interpretar seu personagem, podendo modificar as informações da ficha, se o mestre permitir, além de controlar a posição do modelo que representa seu personagem. O jogador também pode realizar testes de dados e comunicar-se através de um dos *chats*.

### 3.3 IMPLEMENTAÇÃO

Nesta seção são descritas as técnicas e ferramentas utilizadas para o desenvolvimento do ORC, a descrição do desenvolvimento e a operacionalidade do sistema.

### 3.3.1 Técnicas e ferramentas utilizadas

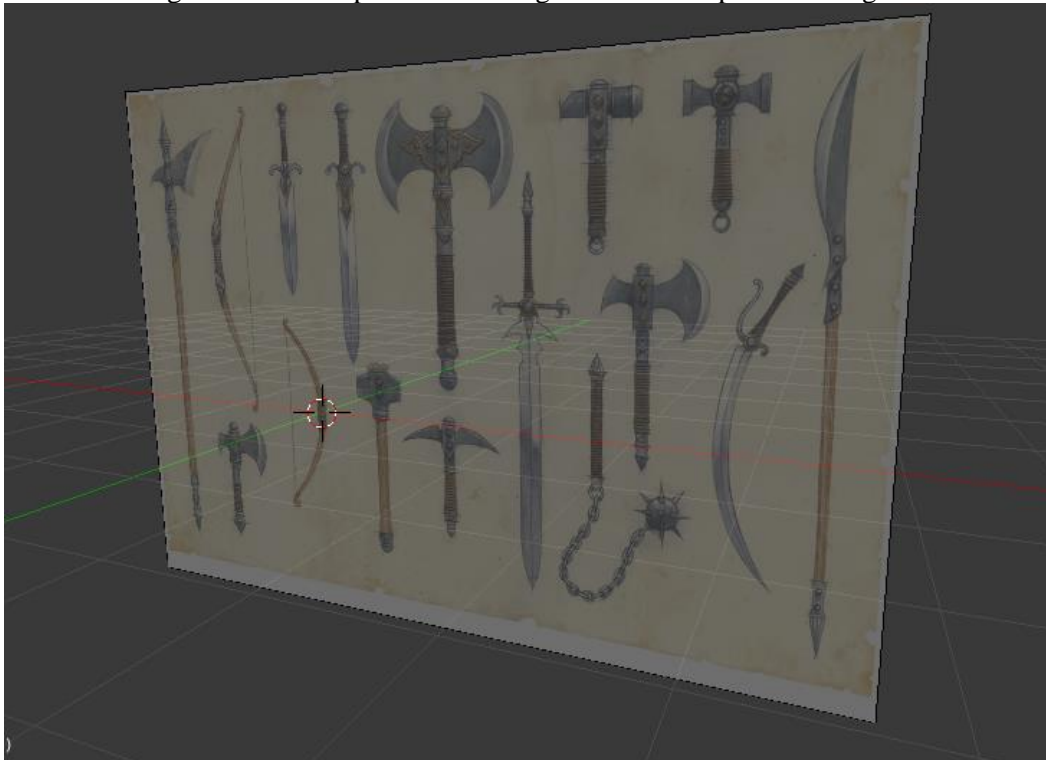
O ORC foi desenvolvido utilizando a linguagem C# da plataforma Microsoft .NET com o auxílio da *Integrated Development Environment* (IDE) Visual Studio 2013, integrado com a *game engine* Unity3D na versão 5.3. Também foi utilizada a biblioteca Photon PUN Plus. Para os modelos 3D e seus respectivos *rigs*, foi utilizada a ferramenta 3D Blender. Nas próximas seções são detalhadas as principais implementações e a operacionalidade das funcionalidades presentes no ORC.

### 3.3.2 Modelagem 3D

A modelagem 3D do ORC foi feita para que o usuário consiga customizar a malha do modelo. Os modelos referentes aos objetos em geral não seguiram o mesmo conceito. Entretanto, os que representam personagens passaram por algumas etapas especiais, sendo elas: a busca de referências, a modelagem, a criação do esqueleto de *rigging*, o *rigging* e por último a criação de matérias e texturização.

A primeira etapa para a modelagem é a busca de referências, onde se faz necessário uma ilustração, foto ou modelo base. Essa imagem é importada para o editor 3D como textura de um objeto, adicionando uma transparência. Dessa forma, é possível editar o modelo tendo como base a imagem atrás dele. O intuito da modelagem não é ser exatamente igual à referência na maioria dos casos, mas sim como guia de medidas que podem facilitar ou otimizar a modelagem. A Figura 15 exemplifica uma imagem referência para a modelagem 3D.

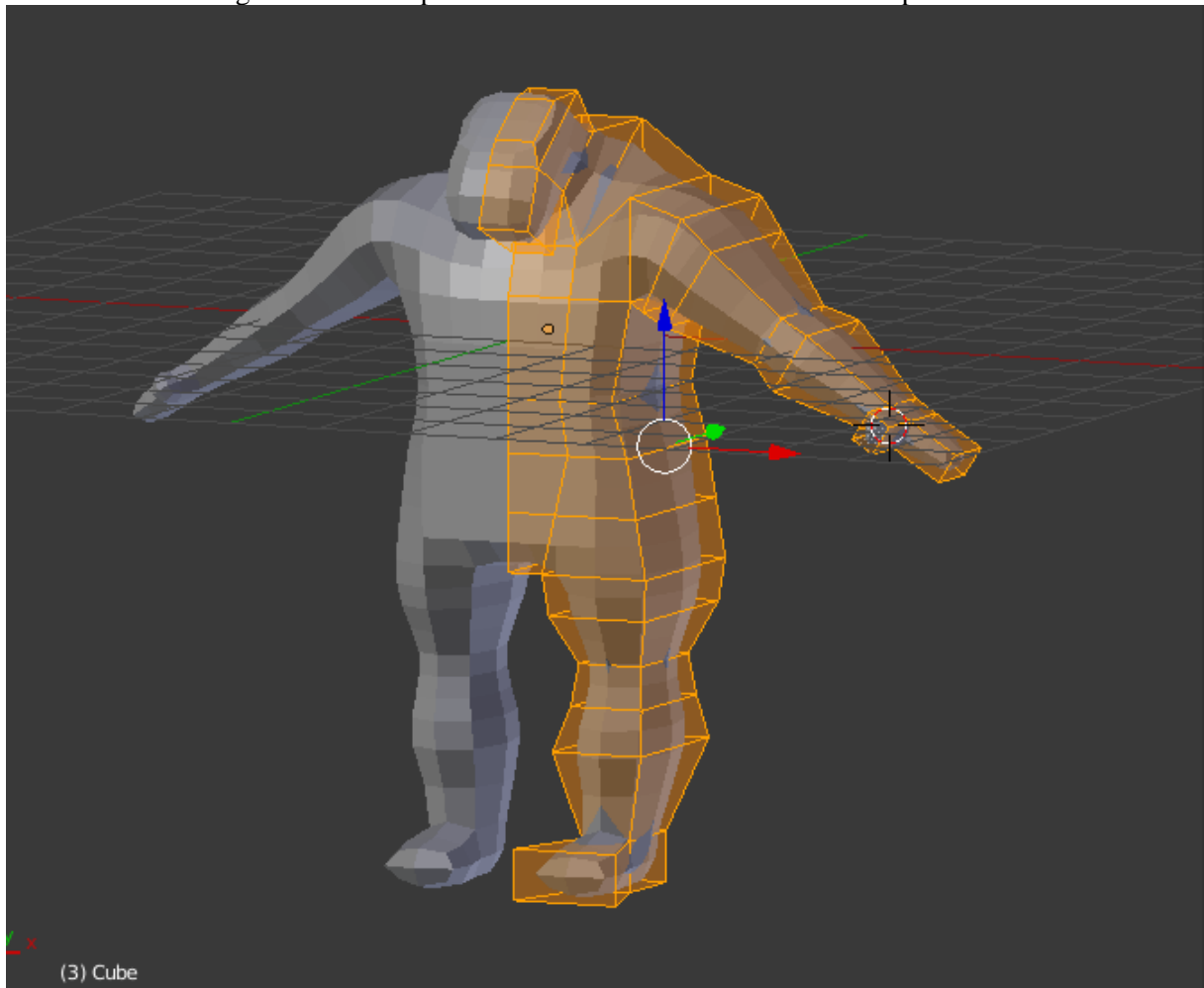
Figura 15 – Exemplo de uma imagem referência para modelagem



A Figura 15 demonstra uma imagem referência para a modelagem de armas. Nota-se que ela possui transparência para que fique em segundo plano e ainda possa ser utilizado atrás de um modelo.

A segunda etapa é a modelagem em si. Tendo como base a imagem referência, é escolhido um modelo de polígonos condizendo com a forma desejada, atentando-se também a necessidade economizar polígonos em formas mais complexas. Para as armas, geralmente se começa com um cilindro, pois todas elas possuem algum tipo de cabo. Para personagens o ideal é começar com um cubo, por se tratarem de modelos mais complexos. Objetos com simetria nesta etapa, são cortados ao meio por meio de um *loop* de arestas. Em seguida, as faces direitas são deletadas para que se trabalhe apenas com um lado e depois seja feita a simetria. São feitas extrusões a partir do objeto inicial circundando a forma básica do modelo. As extrusões são feitas em visão ortográfica, começando com a imagem frontal, e em seguida a lateral esquerda. Com o modelo básico feito, se aplica os modificadores de malha, primeiramente o modificador para espelhar a malha no eixo, que no caso de personagens é o eixo X. Em seguida, o modificador de subdivisão para visualizar o objeto com suavizações, conforme apresenta a Figura 16.

Figura 16 – Exemplo de malha básica com modificadores aplicados



Esta imagem representa o modelo básico para criação de um personagem, feito de um cubo, no qual foram aplicadas extrusões até adquirir o aspecto primário desejado. A partir disso, a malha é aprimorada até possuir o aspecto desejado, é preciso tomar as precauções de não deixar faces com mais de quatro vértices, sendo o ideal de três. Para a próxima etapa é preciso manter os vértices e arestas da malha condizentes com a anatomia do personagem. Se for o caso, seguindo as linhas musculares e de expressões. A Figura 17 apresenta um modelo já detalhado.



ossos que dependem da posição de outros, mas que não possuem conexão são unidos, fazendo que a distância entre os mesmos permaneça sempre a mesma. Cada elemento do esqueleto recebe um nome condizente, e se for o caso, seguido de um sinal do lado em que se encontra (.L para os que se encontram a esquerda). Por fim, os elementos da armadura que possuem um sinal de orientação são espelhados. A Figura 18 mostra um esqueleto completo.

Figura 18 – Exemplo de modelo com esqueleto

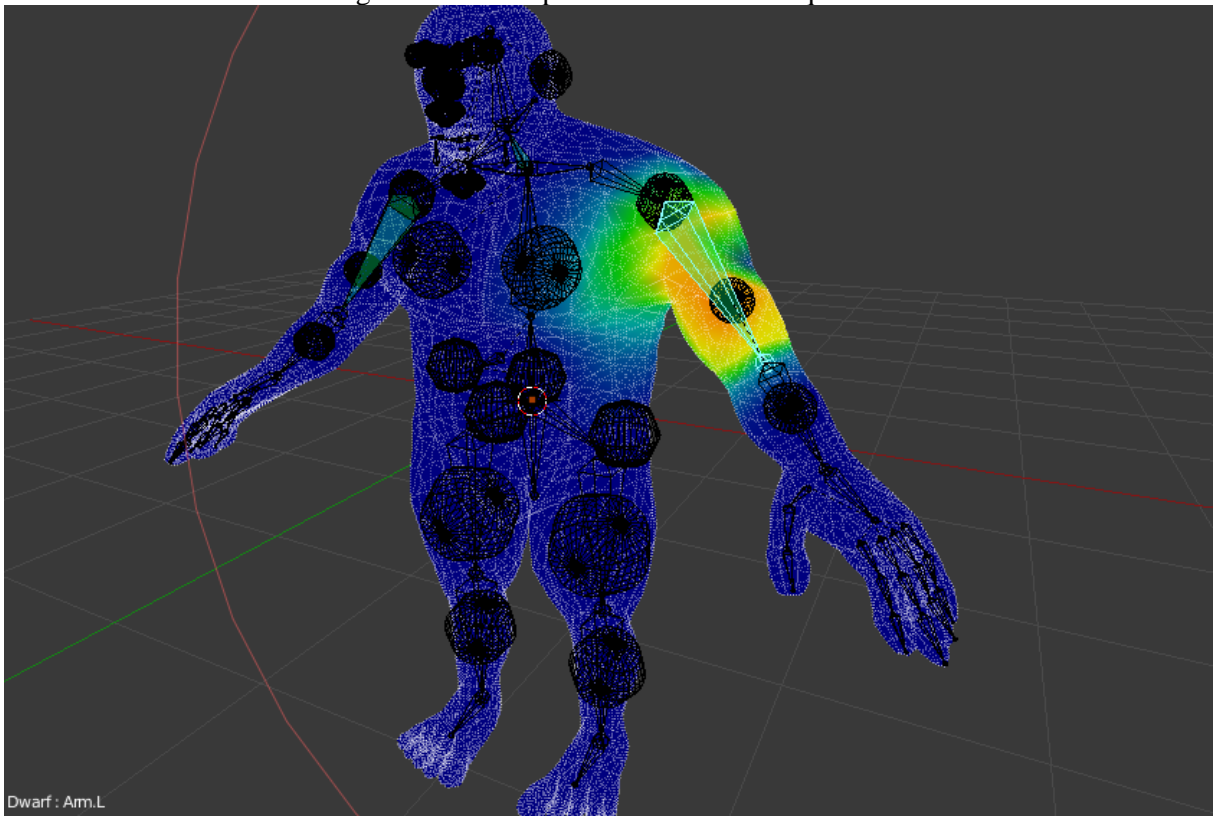


A Figura 18 mostra um esqueleto pronto para *rig*. Os ossos piramidais representam os de movimento, e os circulares servem para edição de características. Os que apresentam cor esverdeada são ossos controladores, que podem bloquear a escala de ossos específicos, impedindo que esses mudem o tamanho dos ossos do tronco.

A quarta etapa é a mais importante, pois nela é feita o *rigging*. O objeto recebe um modificador de armadura, tendo o esqueleto preparado como referência. A partir disso, o modelo pertence ao esqueleto, a malha deve passar por uma pintura de interferência. A pintura é feita de acordo com cada osso, os pontos na malha são pintados de acordo com a interferência que cada osso tem sobre ela, quanto maior a influência mais avermelhada é a cor do *rigging*. A Figura 19 exemplifica um modelo com *rigging* pronto.



Figura 19 – Exemplo de modelo com esqueleto



A Figura 19 retrata o modelo anterior utilizando sua pintura de *rigging*. Com o osso do braço selecionado, é possível ver o quanto ele interfere na malha naquele ponto, quanto mais avermelhada a malha, maior a interferência do osso.

A quinta etapa é a criação de materiais e texturização. A malha é preenchida com os materiais específicos para cada parte, um ser humano, por exemplo, tem a cor da pele e a cor interna da boca na mesma malha. A partir disso, é escolhida uma textura, de preferência em tons brancos, cinzas e pretos, fazendo com que a escolha do material dentro do editor de jogo tenha o mesmo efeito, independentemente da cor escolhida. É feito mapeamento de *Unpleasant Visual* (UV) para que a textura possa ser preenchida corretamente quando o modelo for exportado. Ao final desta etapa, o modelo está pronto para ser exportado.

### 3.3.3 Implementação do ORC

O ORC possui três etapas básicas, são elas, a etapa de criação de personagens, a criação de mapas e a etapa de jogo. O uso de cada etapa depende do tipo do usuário, sendo que existe o usuário jogador e o usuário mestre. Para melhor elucidar o funcionamento de cada etapa do sistema foi elaborado um fluxo de atividades na **Erro! Fonte de referência não encontrada.**

Quadro 2 – Atividades por módulos

Usuário	Criação de personagem	Criação de mapas	Jogo
Jogador	Escolha de raça Escolha de classe Definição de atributos Treinamento de perícias Escolha de poderes Escolha de talentos Informações básicas Personalização de modelo	Não tem acesso	Movimentação do personagem Rotação do personagem Comunicação por <i>chat</i> escrito Comunicação por <i>chat</i> voip Jogadas de dados Testes de ataque Testes de perícias
Mestre	Informações básicas Definindo defesas Definição de vida Definição de atributos Definição de modificadores Definição de perícias Definição de ataques e manobras Personalização de modelo	Criação de objetos Definição do tamanho do mapa Controle da posição dos objetos Controle da rotação dos objetos Controle do tamanho dos objetos	Movimentação de todos os objetos Rotação de todos os objetos Escala de todos os objetos Comunicação por <i>chat</i> escrito Comunicação por <i>chat</i> voip Jogadas de dados Narração de campanha Testes de perícias Testes de ataque

O quadro acima exemplifica o que cada usuário (jogador e mestre) pode fazer, assim como, diferencia os papéis a serem executados antes e durante o jogo. Nas próximas seções são detalhadas as funcionalidades do ORC.

### 3.3.4 Seleção de usuário

Na primeira tela de jogo, ilustrada pela Figura 20, o usuário deve escolher ou criar um usuário *offline*. Este usuário serve para que as criações do jogador ou mestre sejam gravadas mesmo que feitas sem uma conexão com a internet.

Figura 20 – Lista de usuários *offline*

A Figura 20 exibe uma lista com os usuários cadastrados. Se o usuário não possuir cadastro, basta inserir um nome único na seção de criação de usuário. Ao entrar no ORC, o jogador é então direcionado para a tela de menu principal, demonstrada na Figura 21.

Figura 21 – Menu principal



No menu principal, o usuário pode entrar na sala de servidores para jogar, criar um personagem PdJ (Personagem do Jogador), criar um personagem PdM (Personagem do Mestre), criar mapa, trocar de usuário ou sair da aplicação.

### 3.3.5 Criação de personagens

Para a criação de personagens os passos a serem seguidos foram extraídos de Collins et al. (2009) para o usuário jogador. Descrito em 9 etapas dentro do ORC, a primeira delas é a escolha da raça, onde o jogador pode escolher entre anões, humanos ou elfos. Cada raça tem suas peculiaridades que os caracterizam dentro do universo de D&D. A segunda etapa é a escolha da classe, as opções são entre guerreiro, ladino ou mago. A classe define como o jogador será e quais serão suas habilidades principais, assim como outros aspectos básicos. A terceira etapa é a distribuição de atributos, na qual são escolhidos os valores para representarem a força, constituição, destreza, inteligência, sabedoria e carisma. Os atributos interferem em quase todos os testes durante o jogo, além de influenciarem em outros campos, como por exemplo, o valor de constituição é somado a vida do personagem. A quarta etapa é a escolha de perícias treinadas. As perícias são calculadas com base nos atributos, a classe define quais perícias podem ser treinadas, quando treinada seu valor é acrescido de cinco pontos, conforme exemplifica a Figura 22.

Figura 22 – Definição de Perícias

Novo Personagem

Perícias	Bônus	Mod +1/2 nvl	TRND (+5)	Armor Penalty	Misc	Perícias	Bônus	Mod +1/2 nvl	TRND (+5)	Armor Penalty	Misc
<input type="checkbox"/> Acrobacia - DES	1	1	0	0	0	<input type="checkbox"/> Percepção - SAB	1	1	0	n/a	0
<input type="checkbox"/> Arcanismo - INT	1	1	0	n/a	0	<input type="checkbox"/> Religião - INT	1	1	0	n/a	0
<input checked="" type="checkbox"/> Atletismo - FOR	8	3	5	0	0	<input type="checkbox"/> Socorro - SAB	1	1	0	n/a	0
<input type="checkbox"/> Blefe - CAR	2	2	0	n/a	0	<input checked="" type="checkbox"/> Tolerância - CON	7	2	5	0	0
<input type="checkbox"/> Diplomacia - CAR	2	2	0	n/a	0						
<input type="checkbox"/> Exploração - SAB	1	1	0	n/a	0						
<input type="checkbox"/> Furtividade - DES	1	1	0	0	0						
<input type="checkbox"/> História - INT	1	1	0	n/a	0						
<input checked="" type="checkbox"/> Intimidação - CAR	7	2	5	n/a	0						
<input type="checkbox"/> Intuição - SAB	1	1	0	n/a	0						
<input type="checkbox"/> Ladinagem - DES	1	1	0	0	0						
<input type="checkbox"/> Manha - CAR	2	2	0	n/a	0						
<input type="checkbox"/> Natureza - SAB	1	1	0	n/a	0						

Perícias escolhidas: 3/4

A Figura 22 mostra um personagem em construção. As perícias marcadas já foram treinadas, ainda resta uma a ser treinada, podendo ser manha ou socorro. As perícias possíveis para este personagem são definidas pela classe guerreiro.

A quinta etapa é a escolha de talentos. Cada talento possui um benefício para o personagem. Em geral esse benefício é aplicado durante o jogo pelo mestre. A sexta etapa é a escolha de poderes, que são definidos pela classe do personagem. A sétima etapa é a compra de equipamentos. O jogador escolhe armaduras, armas e equipamento de aventura. A oitava etapa é a finalização da ficha em si, completando as informações da mesma, como nome, peso, altura, idade, gênero, divindade e idioma. A nona etapa apresenta o modelo do personagem para edição. O jogador pode editar a representação do personagem, alterando as características físicas, ou adicionando partes ao modelo, como cabelo e barba.

O preparo da edição da malha acontece na classe `CharacterModelController`, através dos métodos que controlam os ossos e as partes do personagem. Estes métodos são descritos no Quadro 3.

Quadro 3 – Carregamento de itens editáveis

```

01 void LoadItensBones ()
02     {
03         foreach (BonesVariations_Class boneVariation in bonesVariationsClass)
04         {
05             GameObject sliderObject = Instantiate(slider);
06             sliderObject.transform.SetParent(contentPanel, false);
07             text = sliderObject.GetComponentInChildren<Text>();
08             text.text = boneVariation.bonesVariationsName;
09
10             modelBonesClass =
11                 modelBonesDal.makeSelectBonesForVariation (
12                     boneVariation.bonesVariationsID);
13                 sliderObject.GetComponentInChildren<InsideSliderScript>().
14                     SetBones (modelBonesClass);
15         }
16     }
17 void LoadItensParts ()
18     {
19         modelParts = modelPartsDal.makeSelectModelPartsByModel (
20             modelDal.makeSelectByName (model.name) .modelID);
21         Transform[] allChildren = model.GetComponentInChildren<Transform> (true);
22         foreach (Transform child in allChildren)
23         {
24             for (int i = 0; i < modelParts.Count; i++)
25             {
26                 if (child.name == modelParts[i].modelPartsName)
27                 {
28                     itensList.Add(child);
29
30                     GameObject ToggleObject = Instantiate(toggle);
31                     ToggleObject.transform.SetParent(contentPanel, false);
32                     text = ToggleObject.GetComponentInChildren<Text>();
33                     text.text = modelParts[i].modelPartsNameToShow;
34
35                     ToggleObject.GetComponentInChildren<InsideToggleScript>().
36                         SetObject (child);
37
38                     break;
39                 }
40             }
41         }
42     }

```

O método `LoadItensBones` busca as possíveis modificações de malha padrão, em seguida para cada uma delas, instancia-se um *slider* e um texto descrevendo qual a parte do corpo o *slider* deve modificar. Os ossos alterados por aquela modificação são buscados e informados ao *script* `InsiderSliderScrip` de cada *slider* instanciado. Este *script* é detalhado no Quadro 4.

O método `LoadItensParts` busca as possíveis partes adicionais de cada personagem. Cada modelo de personagem possui dentro do objeto, as partes extras, sendo que nos modelos básicos elas são desativadas para que não tenham processamento. Depois de cada parte ser localizada na malha, é instanciado um *toggle* com o nome da parte adicionável e o objeto que ela é responsável. Se for marcado, o objeto é ativado, caso contrário, se mantém com o *status* desativado.

Quadro 4 – Cálculo de deformação do osso

```

01 public void SliderOnValueChanged(float value)
02     {
03         int i= 0;
04         foreach (Transform bone in bonesTransform)
05             {
06                 if (bones[i].maxPositionX != 0 || bones[i].maxPositionY != 0 ||
07                     bones[i].maxPositionZ != 0)
08                 {
09                     bone.localPosition = new Vector3(
10                         (value * (bones[i].maxPositionX - bones[i].minPositionX) +
11                         bones[i].minPositionX) , (value * (bones[i].maxPositionY -
12                         bones[i].minPositionY) + bones[i].minPositionY),
13                         (value * (bones[i].maxPositionZ - bones[i].minPositionZ)) +
14                         bones[i].minPositionZ);
15                 }
16                 if(bones[i].maxScaleX != 0 || bones[i].maxScaleY != 0 ||
17                     bones[i].maxScaleZ != 0)
18                 {
19                     bone.localScale = new Vector3(
20                         (value * (bones[i].maxScaleX - bones[i].minScaleX) +
21                         bones[i].minScaleX) , (value * (bones[i].maxScaleY -
22                         bones[i].minScaleY) + bones[i].minScaleY),
23                         (value * (bones[i].maxScaleZ - bones[i].minScaleZ)) +
24                         bones[i].minScaleZ);
25                 }
26                 i++;
27             }
28     }

```

O método `SliderOnValueChanged` é chamado quando o *slider* que representa a variação da malha é alterado pelo usuário. Ele recebe um valor relativo à posição do *slider* como parâmetro. Em seguida, cada osso pelo qual ele é responsável será alterado. Primeiro é verificado qual tipo de alterações que o osso possui, sendo a posição e a escala as possíveis mudanças. A modificação é feita por meio de um cálculo que usa a posição relativa do *slider* comparado ao valor máximo e mínimo da alteração de cada osso em questão.

Para criação de personagens do mestre os passos seguem os de Wyatt (2008). A criação de PdM é mais simples que a de PdJ, pois não há restrições ao que o mestre pode criar. Tudo se resume a uma única grande etapa na qual o mestre descreve os valores das defesas, vida, iniciativa, sentidos, etc. O mestre pode adicionar e criar os poderes que desejar para o personagem. Em seguida, decide quais os valores de perícias, se possuir, e dos atributos, descrevendo quantos e quais equipamentos existem. Então, o modelo é editado da mesma forma que na criação de PdJ, a Figura 23 mostra uma parte da criação de PdM.

Figura 23 – Edição de modelo de PdM

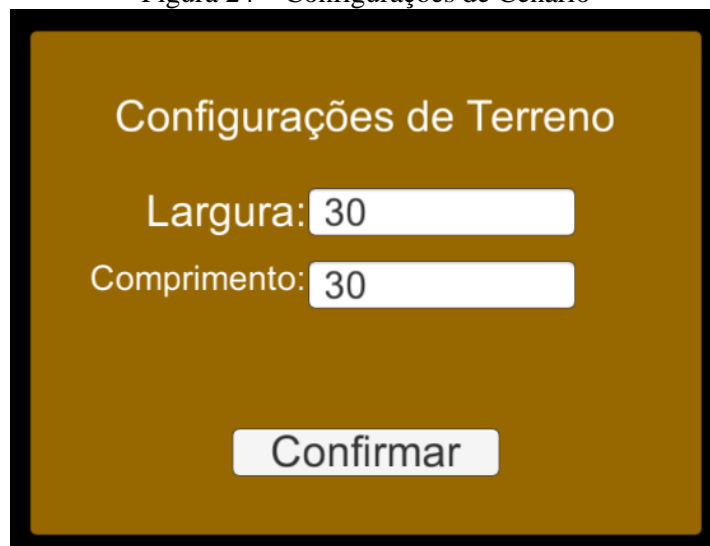


A Figura 23 mostra como é feita a edição de modelo de personagem do mestre. O modelo representa um Orc, onde é possível verificar que o mestre está adicionando partes, como dentes e garras. Na parte superior da imagem, também se pode ver a parte da criação de PdM, no que diz respeito a ficha de personagem com os valores de atributos e equipamentos.

### 3.3.6 Criação de mapas

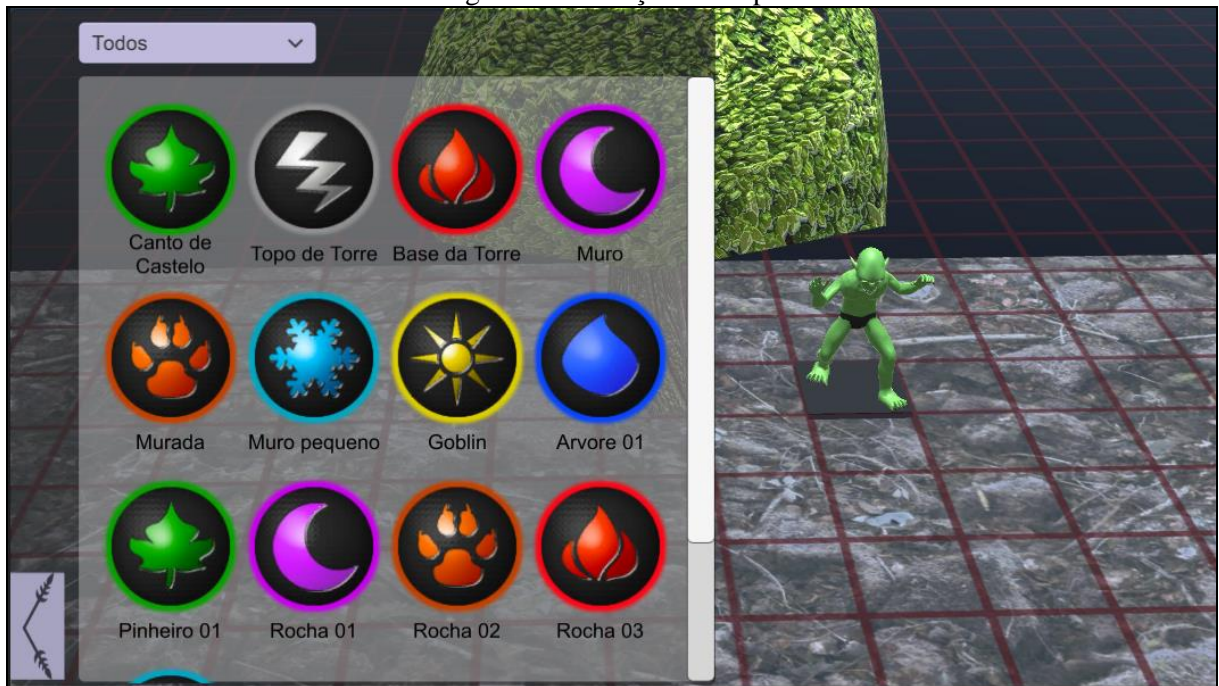
A criação de mapas é restrita ao usuário mestre, pois ele é o responsável por criar a narrativa e determinar as dimensões do mapa, conforme mostra a Figura 24. Lembrando que o terreno segue as especificações do jogo de D&D, sendo que cada quadrado do *grid* corresponde a 1,5 metros.

Figura 24 – Configurações de Cenário



O usuário dispõe de uma gama de objetos básicos, além dos que ele mesmo pode criar e adicionar ao cenário, controlando posição, rotação e escala. A tela de criação de mapas possui um *grid* quadriculado, simulando um mapa de jogo de D&D. O mestre pode acessar os objetos através do menu lateral. A Figura 25 mostra o menu de objetos aberto, com suas opções.

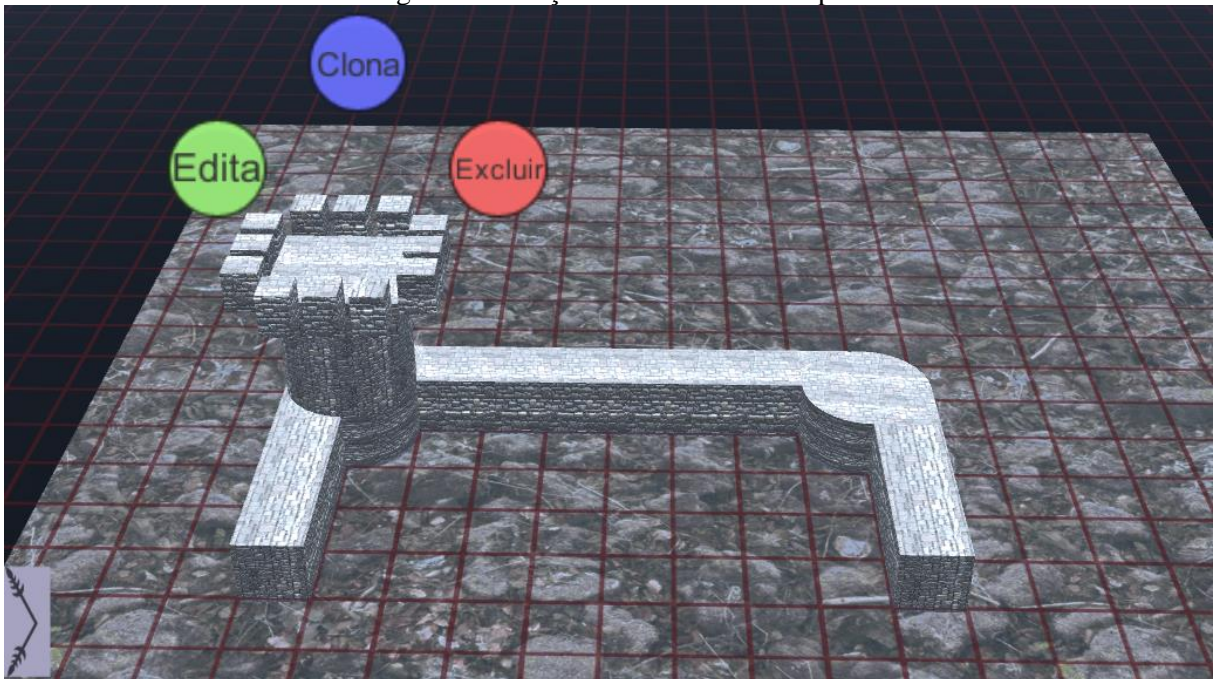
Figura 25 – Criação de mapa



A Figura 25 demonstra como é feito o posicionamento dos objetos em tela. Além disso, ela também mostra como um objeto é representado dentro do mapa. Quando um objeto é selecionado, uma instância do modelo que o representa é invocada à tela, seguindo a posição do *mouse*, permitindo que o mestre o coloque na posição desejada dentro do *grid* correspondente ao mapa. A Figura 26 mostra as opções de objetos disponíveis para o mestre (editar, clonar ou excluir).

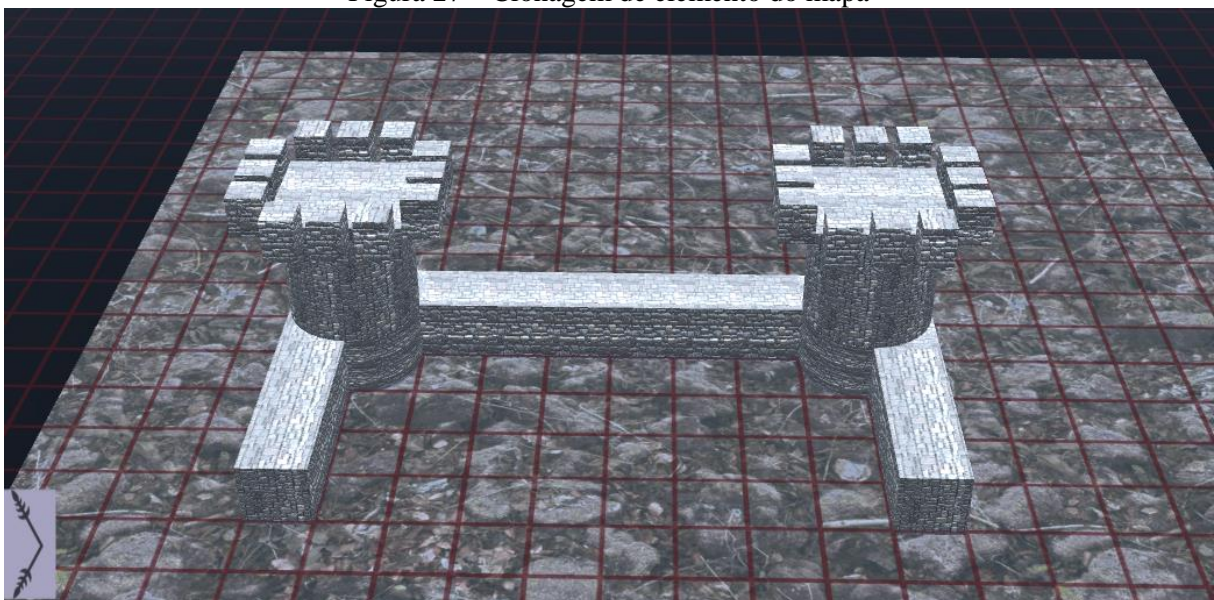


Figura 26– Edição de elemento do mapa



A Figura 26 mostra as opções de edição que são abertas quando o item é clicado. A opção “Edita” informa ao sistema que o item clicado é um objeto sobre controle do mestre, permitindo ao mestre reposicionar, girar ou alterar seu tamanho. A opção “Excluir”, elimina o objeto da cena. E, por fim, a opção “Clonar” permite que o objeto seja instanciado novamente. A Figura 27 mostra a mesma imagem após a clonagem do objeto.

Figura 27 – Clonagem de elemento do mapa



A Figura 27 demonstra a tela de criação de mapa após o usuário clonar o objeto de torre. Quando um objeto é clonado ele é instanciado a partir do objeto pai, herdando a mesma rotação e escala do mesmo, ficando selecionado para que o mestre o coloque na posição desejada.

A construção do cenário é controlada por três classes principais, `BuildingManager`, `BuildingPlacement` e `PlacebleBuilding`. Na classe `BuildingManager` são armazenados todos os modelos do cenário, personagens e objetos. O principal método, desta classe, `InstantiateObject` é mostrado no Quadro 5.

Quadro 5 – Controle de item selecionado

```

01 public void InstantiateObject(string objectName, bool isPlayer)
02 {
03     for (int i = 0; i < buildings.Length; i ++)
04     {
05         if(buildings[i].name == objectName)
06         {
07             buildingPlacement.SetItem(buildings[i], false, false, isPlayer);
08         }
09     }
10 }

```

O Quadro 5 descreve como é feito o controle do item selecionado pelo usuário. O método `InstantiateObject` recebe o nome do objeto e a *flag* avisando que se trate de um modelo de jogador. A partir disso, é feita a busca através dos objetos armazenados no *script* em forma de um componente `GameObject`, que informa à classe `BuildingPlacement` a qual item precisa ser instanciado.

O método recebe o modelo como `GameObject`, além das *flags* de edição, clonagem e se é um jogador. O método então marca o objeto como não posicionado e o instancia na tela. A partir deste momento, ele segue o movimento do mouse do usuário que está criando o cenário, conforme poder ser visto no Quadro 6.

Quadro 6 – Posicionamento do item

```

01 if (currentBuilding != null
02     && !hasPlaced
03     && (!PhotonNetwork.connected
04     || PhotonNetwork.isMasterClient))
05 {
06     DoCorrectCenter();
07     if (Physics.Raycast(defaultRay, out defaultHit, Mathf.Infinity))
08     {
09         if (defaultHit.collider.name.Equals("Terrain")
10             || (defaultHit.collider.tag.Equals("Building")
11                 && !defaultHit.collider.name.Equals(currentBuilding.name)))
12         {
13             defaultPosition = defaultHit.point;
14             currentBuilding.position = new Vector3(
15                 defaultPosition.x - (defaultPosition.x % (1.5f)) + correctCenter,
16                 currentBuilding.position.y, (defaultPosition.z -
17                 (defaultPosition.z % (1.5f)) + correctCenter));
18         }
19     }
20 }

```

No método `Update` da classe `BuildingPlacement`, são feitas as verificações se o item existe. Se ele já foi inserido no cenário ou se o usuário não é um jogador. O método `DoCorectCenter` muda o centro da malha de acordo com a escala, para que ocupe corretamente o espaço dentro do *grid*. Em seguida um `Raycast` é emitido pelo objeto da `Main Camera` onde o *script* se encontra. A posição do objeto segue a posição do mouse no eixo X e

no eixo Z, mas mantém o eixo Y, sendo 0 como padrão, mas pode ser modificado. Posteriormente, é feito um cálculo simples para manter a posição exata no *grid*, movendo o objeto apenas entre os quadrados. Para fixar a posição basta alterar a variável `hasPlaced` para *true*.

### 3.3.7 Durante o jogo

Ao clicar no botão Jogar na tela da Figura 21, o usuário é enviado primeiramente para a tela de conexão com o servidor, onde são exibidas as salas existentes. Caso o usuário seja mestre, ele deve criar a sala. A tela de servidores é exemplificada na Figura 28.

Figura 28 – Tela de Servidores



A tela de servidores exibe o *status* da conexão, as salas criadas pelos mestres e a opção de criar novas salas. Quando o usuário cria uma sala, ele se torna o mestre de jogo, os usuários que entrarem subsequentemente a sala, se tornam jogadores.

Ao entrar na sala de jogo, o jogador recebe todos os itens que o mestre distribuiu na tela com suas posições, rotações e escalas. Em seguida instancia o seu comunicador *voip* para todos os usuários conectados, ou que ainda serão conectados, para que possam receber o áudio deste. Todos os itens de interação entre os usuários, como modelos e o próprio comunicador precisam de uma instancia da classe `PhotonView`. O `Photon PUN` se utiliza da Classe `PhotonView` como componente para encontrar os itens nas aplicações de outros usuários. O Quadro 7, mostra como é feita a transferência de dados entre os usuários, exemplificando através da comunicação por voz.

Quadro 7 – Comunicação voip

```

01 void Start()
02 {
03     if (photonView.isMine)
04     {
05         GetComponent<UnityAudioPlayer>().enabled = false;
06     }
07     else
08     {
09         GetComponent<MicrophoneInputDevice>().enabled = false;
10     }
11 }

12 protected override void OnAudioDataEncoded(VoicePacketWrapper encodedFrame)
13 {
14     if (photonView.isMine)
15     {
16         byte[] headers = encodedFrame.ObtainHeaders();
17         photonView.RPC("OnAudioRecived", PhotonTargets.Others, headers,
18                                     encodedFrame.RawData);
19     }
20 }

21
22 [PunRPC]
23 void OnAudioRecived(byte[] headers, byte[] rawData)
24 {
25     if (!photonView.isMine)
26     {
27         VoicePacketWrapper packet = new VoicePacketWrapper(headers, rawData);
28         ReceiveAudioData(packet);
29     }
30 }

```

A classe `MyLocalVoiceController` é instanciada individualmente por cada usuário. Cada instancia verifica se o componente `PhotonView` pertence a elas, caso pertença, o áudio dele é desativado, caso contrário, o microfone é desativado. Desativando o áudio o usuário não ouve a si mesmo e, desativando o microfone, o usuário não interfere no áudio dos outros jogadores.

O método `OnAudioDataEncoded`, nativo do Unity, é sobrescrito para adaptar-se ao envio de voz, assim como mostra o Quadro 8. Todo áudio recebido é transformado em dois *arrays* de byte, um com as informações de quem envia e o outro com o áudio em si. O RPC `OnAudioRecived` então é chamado, onde os bytes são convertidos em áudio. O parâmetro `PhotonTargets.Others`, garante que todos os usuários conectados com a exceção de quem envia o RPC, receba seu conteúdo.

O mestre posiciona inicialmente a representação da miniatura do personagem no cenário, a partir disso, o jogador também pode movê-lo. A tela principal do ORC é apresentada na Figura 29.

Figura 29 – Tela principal do ORC



A Figura 29 mostra que a tela de jogo se assemelha à de criação de mapas. Porém, a tela de jogo possui comunicação entre os jogadores, por meio do *chat* escrito, conforme pode ser visto no canto inferior direito da imagem. A comunicação também é feita por meio de *chat* voip. Sendo que para não sobrecarregar o envio de dados, o usuário deve pressionar a tecla `Alt` para falar. A Figura 29 também mostra um exemplo de jogada de dados, que é enviada por meio do *chat*.

### 3.3.8 Conexão com o Photon PUN

As principais funções para a conexão com o Photon, se encontram abstraídas na classe `PhotonNetwork`. Essa classe pertence ao pacote `Photon PUN +`, disponível na *Asset Store* do Unity. O Quadro 8 mostra um trecho do *script* `NetworkManager`, onde é feita a conexão ao *lobby* do Photon.

Quadro 8 – Conexão com o Photon

```

01
02 void Start()
03 {
04     PhotonNetwork.automaticallySyncScene = true;
05     PhotonNetwork.ConnectUsingSettings ("0.1");
06 }
07
08 void GetRooms()
09 {
10     if (PhotonNetwork.insideLobby)
11     {
12         foreach ( RoomInfo game in PhotonNetwork.GetRoomList())
13         {
14             AddRoom(game.name);
15         }
16     }
17 }

```

O método `Start` é chamado assim que o objeto que possui o *script* anexado aparece na tela, no caso, o `_NetworkManager`. A primeira coisa que o método faz é sincronizar a cena automaticamente. A seguir, chamando a função `ConnectUsingSettings`, o usuário inicia a conexão, passando uma *string* com o número que representa a versão do aplicativo. Efetivando a conexão, o método `GetRooms` verifica se a conexão obteve sucesso. Se estiver conectado ao *lobby* Photon, realiza a busca e a exibição gráfica de todas as salas já criadas por outros usuários. A criação das salas é de responsabilidade do *script* `CreatRooms`, exibido no Quadro 9.

Quadro 9 – Criação de salas

```

01 public void CreateRoom ()
02 {
03     roomName = roomNameText.text;
04     if (isValidRoomName && PhotonNetwork.insideLobby)
05     {
06         RoomOptions ro = new RoomOptions () {isVisible = true, maxPlayers = 20};
07         PhotonNetwork.CreateRoom (roomName, ro, TypedLobby.Default);
08     }
09 }

```

O método `CreateRoom`, verifica se o nome da sala a ser criada é válido e, se a conexão foi estabelecida. Após estas verificações, são determinadas as opções de sala. Neste caso, somente a visibilidade e o número máximo de jogadores foi estipulado. A sala então é criada e posicionada no *lobby*, para que os outros usuários possam ver e interagir com ela.

### 3.4 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados os experimentos realizados com o protótipo. O experimento foi realizado com usuários de diferentes níveis de familiaridade com RPG de mesa e digital.

### 3.4.1 Metodologia

O experimento aconteceu durante o mês de junho por meio de testes realizados com 6 usuários, tanto individualmente, quanto em grupo. Durante o experimento foi fornecido a cada usuário um livro do jogador, um livro do monstro, um notebook e um questionário para avaliar o protótipo.

### 3.4.2 Cenário de teste

O teste consistia em seguir as etapas de cada tipo de usuário, mestre e jogador, para um jogo de RPG. Antes de começar o teste, o jogo foi apresentado de maneira geral. No Quadro 10 são exibidos os perfis dos usuários envolvidos no experimento.

Quadro 10 – Perfis dos usuários

Idade	66,66% de 16 e 20 anos 33,33% de 21 e 25 anos
Sexo	66,66% masculino 33,33% feminino
Grau de formação	16,66% ensino médio completo 83,33% ensino superior incompleto
Qual a experiência com RPG	33,33% jogou pouco 16,66% joga às vezes 50% joga sempre que pode
Usa ferramentas de RPG de mesa online	50% nunca 50% às vezes
Se você já jogou RPG, que tipo de jogador é	66,66% jogador 33,33% às vezes mestre, às vezes jogador

O Quadro 10 mostra que os usuários que participaram do teste são em sua maioria homens de 16 a 20 anos com ensino superior incompleto. Metade deles joga RPG com certa frequência. Nota-se também que a maioria dos usuários é apenas jogador, sendo que uma pequena parcela deles joga também como mestre de RPG. Vale destacar que metade deles já usaram ferramentas de RPG de mesa *online*. A seguir são analisados os resultados dos testes realizados em relação à usabilidade do ORC.

A primeira etapa foi a criação de usuário dentro do jogo. Em seguida todos os jogadores começaram os testes na perspectiva de jogador, criando e personalizando um PdJ. Depois disso, foram feitos testes como usuário mestre, criando um personagem PdM, mapa e por fim, jogando como mestre.

Após a realização dos testes, os usuários informaram se conseguiram ou não cumprir os objetivos do ORC. Os resultados obtidos estão detalhados no Quadro 11.

Quadro 11 – Respostas do questionário sobre a avaliação dos instrumentos do ORC

Conseguiu criar um usuário	100% sim
Conseguiu criar um PdJ	100% sim
Conseguiu criar um PdM	100% sim
Conseguiu criar um mapa	100% sim
Conseguiu criar uma sala de jogo	100% sim
Conseguiu conectar-se a uma sala de jogo	100% sim
Conseguiu comunicar-se por chat de voz	100% sim
Conseguiu comunicar-se por chat escrito	100% sim
Conseguiu fazer jogada de dados	100% sim

Conforme mostra o Quadro 11, todos os usuários conseguiram realizar as tarefas propostas durante os testes do ORC. O que foi um resultado positivo, demonstrando que os módulos são relativamente simples de se operar. Ao final, cada usuário respondeu mais especificamente sobre cada módulo do ORC. O primeiro foi a criação de personagens PdJ. Os resultados estão detalhados no Quadro 12.

Quadro 12 – Respostas do questionário de usabilidade da criação de PdJ

Como foi criar um PdJ	100% fácil
De modo geral foi intuitivo criar o personagem	100% sim
Avaliação da criação de PdJ	16,66% muito bom 83,33% bom
A criação de PdJ facilita a criação de um personagem de RPG	100% sim
O que achou da edição de modelo de personagem	66,66% muito bom 33,33% bom

Durante a criação de personagens do jogador, pode-se averiguar que todos os usuários participantes do teste acharam fácil criar o seu personagem. Assim como, todos acharam que o módulo é intuitivo. A avaliação da criação de PdJ foi positiva. Entretanto, avaliando as opções possíveis de resposta, o ORC ainda precisa melhorar. Todavia, todos também acharam que a criação de PdJ do ORC facilita a criação de personagem de RPG, sendo que esta etapa costuma levar algum tempo do jeito tradicional. A customização do modelo foi considerada, pela maioria, como sendo muito boa. Isso indica que o ORC facilita a personalização dos objetos gráficos existentes no jogo.

Na sequência, os usuários responderam sobre a tela de jogo, primeiro sobre o ponto de vista do jogador. Os resultados estão detalhados no Quadro 13.



Quadro 13 – Respostas do questionário de usabilidade da tela de jogo (Jogador)

Como foi jogar	50% muito fácil 50% fácil
De modo geral foi intuitivo interagir com o jogo	100% sim
Avaliação da tela de jogo	100% muito bom
Como foi a comunicação	83,33% muito bom 16,66% bom
A tela de jogo simula bem um jogo de RPG para um jogador	100% sim

Em relação a tela de jogo, como jogador, a avaliação foi muito positiva. Todos os usuários consideraram o módulo intuitivo, e conseqüentemente todos acharam que foi fácil de usá-lo. A tela de jogo foi avaliada por todos como muito boa, assim como, a comunicação. Em seguida, foi respondido sobre a criação de personagens PdM. Os resultados obtidos estão detalhados no Quadro 14.

Quadro 14 – Respostas do questionário de usabilidade da criação de PdM

Como foi criar um PdM	100% muito fácil
De modo geral foi intuitivo criar o personagem	100% sim
Avaliação da criação de PdM	16,66% ótimo 83,33% muito bom
A criação de PdM facilita a criação de um personagem de RPG	100% sim
O que achou da edição de modelo de personagem	83,33% muito bom 16,66% bom

A criação de personagens do mestre obteve bons resultados. Ele foi considerado por todos os usuários, como sendo intuitivo e de fácil manipulação. O módulo teve a melhor avaliação do ORC, sendo que todos os usuários avaliaram este item como sendo muito bom ou ótimo. Também foi constatado que o módulo facilita a criação e edição de personagens de RPG. O que demonstra que a permissão de escolha do modelo básico agradou os usuários. Depois disso, cada usuário respondeu sobre o módulo de criação de mapas. Os resultados estão detalhados do Quadro 15.

Quadro 15 – Respostas do questionário de usabilidade da criação de mapas

Como foi criar um mapa	50% muito fácil 50% fácil
De modo geral foi intuitivo criar o mapa	100% sim
Avaliação da criação de mapas	100% muito bom
Qual foi a maior dificuldade	100% não houve dificuldades

A criação de mapas foi considerada intuitiva e fácil uso. Além disso, o módulo foi avaliado como sendo muito bom por todos, conforme os resultados do Quadro 15.

O último módulo analisado foi o jogo do ponto de vista do mestre. Para esse módulo, os resultados encontram-se detalhados no Quadro 16.

Quadro 16 – Respostas do questionário de usabilidade da tela de jogo (Mestre)

Como foi jogar	33,33% muito fácil 66,66% fácil
De modo geral foi intuitivo interagir com o jogo	100% sim
Avaliação da tela de jogo	50% muito bom 50% bom
Como foi a comunicação	83,33% muito bom 16,66% bom
A tela de jogo simula bem um jogo de RPG para um mestre	100% sim
A tela facilita o trabalho do mestre	100% sim

A partir do Quadro 16, percebe-se que os usuários ficaram contentes com o jogo, sendo que todos indicaram que ele é fácil, intuitivo e simula bem um jogo de RPG.

Por fim, os usuários responderam sobre a opinião geral sobre o ORC. Os resultados estão detalhados no Quadro 17.

Quadro 17 – Respostas do questionário de usabilidade geral do ORC

Como foi usar o ORC	50% muito fácil 50% fácil
Avaliação geral do ORC	100% muito bom
O ORC é útil como ferramenta de RPG	100% sim
Opinião sobre os modelos 3D	66,66% muito bons 33,3% bons
Opinião sobre o cenário	50% muito bom 50% bom
Gostou do ORC	100% sim

O Quadro 17 mostra que o protótipo obteve resultados positivos em todos os módulos. Porém, um ponto importante a ser observado é que somente metade dos usuários já tinham usado ferramentas de RPG *online*. Além disso, alguns usuários jogaram pouco, demonstrando que o ORC é uma ferramenta que apresenta pouca dificuldade para quem nunca usou ferramentas do tipo. Todos os usuários também acreditam que o ORC é útil para jogar RPG, e que pode crescer, principalmente por ser simples de usar, assemelhando-se aos jogos digitais, trazendo familiaridade ao usuário.

### 3.4.3 Comparação entre os trabalhos correlatos e o trabalho proposto

No Quadro 17 é realizada a comparação entre as principais características dos trabalhos correlatos e o protótipo desenvolvido neste trabalho.

Quadro 18 – Comparação entre os trabalhos correlatos e o ORC

Características/ Trabalhos	Trabalho proposto	<i>Tabletop Forge</i> (2012)	Taulukko (2008)	RRPG <i>Firecast</i> (2009)
Gráficos 3D	Sim	Não	Não	Não
Modificação gráfica de personagens	Sim	Parcial	Não	Parcial
Dinâmica de luz	Sim	Sim	Não	Parcial
Facilitação de encontro de mesa	Não	Sim	Sim	Não
Rolagem de dados	Sim	Sim	Sim	Sim
Chat <i>voip</i>	Sim	Sim	Sim	Não
Chat escrito	Sim	Sim	Sim	Sim

A partir do Quadro 17 pode-se perceber que existem muitas características em comum entre as ferramentas de RPG. Observa-se também que apenas o trabalho proposto (ORC), possui recursos 3D e opções de personalização gráfica. Devido a esta característica, os modelos podem interagir com a luz do cenário, mudando de acordo com sua rotação. Entretanto devido à complexidade de cada modelo, torna-se limitado o número de modelos disponíveis.

O ORC, assim como o RRPg *Firecast* (2009), não possui um sistema para facilitar um encontro de mesa, sendo que o ORC suporta no máximo 100 jogadores simultâneos. Além disso, todos apresentam algum tipo de rolagem de dados ou de comunicação.

## 4 CONCLUSÕES

Este trabalho apresentou o desenvolvimento do ORC, cujo principal objetivo é simular um ambiente *online* de RPG de forma satisfatória, permitindo aos usuários uma interação mais permissiva no que diz respeito à representação visual, tanto de seus personagens quanto de suas representações de tabuleiro.

Durante os testes, pode-se perceber que os objetivos foram alcançados, de forma que todos os usuários ficaram satisfeitos com o ORC. Também foi possível averiguar algumas melhorias possíveis ao sistema, principalmente em relação à criação de personagens, onde as diversas etapas poderiam ter sido simplificadas ao que o usuário deseja descrever, pois as regras podem variar de acordo com o que os jogadores desejam ou acordam.

Ao desenvolver o protótipo encontrou-se dificuldade em disponibilizar modelos que podem ser personalizados. Devido ao tempo de que cada modelo necessita para que o *rigging* seja feito de forma adequada, o ORC se encontra limitado quanto as opções de modelos.

Conforme as pesquisas realizadas, notou-se que o ORC é possivelmente o único protótipo em desenvolvimento que simula uma mesa de jogo de RPG e permite a customização de um personagem com modelos 3D. Sendo assim, o ORC se mostrou uma boa opção para simular um ambiente *online* onde é possível jogar RPG e ter representação visual personalizada.

### 4.1 EXTENSÕES

Algumas possíveis extensões para o ORC são:

- a) estender o ORC para que o mesmo possa simular outros sistemas de RPG, além do *Dungeons & Dragons*;
- b) permitir que os usuários alterem as cores dos modelos 3D, permitindo assim um leque muito maior de possibilidades de customização;
- c) criar mais modelos passíveis de alteração da malha para serem usados como personagens e objetos;
- d) criar uma funcionalidade que melhore a edição de modelos, permitindo que outros modelos possam ser adicionados;
- e) possibilitar a criação de personagens mais adaptados. Ao quais os jogadores poderão criar fichas personalizadas, variações de regras ou outros jogos de RPG.

## REFERÊNCIAS

- ANDRADE, Flávio. **Caminhos para o uso do RPG na Educação**. [S.l.], [2000?]. Disponível em: <[http://www.uff.br/aleph/textos\\_em\\_pdf/caminhos\\_para\\_o\\_uso\\_do\\_rpg\\_na\\_educacao.pdf](http://www.uff.br/aleph/textos_em_pdf/caminhos_para_o_uso_do_rpg_na_educacao.pdf)>. Acesso em: 20 jun. 2016.
- JONES, Clave. **It's a Tyranny of Dragons! Reviewing the New D&D Miniatures**. [S.l.], 2014. Disponível em: <<http://www.nerdsonearth.com/2014/11/dnd-miniatures-review-tyranny-dragons-pathfinder-battles-minis-dungeons-and-dragons/>>. Acesso em: 28 jun. 2016.
- JOSÉ FILHO, Ricardo A. **O Livro de Role-playing game: uma análise das ferramentas da obra de RPG, o hipertexto e a interpretação**. 2014. 125 f. Trabalho de Conclusão de Curso (Bacharelado em Biblioteconomia) – Centro de Ciências Humanas e Sociais, Universidade Federal do Rio de Janeiro, Rio de Janeiro.
- JUUL, Jesper. **The Game, the Player, the World: Looking for a Heart of Gameness**. [S.l.], 2003. Disponível em: <[http://ocw.metu.edu.tr/pluginfile.php/4471/mod\\_resource/content/0/ceit706/week3\\_new/JesperJuul\\_GamePlayerWorld.pdf](http://ocw.metu.edu.tr/pluginfile.php/4471/mod_resource/content/0/ceit706/week3_new/JesperJuul_GamePlayerWorld.pdf)>. Acesso em: 26 jun. 2016.
- KEVIN OWENS ENTERPRISES. **RPG Dice**. [S.l.], 2012. Disponível em: <<http://www.rpg-dice.com/>>. Acesso em: 28 jun. 2016.
- LEON, Leandro R. O. T. C. **MMORPG: Projeto e Implementação**. 2004. 66 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Universidade Federal de Santa Catarina, Florianópolis.
- MACHADO, Marlos C. et al. **Player Modeling: Towards a Common Taxonomy**. Belo Horizonte, 2011. Disponível em: <<http://homepages.dcc.ufmg.br/~chaimo/public/CGames11.pdf>>. Acesso em: 26 jun. 2016.
- MALLMANN, Eduardo R. **Estudo e Desenvolvimento de um Jogo Utilizando Unreal Development Kit**. 2012. 67 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Universidade Regional do Noroeste do Estado do Rio Grande do Sul, Santa Rosa.
- PRIETTO, T. G. **Literatura e os jogos de RPG: Trajetória de apropriações e intertextos**. Revista Translatio [S.l.], n. 6, p.77-85, 2013.
- RIYIS, Marcos T. **SIMPLES: Sistema Inicial para Mestres-Professores Lecionarem através de uma Estratégia Motivadora**. São Paulo: Ed. do Autor, 2004.
- RODRIGUES, Lia C.; LOPES, Rodrigo A. S. P.; MUSTARO, Pollyana N. **Impactos sócio-culturais da evolução dos jogos eletrônicos e ferramentas comunicacionais: um estudo sobre o desenvolvimento de comunidades virtuais de jogadores**. [2008?]. 9 f. Dissertação (Bacharelado em Ciências da Computação) - Dept. Ciência da Computação, Universidade Presbiteriana Mackenzie, São Paulo.
- RRPG FIRECAST. **Jogue pela internet**. [S.l.]. 2009. Disponível em: <<http://www.rrpg.com.br/>>. Acesso em: 21 jun. 2016.
- SALES, Mateus. **RPG (Role-Playing Game)**. [S.l.], 2014. Disponível em: <<http://www.brasilecola.com/curiosidades/rpg.htm>>. Acesso em: 22 jun. 2016.
- TABLET FORGE. **Roll20 Virtual Tabletop**. [S.l.]. 2012. Disponível em: <<http://roll20.net/>>. Acesso em: 20 jun. 2016.

TAULUKKO. **Sua mesa de RPG na web**. [S.l.], 2008. Disponível em: <<http://taulukko.com.br/>>. Acesso em 21 jun. 2016.

WYATT, James. **Dungeons & Dragons: Guia do Mestre**. 4. ed. São Paulo: Devir Livraria, 2008.

VERAS, A.; SANTOS, C. B. **RPG como ferramenta de ensino de literatura**. [S.l.], 2004. Disponível em: <<http://www.rpgeduc.com/old/artigo02.pdf>>. Acesso em: 22 jun. 2016.

ZAPT MÍDIA. **PESQUISA RPG BRASIL Hábitos e consumo junho-2009**. [S.l.], 2009. Disponível em: < <http://www.metonimia.com.br/pesquisa/rpg.pdf> > Acesso em: 21 jun. 2016.

## APÊNDICE A – Descrição dos Casos de Uso

A seguir é descrito o detalhamento dos casos de uso, com a descrição dos cenários. O caso de uso UC01 - Cadastro de usuários descreve a interação entre o mestre, jogador e a funcionalidade de cadastrar novos usuários. Os detalhes deste caso de uso são apresentados no Quadro 19.

Quadro 19 – Caso de uso UC01 - Cadastro de usuários

Número	01
Caso de uso	Cadastro de usuários
Ator	Mestre/Jogador
Cenário principal	<ol style="list-style-type: none"> <li>1. O usuário escreve o nome do usuário desejado.</li> <li>2. O usuário clica no botão criar usuário.</li> </ol>
Cenário de exceção	No passo 1, caso já exista usuário com o mesmo nome, ou caso o nome esteja em branco, será exibida a mensagem “Usuário inválido”
Pós-condições	O usuário foi criado.

O caso de uso UC02 - Implementar ficha de PdJs descreve a relação entre o jogador e a funcionalidade que permite a implementação da ficha do personagem do jogador. Detalhes deste caso de uso estão descritos no Quadro 20.

Quadro 20 – Caso de uso UC02 - Implementar ficha de PdJs

Número	02
Caso de uso	Implementar ficha de PdJs
Ator	Jogador
Pré-condições	UC01 - Cadastro de usuários
Cenário principal	<ol style="list-style-type: none"> <li>1. O usuário informa todos os dados da ficha do personagem do jogador.</li> <li>2. O usuário clica em criar personagem.</li> </ol>
Cenário de exceção	No passo 1, caso exista algum erro durante o cadastro, o usuário é direcionado à página onde o erro ocorreu, para que possa corrigi-lo.
Pós-condições	O personagem do jogador foi criado.

O caso de uso UC03 - Customizar PdJs descreve a relação entre o jogador e a funcionalidade que permite a personalização do modelo 3D que representa o personagem do jogador. Detalhes deste caso de uso estão descritos no Quadro 21.

Quadro 21 – Caso de uso UC03 - Customizar PdJs

Número	03
Caso de uso	Customizar PdJs
Ator	Jogador
Pré-condições	UC01 - Cadastro de usuários
Cenário principal	<ol style="list-style-type: none"> <li>1. Implementação de ficha de personagem não finalizada.</li> <li>2. O usuário modifica as partes do corpo que deseja no personagem.</li> <li>3. O usuário adiciona as partes do corpo que deseja.</li> </ol>
Cenário de exceção	Caso o usuário não realize o passo 2 e 3, o personagem será criado com um modelo padrão.
Pós-condições	O modelo representativo do personagem foi criado.

O caso de uso UC04 - Cadastrar PdMs descreve a relação entre o mestre a funcionalidade de cadastro de um personagem do mestre. Detalhes deste caso de uso estão descritos no Quadro 22.

Quadro 22 – Caso de uso UC04 - Cadastrar PdMs

Número	04
Caso de uso	Cadastrar de PdMs
Ator	Mestre
Pré-condições	UC01 - Cadastro de usuários
Cenário principal	<ol style="list-style-type: none"> <li>1. O usuário descreve as estatísticas exigidas pelo ORC sobre o PdM que deseja cadastrar.</li> <li>2. O usuário clica em criar personagem.</li> </ol>
Cenário de exceção	No passo 1, caso exista algum campo em branco, será exibida a mensagem “Por favor, preencha todos os itens para que possa ser criado o PdM.”.
Pós-condições	O personagem do mestre foi criado.

O caso de uso UC05 - Customizar PdMs descreve a relação entre o mestre e a funcionalidade que permite a personalização dos modelos 3D que representam os personagens do mestre. Detalhes deste caso de uso estão no Quadro 23.



Quadro 23 – Caso de uso UC05 - Customizar PdMs

Número	05
Caso de uso	Customizar PdMs
Ator	Mestre
Pré-condições	UC01 - Cadastro de usuários
Cenário principal	<ol style="list-style-type: none"> <li>1. Cadastro de PdM não finalizado.</li> <li>2. O usuário escolhe o modelo que deseja que represente seu PdM.</li> <li>3. O personagem modifica as partes da malha que deseja.</li> <li>4. O personagem adiciona as partes que deseja à malha.</li> </ol>
Cenário de exceção	No passo 2 se o usuário não escolher um modelo, será usado um modelo aleatório. Se o usuário não realizar os passos 3 e 4, será usado um modelo padrão.
Pós-condições	O personagem do mestre foi criado.

O caso de uso UC06 - Criar mapas descreve a relação entre o mestre a funcionalidade que permite criar um mapa de jogo de RPG. Detalhes deste caso de uso estão no Quadro 24.

Quadro 24 – Caso de uso UC06 - Criar mapas

Número	06
Caso de uso	Criar mapas
Ator	Mestre
Pré-condições	UC01 - Cadastro de usuários
Cenário principal	<ol style="list-style-type: none"> <li>1. O usuário informa as dimensões desejadas para o mapa a ser criado.</li> <li>2. O usuário pode adicionar os objetos que deseja ao mapa.</li> <li>3. O usuário redimensiona os objetos que deseja, mudando a escala, rotação e altura.</li> <li>4. O usuário clona os objetos que deseja.</li> </ol>

O caso de uso UC07 - Criar narrativa descreve a relação entre o mestre a funcionalidade de poder criar narrativas de RPG para simular uma mesa de jogo. Detalhes deste caso de uso estão no Quadro 25.

Quadro 25 – Caso de uso UC07 – Criar narrativa

Número	07
Caso de uso	Criar narrativa
Ator	Mestre
Pré-condições	UC01 - Cadastro de usuários UC04 - Cadastrar PdMs UC05 - Customizar PdMs UC06 - Criar mapas
Cenário principal	<ol style="list-style-type: none"> <li>1. O usuário controla o cenário.</li> <li>2. O usuário conduz o jogo.</li> <li>3. O usuário aplica as rolagens de dados.</li> </ol>

O caso de uso UC08 – Gerar rolagem de dados descreve a relação entre o sistema a funcionalidade de gerar rolagem de dados. Os detalhes deste caso de uso estão descritos no Quadro 26.

Quadro 26 – Caso de uso UC08 – Gerar rolagem de dados

Número	08
Caso de uso	Gerar rolagem de dados
Ator	Sistema
Pré-condições	UC07 - Criar narrativa
Cenário principal	<ol style="list-style-type: none"> <li>1. O sistema gera as rolagens de dados exigidas pela narrativa.</li> </ol>

O caso de uso UC09 – Restringir jogadas descreve a relação entre o mestre e a funcionalidade de restringir jogadas. Os detalhes deste caso de uso estão descritos no Quadro 27.

Quadro 27 – Caso de uso UC09 – Restringir jogadas

Número	09
Caso de uso	Restringir jogadas
Ator	Mestre
Pré-Condições	UC01 - Cadastro de usuários UC04 - Cadastrar PdMs UC05 - Customizar PdMs UC06 - Criar mapas UC07 - Criar narrativa
Cenário principal	<ol style="list-style-type: none"> <li>1. O usuário restringe as jogadas que achar necessárias.</li> <li>2. O usuário refaz as jogadas, se achar necessário.</li> </ol>

O caso de uso UC10 – Controlar pontos de experiência descreve a relação entre o mestre e a funcionalidade de controlar os pontos de experiência dos jogadores. Os detalhes deste caso de uso estão descritos no Quadro 28.

Quadro 28 – Caso de uso UC10 - Controlar pontos de experiência

Número	10
Caso de uso	Controlar pontos de experiência
Ator	Mestre
Pré-Condições	UC01 - Cadastro de usuários UC04 - Cadastrar PdMs UC05 - Customizar PdMs UC06 - Criar mapas UC07 - Criar narrativa
Cenário principal	<ol style="list-style-type: none"> <li>1. O usuário adiciona os pontos de experiência caso julgue necessário.</li> <li>2. O usuário retira os pontos de experiência caso julgue necessário.</li> </ol>

O caso de uso UC11 - Controle de vida e fichas descreve a relação entre o mestre e a funcionalidade de controlar os pontos de vida e as fichas de todos os personagens. Os detalhes deste caso de uso estão descritos no Quadro 29.

Quadro 29 – Caso de uso UC11 - Controle de vida e fichas

Número	11
Caso de uso	Controle de vida e fichas
Ator	Mestre
Pré-Condições	UC01 - Cadastro de usuários UC04 - Cadastrar PdMs UC05 - Customizar PdMs UC06 - Criar mapas UC07 - Criar narrativa
Cenário principal	<ol style="list-style-type: none"> <li>1. O usuário controla as estatísticas de todos os personagens.</li> <li>2. O usuário faz o controle de pontos de vida dos personagens.</li> </ol>

O caso de uso UC12 - Comunicação por chat descreve a relação entre o mestre, jogador e a funcionalidade de comunicação através de um dos chats disponibilizados pelo ORC. Os detalhes deste caso de uso estão descritos no Quadro 30.

Quadro 30 – Caso de uso UC12 - Comunicação por chat

Número	12
Caso de uso	Comunicação por chat
Ator	Mestre/Jogador
Pré-Condições	UC01 - Cadastro de usuários UC02 - Implementar ficha de PdJs UC03 - Customizar PdJs UC04 - Cadastrar PdMs UC05 - Customizar PdMs UC06 - Criar mapas UC07 - Criar narrativa
Cenário principal	<ol style="list-style-type: none"> <li>1. O usuário pressiona a tecla Alt para poder falar através do chat voip.</li> <li>2. O usuário fala o que deseja para dar continuidade ao jogo.</li> </ol>
Cenário alternativo	<ol style="list-style-type: none"> <li>1. O usuário escreve o que deseja.</li> <li>2. O usuário escolhe se deseja falar com todos, ou somente com o usuário mestre.</li> <li>3. O usuário clica no botão enviar.</li> </ol>

O caso de uso UC13 - Dispor prefabs descreve a relação entre o sistema e a funcionalidade de dispor modelos 3D como base para montar os personagens, objetos e mapas. Os detalhes deste caso de uso estão descritos no Quadro 31.

Quadro 31 – caso de uso UC13 - Dispor prefabs

Número	13
Caso de uso	Dispor <i>prefabs</i>
Ator	Sistema
Cenário principal	<ol style="list-style-type: none"> <li>1. O sistema disponibiliza modelos base para os usuários customizarem os personagens.</li> <li>2. O sistema disponibiliza modelos base para criação de mapas.</li> </ol>

## APÊNDICE B – Questionário de avaliação do ORC

Este apêndice apresenta os questionários de teste que os usuários responderam. Começando pelo questionário de perfil que se encontra no Quadro 32. Em seguida os testes como usuário jogador. No

Quadro 33 está a lista de atividades do usuário jogador. Em seguida os usuários passam pela lista de tarefas realizadas pelo o usuário mestre, apresentado no Quadro 34. E por fim, foi feita a avaliação do ORC de forma geral. O questionário de avaliação geral encontra-se no Quadro 35.

Quadro 32 – Questionario de perfil de usuário

<p><b>PERFIL DE USUARIO</b></p> <p>Nome: _____</p>
<p><b>Faixa etária:</b></p> <p>( ) Tenho de 5 e 10 anos.  ( ) Tenho de 11 e 15 anos.  ( ) Tenho de 16 e 20 anos.  ( ) Tenho de 21 e 25 anos.  ( ) Tenho de 26 e 30 anos.  ( ) Tenho mais de 30 anos.</p>
<p><b>Grau de formação:</b></p> <p>( ) Ensino fundamental incompleto.  ( ) Ensino fundamental completo.  ( ) Ensino médio incompleto.  ( ) Ensino médio completo.  ( ) Ensino superior incompleto.  ( ) Ensino superior completo.  ( ) Pós-graduação.</p>
<p><b>Qual a sua experiência com RPG?</b></p> <p>( ) Nunca joguei.  ( ) Joguei pouco.  ( ) Jogo às vezes.  ( ) Jogo sempre que posso.</p>
<p><b>Já usou ferramentas de RPG de mesa online?</b></p> <p>( ) Nunca.  ( ) Pouco.  ( ) Às vezes.  ( ) Sempre que jogo.</p>
<p><b>Que tipo de jogador você é?</b></p> <p>( ) Não jogo.  ( ) Mestre.  ( ) Jogador.  ( ) Às vezes mestre, às vezes jogador.</p>

Quadro 33 – Lista de tarefas a serem realizadas pelo jogador

<p><b>INSTRUÇÕES</b></p> <p>Este questionário busca avaliar a usabilidade do ORC do ponto de vista do jogador de RPG. O ORC permite que você crie um personagem de RPG e use-o em uma aventura de D&amp;D.</p> <p>Você pode utilizar o ORC a sua vontade por até 5 minutos, para que possa se familiarizar. Quando se sentir apto, por favor prossiga com os testes solicitados a seguir. Para esses testes é preciso que você use o Livro do Jogador.</p>
<p><b>Tarefas a serem realizadas pelo Jogador:</b></p> <p><b>Criando um personagem</b></p> <p>Comece a criação de um personagem PdJ, informando a raça, classe, valores de atributos, perícias treinadas, talentos, poderes, itens e nome.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p>
<p><b>Customizando o Personagem</b></p> <p>Modifique as partes do corpo de seu personagem ao seu gosto.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>Adicione partes ao corpo de seu personagem como desejar?</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p>
<p><b>Jogando</b></p> <p>Entre em uma sala de jogo previamente criada.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>Comunique-se com os outros usuários conectados.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>Realize uma jogada de dados</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>O ORC conseguiu simular uma mesa de jogo? ( ) Sim ( ) Não</p>

Quadro 34 – Lista de tarefas a serem realizadas pelo mestre

<p><b>INSTRUÇÕES</b></p> <p>Este questionário busca avaliar a usabilidade do ORC do ponto de vista do mestre de RPG. O ORC permite que você crie personagens e mapas de RPG e use-os em uma aventura de D&amp;D.</p> <p>Você pode utilizar o ORC a sua vontade por até 5 minutos, para que possa se familiarizar. Quando se sentir apto, por favor prossiga com os testes solicitados a seguir. Para esses testes é preciso que você use o Manual do Monstro.</p>
<p><b>Tarefas a serem realizadas pelo Mestre:</b></p> <p><b>Criando um personagem</b></p> <p>Comece a criação de um personagem PdM, informando nome, função nível, poderes, equipamentos, pericias, atributos etc.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p>
<p><b>Customizando o Personagem</b></p> <p>Escolha o modelo que parecer mais adequado</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>Modifique as partes do corpo de seu personagem ao seu gosto.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>Adicione partes ao corpo de seu personagem como desejar?</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p>
<p><b>Criando um Mapa</b></p> <p>Comece a criação do mapa, informando as dimensões desejadas para o mesmo</p>

<p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>Adicione os modelos que desejar.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>Posicione os modelos.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>Mude o tamanho dos modelos e suas rotações conforme achar melhor.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p>
<p><b>Jogando</b></p> <p>Crie uma sala de jogo.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>Comunique-se com os outros usuários conectados.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>Realize uma jogada de dados.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>Simule uma curta partida de RPG, narrando uma cena de combate.</p> <p>A tarefa foi realizada? ( ) Sim ( ) Não</p> <p>O ORC conseguiu simular uma mesa de jogo? ( ) Sim ( ) Não</p> <p>O ORC facilitou o seu trabalho como mestre? ( ) Sim ( ) Não</p>

Quadro 35 – Questionário geral sobre o ORC

<b>QUESTIONARIO GERAL SOBRE O ORC</b>
<p><b>Como foi utilizar o ORC?</b></p> <p>( ) Muito fácil.</p> <p>( ) Fácil.</p> <p>( ) Dificuldade media.</p> <p>( ) Difícil.</p> <p>( ) Muito difícil.</p>
<p><b>Qual a sua avaliação geral do ORC?</b></p> <p>( ) Ótimo.</p> <p>( ) Muito bom.</p> <p>( ) Bom.</p> <p>( ) Ruim.</p> <p>( ) Muito ruim.</p>
<p><b>O ORC é útil como ferramenta de RPG?</b></p> <p>( ) Sim.</p> <p>( ) Não.</p>
<p><b>Qual a sua opinião sobre os modelos 3D?</b></p> <p>( ) Ótimos.</p> <p>( ) Muito bons.</p> <p>( ) Bons.</p> <p>( ) Ruins.</p> <p>( ) Muito ruins.</p>
<p><b>Qual a sua opinião sobre o cenário?</b></p> <p>( ) Ótimo.</p> <p>( ) Muito bom.</p> <p>( ) Bom.</p> <p>( ) Ruim.</p> <p>( ) Muito ruim.</p>

**Você gostou do ORC?**

Sim.

Não.