



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS  
PRÓ-REITORIA DE ENSINO  
CÂMPUS GOIÂNIA  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

CHRISLEY WASHINGTON AUGUSTO DA SILVA SANTOS

# **Sistema WEB gerencial para o jogo Mysthen RPG**

**Automatização das mecânicas de um RPG autoral**

Goiânia, 2023.

**TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO  
NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG**

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem resarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

**Identificação da Produção Técnico-Científica**

- |  |   |
|--|---|
| <input type="checkbox"/> Tese  | <input type="checkbox"/> Artigo Científico              |
| <input type="checkbox"/> Dissertação                                 | <input type="checkbox"/> Capítulo de Livro              |
| <input type="checkbox"/> Monografia - Especialização                 | <input type="checkbox"/> Livro                          |
| <input checked="" type="checkbox"/> TCC - Graduação                  | <input type="checkbox"/> Trabalho Apresentado em Evento |
| <input type="checkbox"/> Produto Técnico e Educacional - Tipo: _____ |   |

Nome Completo do Autor: Chrisley Washington Augusto da Silva Santos  
Matrícula: 20172011090084

Título do Trabalho: Automatizando o Mysthen RPG

**Autorização - Marque uma das opções**

1. ( ) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. ( ) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data \_\_\_\_/\_\_\_\_/\_\_\_\_ (Embargo);
3. ( X ) Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção 2 ou 3, marque a justificativa:

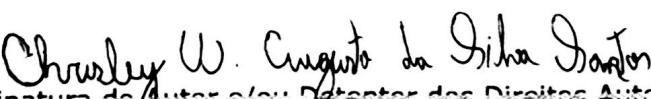
- ( X ) O documento está sujeito a registro de patente.  
( X ) O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.  
( ) Outra justificativa: \_\_\_\_\_

**DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA**

O/A referido/a autor/a declara que:

o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade; obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue; cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Goiânia GO, 04/07/2023.  
Local Data

  
Assinatura do Autor e/ou Detentor dos Direitos Autorais

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS

PRÓ-REITORIA DE ENSINO

CÂMPUS GOIÂNIA

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

CHRISLEY WASHINGTON AUGUSTO DA SILVA SANTOS

# **Sistema WEB gerencial para o jogo Mysthen RPG**

**Automatização das mecânicas de um RPG autoral**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Bacharelado em Sistemas de Informação como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

**Orientador:** Prof. Dr. Frederico Nogueira Leite

**Coorientadora:** Profa. Ma. Carina Calixto Ribeiro de Araujo

Goiânia, 2023.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS  
PRÓ-REITORIA DE ENSINO  
CÂMPUS GOIÂNIA  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

CHRISLEY WASHINGTON AUGUSTO DA SILVA SANTOS

# **Sistema WEB gerencial para o jogo Mysthen RPG**

**Automatização das mecânicas de um RPG autoral**

Trabalho de Conclusão de Curso defendido no Curso de Bacharelado em Sistemas de Informação como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação, aprovada em 04 de Julho de 2023, pela Banca Examinadora constituída pelos professores:

---

**Prof. Dr. Frederico Nogueira Leite**  
Departamento IV - IFG / Câmpus Goiânia  
Presidente da Banca

---

**Profa. Ma. Carina Calixto Ribeiro de Araujo**  
Departamento IV - IFG / Câmpus Goiânia

---

**Prof. Me. Renan Rodrigues de Oliveira**  
Departamento IV - IFG / Câmpus Goiânia

# Dedicatória

Dedico aos meus amigos e familiares parceiros de mesa, por compartilharem conigo inúmeras aventuras de RPG. Juntos exploramos castelos, lutamos batalhas, desvendamos mistérios, e demos vida a vários personagens em histórias épicas e emocionantes. Esses momentos de diversão e conexão me motivaram a explorar mais a fundo os elementos que compõem esse jogo tão especial, me inspirando a criar meu próprio jogo e, com este trabalho, uma aplicação capaz de gerencia-lo.

Também dedico especialmente a minha companheira, Eliza Maria Bernardes, pelo constante apoio emocional e incentivo, além da paciência e compreensão com a minha ausência ao longo dos meses em que estive ocupado com este trabalho. Dedico essa conquista a você, por estar sempre ao meu lado.

Chrisley Washington Augusto da Silva Santos

# **Agradecimentos**

Gostaria de expressar meus sinceros agradecimentos aos meus professores e orientador, cujos ensinamentos foram fundamentais para minha formação acadêmica. Aos meus pais, avós, companheira, filho, irmão e demais familiares por seu apoio, incentivo e encorajamento. A todos os mencionados, meu sincero agradecimento por terem contribuído para o sucesso deste trabalho.

Agradecimento especial ao meu supervisor, Samuel Rosa Veras, por sua valiosa assistência, orientação e amizade. Suas contribuições foram essenciais para o aprimoramento deste trabalho, portanto, sou imensamente grato pela cooperação.

Chrisley Washington Augusto da Silva Santos

# Resumo

**Título:** Sistema WEB gerencial para o jogo Mysthen RPG

**Autor:** Chrisley Washington Augusto da Silva Santos

**Orientador:** Dr. Frederico Nogueira Leite

O objetivo principal deste trabalho consistiu no desenvolvimento de uma aplicação web voltada para a gestão e otimização de um jogo de RPG, com a transposição de atividades analógicas para o meio digital. Foram abordados conceitos fundamentais sobre RPG, incluindo seu funcionamento, bem como as regras e mecânicas específicas do jogo autoral adaptado pela aplicação, o "Mysthen RPG".

A proposta visou planejar e implementar uma aplicação capaz de abstrair e automatizar as mecânicas e regras de um jogo de RPG de mesa específico, proporcionando aos jogadores uma maior praticidade e eficiência no gerenciamento de mecânicas de jogo, eliminando cálculos manuais e possibilitando a automatização de tarefas.

Ao adaptar o jogo de RPG de mesa para o meio digital, exploram-se as vantagens oferecidas pela tecnologia, proporcionando uma interface intuitiva, interativa e de fácil utilização. Dessa forma, a aplicação proporciona uma experiência aprimorada de jogo e abre caminho para a possibilidade de exploração de um novo modelo de jogo, que combina elementos analógicos e digitais de forma inovadora.

## Palavras-chave

sistema web, automatização, otimização, RPG de mesa, jogo autoral

# Abstract

**Title:** Automating the Mysthen RPG.

**Author:** Chrisley Washington Augusto da Silva Santos

**Advisor:** Dr. Frederico Nogueira Leite

This work presents the development of a web application for the management and optimization of an RPG game, aiming to abstract the game mechanics to enhance the players' experience. The proposal is based on the need to utilize computational resources to evolve and preserve ludic and cultural activities, following market trends.

The web application was developed to manage sessions of the “Mysthen RPG”, an unpublished self-authored game. The business rules and project scope were derived from this game, with a focus on character creation and automation of exploration rules.

The planning and development of the application involved the use of appropriate technological tools, such as PostgreSQL, Java, SpringBoot, JavaScript, VueJs, Bootstrap-Vue, and SSE, to enable the project. The implementation followed the agile Kanban methodology, utilizing platforms like Jira and BitBucket for project management and version control.

The work demonstrated the author's skills as a software developer, attesting to their ability to meet the proposed requirements and provide solutions to the established problems. Thus, the project stands as evidence of the author's competencies in software development.

## Keywords

Web application, RPG game, analog-to-digital transposition, abstraction of mechanics, Kanban methodology

# **Lista de Figuras**

2.1	Sessão de RPG no quadro “CelebriD&D”, na imagem Marisha Ray, Ashly Burch, Terry Crews e Matt Mercer . . . . .	22
2.2	Livro de regras do RPG <i>Star Wars: Force and Destiny</i> . . . . .	25
2.3	Trecho exemplificando a atuação de mestre e jogador no livro de regras do “ <i>Dungeons and Dragons</i> ” . . . . .	26
2.4	Livros do RPG “ <i>Dungeons and Dragons</i> ” . . . . .	28
2.5	Ficha do RPG “ <i>Cyberpunk</i> ” . . . . .	29
2.6	Dados de 20, 10, 8, 6 e 12 faces . . . . .	30
2.7	Miniaturas sobre um painel quadriculado . . . . .	30
2.8	Sessão remota de RPG transmitida no YouTube . . . . .	33
2.9	Fluxo de jogo . . . . .	34
4.1	Quadro Kaban da aplicação no Jira Software . . . . .	62
4.2	Quadros concluídos . . . . .	63
4.3	Diagrama Entidade-Relacionamento da aplicação . . . . .	64
4.4	Tabelas implementadas do esquema “Game Data” . . . . .	65
4.5	Tabelas implementadas do esquema “Personagem” . . . . .	66
4.6	Tabelas implementadas do esquema “Capacidades” . . . . .	67
4.7	Tabelas implementadas do esquema “Capacidades” . . . . .	68
4.8	Tabelas implementadas do esquema “Sala” . . . . .	69
4.9	Tabelas implementadas do esquema “Auditoria” . . . . .	70
4.10	Trecho da classe WebSecurityConfig no servidor . . . . .	74
4.11	Registros de usuários salvos na tabela “usuarios” . . . . .	75
4.12	Código de abertura da conexão SSE. . . . .	77
4.13	Código do controller SSE. . . . .	78
4.14	Método updateSala no servidor. . . . .	79
5.1	Tela de cadastro . . . . .	83
5.2	Tela principal da aplicação . . . . .	84
5.3	Tela principal da aplicação em dispositivo mobile . . . . .	85
5.4	Exibição das opções do menu lateral . . . . .	86
5.5	Cabeçalho da tela de criação de personagens . . . . .	87
5.6	Aba informativa da tela de criação de personagens . . . . .	87
5.7	Relação de capacidades do personagem em criação . . . . .	88
5.8	Relação de capacidades do personagem em criação em dispositivo móvel	88
5.9	Primeira etapa da criação de personagem . . . . .	89
5.10	Segunda etapa da criação de personagem . . . . .	90
5.11	Terceira etapa da criação de personagem . . . . .	91
5.12	Trecho do controller CriaPersonagem, no servidor . . . . .	91

5.13	Quarta etapa da criação de personagem . . . . .	92
5.14	Quinta etapa da criação de personagem . . . . .	93
5.15	Quinta etapa da criação de personagem . . . . .	94
5.16	Trecho do ControllerPersonagem . . . . .	95
5.17	Interface de listagem de personagens . . . . .	95
5.18	Interface de criação de salas . . . . .	96
5.19	Interface de listagem de salas . . . . .	96
5.20	Painel do jogador . . . . .	97
5.21	Painel do jogador em dispositivo Mobile . . . . .	98
5.22	Aba lateral de redutores . . . . .	98
5.23	Aba lateral de status . . . . .	99
5.24	Demonstração da aba de perfil . . . . .	99
5.25	Demonstração da aba descritiva de capacidades . . . . .	100
5.26	Painel do Mestre . . . . .	100
5.27	Aba lateral da galeria . . . . .	101
5.28	Aba lateral de anotações . . . . .	101
5.29	Aba lateral de inserção de personagens do mestre . . . . .	102
5.30	Aba lateral de inserção de personagens do mestre . . . . .	102
5.31	Aba lateral de inserção de personagens do mestre . . . . .	103
5.32	Painel do mestre com histórico da sala preenchido. . . . .	103
5.33	Demonstração de uma solicitação de acesso. . . . .	105
5.34	Solicitação de acesso em espera. . . . .	105
5.35	Solicitação de acesso recebida. . . . .	106
5.36	Novo personagem adicionado. . . . .	106
5.37	Exemplo de teste de capacidade. . . . .	107
5.38	Trecho do ControllerTeste. . . . .	108
5.39	Registro do teste recebido. . . . .	108
5.40	Interface de solicitação de teste. . . . .	109
5.41	Confirmação da solicitação de teste. . . . .	110
5.42	Resultado da solicitação de teste. . . . .	110
5.43	Adversidade acumulada. . . . .	111
A.1	Pontos do personagem Lammar Hyen . . . . .	120
A.2	Pontos da personagem Talyria’Yamari . . . . .	121
A.3	Pontos do personagem Malvin Malori . . . . .	122
B.1	Diagrama Entidade-Relacionamento referente ao banco de dados modelado para a aplicação . . . . .	133

# **Lista de Tabelas**

2.1 Visão geral sobre as “etapas pré-jogo” de um RPG . . . . .	27
--	----

# **Lista de Abreviaturas e Siglas**

APIs	Application Programming Interfaces
CSS	Cascading Style Sheets
d20	dado de 20 faces
D&D	<i>Dungeons and Dragons</i>
DTO	Data Transfer Object
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
JDBC	Java Database Connectivity
JPA	<i>Java Persistence API</i>
JWT	JSON Web Token
MVC	Model-View-Controller, Nível de Dificuldade
	Password-Based Key Derivation Function 2
	Programação Orientada a Objetos
	<i>Role Playing Games</i>
	Sistema de Gerenciamento de Banco de Dados
	Server-Sent Events
	<i>Unified Modeling Language</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
1.1	Motivação	19
1.2	Objetivos	20
1.2.1	Objetivo Geral	20
1.2.2	Objetivo Específico	20
1.3	Metodologia	20
1.4	Descrição dos Capítulos	21
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>22</b>
2.1	O que é o RPG	22
2.1.1	Narração	23
2.1.2	Regras	24
2.1.3	Cenário	25
2.2	Visão Geral do funcionamento de um RPG	26
2.3	Etapas pré-jogo	27
2.3.1	Obtenção de um RPG	28
2.3.1.1	Livro de regras	28
2.3.1.2	Ficha	29
2.3.1.3	Kit de Dados	29
2.3.1.4	Miniaturas	30
2.3.2	Participantes	31
2.3.2.1	Mestre	31
2.3.2.2	Jogador	31
2.3.3	Criação de Personagens	32
2.3.4	Planejamento da Aventura	33
2.3.5	Preparativos	33
2.4	Etapas do jogo	34
2.4.1	Introdução	35
2.4.2	Cena	36
2.4.3	Interação	36
2.4.4	Reação	37
2.5	Fim do jogo	37
2.6	Observação	37
<b>3</b>	<b>APRESENTAÇÃO DO “MYSTHEN RPG”</b>	<b>38</b>
3.1	Visão Geral	38
3.1.1	Justificativa	39

3.1.2	Considerações sobre o sistema de regras . . . . .	40
3.1.3	Considerações sobre o cenário . . . . .	42
3.2	Recorte de regras implementadas . . . . .	43
<b>4</b>	<b>FERRAMENTAS E TÉCNICAS . . . . .</b>	<b>44</b>
4.1	Requisitos . . . . .	44
4.1.1	Requisitos Funcionais . . . . .	45
4.1.1.1	Cadastro e Usuários . . . . .	45
4.1.1.2	Personagem . . . . .	45
4.1.1.3	Sala . . . . .	46
4.1.1.4	Interações . . . . .	46
4.1.2	Requisitos Não Funcionais . . . . .	47
4.2	Escopo . . . . .	48
4.2.1	Acesso, login e senha . . . . .	48
4.2.2	Criação de personagem . . . . .	49
4.2.3	Criação de sala . . . . .	49
4.2.4	Painel de jogo - Jogador/Personagem . . . . .	50
4.2.5	Painel de jogo - Mestre/Sala . . . . .	51
4.3	Casos de Uso . . . . .	51
4.3.1	Realizar Cadastro de Usuário . . . . .	52
4.3.2	Realizar Login . . . . .	53
4.3.3	Criar Personagem . . . . .	54
4.3.4	Criar Sala . . . . .	55
4.3.5	Acessar painel de jogo com um personagem . . . . .	56
4.3.6	Acessar painel de jogo de uma sala . . . . .	57
4.3.7	Solicitar acesso à sala de jogo . . . . .	58
4.3.8	Conceder acesso à sala de jogo . . . . .	59
4.3.9	Realizar teste de capacidade . . . . .	60
4.3.10	Solicitar teste de resistência . . . . .	61
4.4	Metodologia Ágil . . . . .	62
4.5	Banco de Dados . . . . .	64
4.5.1	Esquema Game Data . . . . .	65
4.5.2	Esquema Personagem . . . . .	66
4.5.3	Esquema Capacidade . . . . .	67
4.5.4	Esquema Cadastro . . . . .	68
4.5.5	Esquema Sala . . . . .	69
4.5.6	Esquema Auditoria . . . . .	70
4.5.7	Considerações . . . . .	70
4.6	Implementação . . . . .	71
4.6.1	Back-End . . . . .	71
4.6.1.1	Banco de dados . . . . .	71
4.6.1.2	Servidor . . . . .	72
4.6.1.3	Dependências . . . . .	73
4.6.1.4	JWT . . . . .	74
4.6.1.5	UtilPassword . . . . .	75
4.6.1.6	Convenções de código . . . . .	75
4.6.1.7	Server-Sent Events . . . . .	77

4.6.2	Front-End . . . . .	80
4.6.2.1	VueJS . . . . .	80
4.6.2.2	Estilização . . . . .	81
4.6.2.3	Elementos temporários . . . . .	82
<b>5</b>	<b>RESULTADOS . . . . .</b>	<b>83</b>
5.1	Acesso ao sistema . . . . .	83
5.2	Tela Principal . . . . .	84
5.3	Opções de navegação . . . . .	86
5.4	Criação de Personagem . . . . .	87
5.5	Listagem de Personagem . . . . .	95
5.6	Criação de Salas . . . . .	96
5.7	Lista de Salas . . . . .	96
5.8	Painel de jogo do Personagem . . . . .	97
5.9	Painel de jogo da Sala . . . . .	100
5.10	Interações em tempo real . . . . .	104
5.10.1	Gerenciamento de solicitações de acesso . . . . .	104
5.10.2	Realização de testes Capacidade . . . . .	107
5.10.3	Solicitação de testes de Resistência . . . . .	109
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>112</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS . . . . .</b>		<b>114</b>
<b>APÊNDICE A – SIMULAÇÃO DE UMA PARTIDA DO “MYSTHEN RPG” . . . . .</b>		<b>117</b>
A.1	Obtenção do jogo e participantes na simulação . . . . .	117
A.2	Planejamento da aventura simulada . . . . .	118
A.3	Criação de personagens . . . . .	119
A.3.1	Jogador 01 - Personagem Lammar Hyen . . . . .	120
A.3.2	Jogador 02 - Personagem Talyria’Yamari . . . . .	121
A.3.3	Jogador 03 - Personagem Malvin Malori . . . . .	122
A.4	Preparativos da partida . . . . .	123
A.5	Introdução da trama . . . . .	124
A.6	Dinâmica do jogo . . . . .	125
A.6.1	Cena . . . . .	125
A.6.2	Interação . . . . .	125
A.6.3	Reação . . . . .	125
A.7	Atuação . . . . .	126
A.8	Aplicação de regras . . . . .	127
A.8.1	Solicitação e realização de teste . . . . .	127
A.8.2	Conclusão e narração do teste . . . . .	127
A.8.3	Iniciando um Combate . . . . .	128
A.8.4	Ordem de Combate . . . . .	128
A.8.5	Turno de Combate . . . . .	128
A.8.6	Ataque . . . . .	128
A.8.7	Postura . . . . .	129
A.8.8	Contra ataque . . . . .	129
A.8.9	Deslocamento em combate . . . . .	129

A.8.10	Ferimento . . . . .	130
A.8.11	Fim do primeiro turno . . . . .	130
A.8.12	Troca de Turnos . . . . .	130
A.8.13	Morte . . . . .	131
A.8.14	Conclusão da simulação . . . . .	131
<b>APÊNDICE B – DIAGRAMA ENTIDADE-RELACIONAMENTO . . . . .</b>		<b>133</b>

# 1 INTRODUÇÃO

É comum que se entenda os jogos apenas como forma de lazer e entretenimento. Entretanto, pouco se discute sobre o valor histórico e cultural de tais atividades. De maneira geral, os jogos surgem como abstração lúdica do contexto de vida das pessoas em sociedade, imitando ou ressignificando suas ferramentas, cerimônias, profissões, entre outros aspectos cotidianos (CREPALDI, 2010). Sob essa ótica, os jogos são formas de expressão cultural, como as artes, folclore e crenças de um povo.

Ligado ao desenvolvimento das ciências, por meio do jogo é possível compreender melhor o ser humano, sua individualidade, sua diversidade, as relações que desenvolve e, por consequência, a construção da sociedade em que vivemos (CREPALDI, 2010, p. 9).

Um exemplo do papel dos jogos como expressão do imaginário popular é o “Senat”, um jogo de tabuleiro que ilustra simbolicamente a passagem para o mundo dos mortos dentro das antigas crenças egípcias, o qual coloca o jogador na posição de enfrentar o próprio destino em partidas repletas de alegorias. Nesse mesmo sentido, as crenças astrológicas dos Sumérios marcam presença nas casas do tabuleiro do “jogo real de Ur”, que data de aproximadamente 2500 a.C. (CUSTÓDIO; SOUZA, 2019).

Quanto produto de um contexto social que está sempre em transformação, os jogos estão sujeitos a eventuais modificações ou adaptações, podendo inclusive surgir ou desaparecer eventualmente. O citado “Senat” é um exemplo de jogo extinto que foi redescoberto por trabalhos arqueológicos. Por outro lado, existem indícios de que o “Jogo Real de Ur” teria alguma ancestralidade comum ao jogo de “Gamão”, praticado desde a Roma de 600 a.C. até os dias atuais, mediante alterações nas regras (CREPALDI, 2010).

Uma forma de analisar o processo de evolução e diversificação dos jogos é olhando para a trajetória dos *Role Playing Games* (RPG), ou “Jogos de interpretação de papéis” em tradução livre. Uma modalidade de jogo que nasceu herdando características estruturais de outra, os *WarGames*, ou “Jogos de Guerra”.

Se aprofundando nessa questão, tem-se que os jogos de guerra descendem de exercícios estratégicos realizados em contexto militar. A prática é embasada em conhecimento tático real, e busca representar um cenário hipotético onde é preciso analisar as possibilidades de ação e tomar decisões estratégicas, se atentando às limitações e dificuldades definidas, para cumprir objetivos, como tomar um território, proteger uma região, buscar determinado recurso, etc (PRIETTO, 2013).

Esse tipo de jogo se popularizou como entretenimento principalmente nos EUA do século vinte, colocando os jogadores na posição de comando de tropas, cujas unidades eram representadas por miniaturas dispostas em um mapa, permitindo que os entusiastas do tema pudessem emular batalhas e guerras reais, hipotéticas ou fantásticas (PRIETTO, 2013). É válido observar que os maiores conflitos armados da história da humanidade ocorreram nesse período, o que fora acompanhado de intensa propaganda pró-guerra, com potencial influência no imaginário popular.

Entretanto, outros fatores criavam tendências em paralelo, é o caso da conjuntura literária da época, com histórias curtas de fantasia sendo consumidas popularmente por meio de revistas bastante acessíveis. Um exemplo que remete a esse contexto são as famosas histórias de “Conan, o Bárbaro” publicadas a partir de 1932 por Robert E.Howard. Outro autor marcante foi J.R.R.Tolkien, com seus livros de alta fantasia, como “O Hobbit” de 1937 e a trilogia “O Senhor dos Anéis” publicada a partir de 1954 (PRIETTO, 2013).

Dado esse cenário, a modalidade de jogos “RPG” nasce na década de setenta, com regras derivadas dos *WarGames*, mas abordando a temática de fantasia tão presente na literatura da época (PRIETTO, 2013). Os autores, Gary Gygax e Dave Arneson, adaptaram as regras para traspor a perspectiva do jogador, trocando o contexto de guerra onde se controla inúmeras tropas, para uma abordagem focada no gerenciamento e interpretação de apenas um personagem.

Assim, diversos elementos foram herdados de um jogo para o outro, como o uso de miniaturas marcando posições num mapa, rolagens de dados para adicionar risco e imprevisibilidade, tabelas e regras que definem inúmeras ações possíveis, etc. Outra característica adaptada é a figura do “juiz”, um jogador com papel arbitrário e grande conhecedor das regras, que ajuda no gerenciamento das partidas de *WarGames* (PRIETTO, 2013). No RPG esse jogador foi adaptado para a figura do “mestre”, que atua tanto como mediador quanto como narrador, descrevendo todos os elementos do cenário.

O jogo *Dungeons and Dragons* (D&D) foi o primeiro da modalidade RPG, publicado em 1974 pelos próprios Gary Gygax e Dave Arneson. Nessa primeira versão o jogo se baseava na busca por tesouros dentro de locais perigosos, importando o clima dos textos de *Sword and Sorcery*, ou “Espada e Feitiçaria”, da literatura já mencionada (PRIETTO, 2013).

Dessa forma, a proposta do jogo era a de um grupo de personagens, cada um controlado por um jogador, que se reúne para explorar calabouços e masmorras, num cenário de fantasia medieval, enfrentando inimigos, armadilhas e enigmas em busca de alguma recompensa. Vale comentar que a proposta e sistema de regras do D&D foram se transformando com o tempo, de forma que hoje o jogo se encontra em sua quinta edição, com a sexta já anunciada para 2024 (RANGEL, 2021).

Demonstrando novamente como os jogos se diversificam, temos outros autores que publicaram seus próprios RPGs, trazendo novas mecânicas e explorando cenários para além da fantasia medieval, com temáticas futuristas, de terror, exploração espacial, mitologia, super-heróis, e até mesmo em universos de grandes produções como *StarWars* e *Matrix*. Um exemplo é o RPG “Call of Cthulhu”, publicado pela primeira vez em 1981 pela editora *Chaosium*, inspirado nos contos de Horror cósmico do escritor H.P. Lovecraft (RAPHAEL, 2020). Outro exemplar é o “Lumenera”, um RPG que mescla fantasia e ficção científica, escrito por Monte Cook, após uma campanha de financiamento coletivo que arrecadou US\$ 517.255,00 em 2015 (MTELLES, 2015).

A proposta dos jogos dessa modalidade também se expandiu, se voltando para a produção coletiva de uma trama ficcional como objetivo mais comum. Isso porque o cenário e regras de um jogo de RPG viabilizam a elaboração de um enredo com objetivos e desafios, além de dispor mecânicas para criação de personagens. Quando o jogo se inicia, o mestre apresenta uma trama em que os personagens estão inseridos, direcionando seus objetivos e descrevendo todos os elementos que compõe a cena em curso, uma vez motivados e imersos na narração os jogadores são livres para tomar suas decisões através do personagem, interagindo livremente com o ambiente, itens, inimigos, aliados, etc.

Dessa maneira, o entretenimento se dá pela imersão, consumo e produção de uma história criada em tempo real, cujo desenrolar é imprevisível tanto para o mestre quanto para os jogadores, uma vez que a liberdade de escolhas, vinculada às mecânicas de jogo, promovem resultados indeterminados, capazes de gerar todo tipo de reviravolta. Assim, o livro de regras de um RPG fornece um cenário fértil para a elaboração de aventuras e um conjunto de regras que mantém a coesão do mundo ficcional imaginado, limitando as capacidades dos personagens e mantendo o nível de desafio.

Outro aspecto da expansão e evolução da modalidade é a relação que surge entre o RPG e os recursos tecnológicos contemporâneos. Um exemplo, é o uso de aplicativos de mensagens e redes sociais para conhecer outros jogadores e marcar as partidas, ou mesmo a possibilidade de efetivamente jogar RPG por video-chamadas. Entusiastas costumam criar e divulgar conteúdo digital em torno do RPG, publicando discussões, apresentações ou gravações de sessões de jogo em blogs, canais de “*YouTube*”, podcasts, transmissões ao vivo entre outras mídias digitais.

Portanto, se tornou comum que recursos digitais sejam usados como ferramentas de apoio, por iniciativa dos próprios jogadores. Adicionalmente, a produção e comercialização de RPGs também acompanham as demandas contemporâneas, dessa forma, é comum que existam sites dedicados a jogos específicos ou mesmo que o “livros de regras” de determinados jogos sejam distribuídos em mídia digital, o que não apenas facilita a disseminação e acesso ao material, como viabiliza que jogos com menor orçamento sejam conhecidos por entusiastas do mundo todo.

## 1.1 Motivação

Embora pontuada a modernização nesse segmento, é notável que a tecnologia é aplicada em elementos adjacentes ao jogo, como a comunicação entre os jogadores ou a distribuição do material. Portanto, é natural que se busque um melhor aproveitamento dos recursos tecnológicos nesse contexto, uma vez que, atualmente, as ferramentas de desenvolvimento, principalmente no cenário web e mobile, viabilizam o desenvolvimento de novas soluções, as quais poderiam abordar aspectos mais intrínsecos da atividade.

Ou seja, apesar do emprego de recursos computacionais, o RPG ainda se trata de um jogo majoritariamente analógico, uma vez que os elementos diretamente relacionados a atividade continuam sujeitos a capacidade humana, como a memorização de regras, aplicação de mecânicas, realização de cálculos, anotações manuais, etc.

Por exemplo, durante a criação de personagem cabe ao jogador anotar, relacionar e calcular valores manualmente, um processo que pode ser considerado moroso e pouco intuitivo, principalmente para iniciantes. Um sintoma dessa problemática é a comum necessidade de auxílio de um veterano para viabilizar a introdução de novos jogadores.

Outro problema é a quantidade elevada de regras a serem aprendidas, memorizadas e aplicadas pelos jogadores. Sendo comum que os mesmos negligenciem, se equivoquem ou se esqueçam de algumas regras do jogo, gerando atritos e interrupções para consulta de regras, criando quebras desnecessárias do ritmo e imersão da narrativa.

No que tange a aplicação de regras, é comum que ocorram rolagens de dados, seguidas de cálculos matemáticos e comparações de valores para obtenção de resultados numéricos. Esses procedimentos são realizados pelos próprios jogadores, estando sujeitos a erros e trapaças, além de causarem interrupções no fluxo narrativo, o qual só poderá ser retomado quando o valor calculado for obtido, interpretado e inserido na trama.

Diante disso, tem-se que, embora o foco e objetivo de um RPG seja a narrativa e a interação, a necessidade de controle manual das mecânicas de jogo comumente desfavorece o cumprimento da proposta da atividade. Existem, portanto, muitos elementos a serem compreendidos, memorizados, aplicados, relacionados, comparados e consultados pelos participantes, o que torna o jogo mais difícil de ser aprendido e executado, além de mais sujeito a falhas e interrupções.

Portanto, a proposta deste trabalho é apresentar o projeto e a implementação de um software que opere quanto plataforma digital para um jogo de RPG, com foco na abstração das mecânicas para aprimoramento da experiência. Tendo em vista a tendência “evolutiva” dos jogos, que acompanha as condições culturais e tecnológicas da sociedade, fica claro que o uso de recursos computacionais para aprimorar a experiência em atividades lúdicas e culturais é tão natural, quanto necessária para preservação e evolução das mesmas.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Desenvolver uma aplicação web gerencial para partidas do jogo “Mysthen RPG”.

### 1.2.2 Objetivo Específico

- Adaptar jogo de RPG analógico para o meio digital.
- Desenvolver aplicação web responsiva para múltiplos dispositivos.
- Implementar mecanismos de comunicação entre usuários em dispositivos distintos.
- Automatizar cálculos e mecânicas analógicas de jogo.

## 1.3 Metodologia

O primeiro passo para atingir os objetivos foi analisar o jogo “Mysthen RPG”, e definir um recorte das regras para adaptar, diante da inviabilidade de se trabalhar com as regras em sua totalidade. Extraí-se dessa análise as regras de negócio da aplicação, base a partir da qual foi realizado o planejamento do sistema, incluindo a definição do escopo, requisitos, modelagem do banco, entre outros elementos.

Em sequência, foi definida a estrutura da aplicação, formada por três componentes principais, um banco de dados para persistência das informações, um servidor que centraliza as requisições e gerencia o trâmite de dados, e as interfaces para interação do usuário, executadas nos dispositivos clientes.

Para definir as ferramentas utilizadas para construir cada parte da referida estrutura, foram realizadas pesquisas e análises de recursos tecnológicos. As principais ferramentas selecionadas foram, o “*postgresql*” no banco de dados, “*Java*” e “*SpringBoot*” no servidor e “*JavaScript*”, “*VueJs*” e “*Bootstrap-Vue*” nas interfaces.

Uma vez definido o que deve ser implementado e com quais ferramentas, foi decidido como a implementação do sistema seria gerenciada. Para isso, foi selecionada a metodologia “*Kanban*”, que opera de forma ágil na administração de demandas, para definição e administração dos objetivos de implementação. A plataforma “*Jira*” foi selecionada para gerenciamento da metodologia ágil, uma vez que disponibiliza painéis para manipulação dos quadros do “*Kanban*”.

Dessa forma, a partir da análise das regras de negócio a aplicação foi projetada e estruturada. As ferramentas empregadas no trabalho foram definidas, assim como a metodologia de gerenciamento da aplicação. Seguindo tal procedimento a aplicação pôde ser de fato implementada.

## 1.4 Descrição dos Capítulos

Essa sessão apresenta a estrutura básica do trabalho, resumindo o conteúdo que consta em cada um dos capítulos principais.

**01 Introdução:** Apresentação introdutória do tema que motiva a existência do trabalho, levantamento das problemáticas que derivam do assunto apresentado, definição dos objetivos a serem cumpridos para solucionar a problemática, descrição do processo a ser realizado para atingir os objetivos propostos e explicação da estrutura geral do trabalho.

**02 Revisão Bibliográfica:** Definição geral de RPG e seus principais elementos, descrição do funcionamento de jogos da modalidade RPG, abordando as etapas que ocorrem antes e durante a atividade para que seja possível compreender o que é e como funciona um jogo dessa modalidade.

**03 Apresentação do “Mysthen RPG”:** Apresenta o jogo específico adaptado na aplicação, abordando suas as regras e mecânicas de funcionamento. Nesta sessão apenas a parcela do jogo inclusa no escopo do sistema será abordada, permitindo a compreensão das mecânicas nas quais são embasadas as funcionalidades da aplicação. Adicionalmente, esta sessão conta com comentários para pontuar como o software se propõe a representar os comportamentos apresentados.

**04 Ferramentas e Técnicas:** Levantamento das ferramentas e técnicas utilizadas no planejamento e implementação da proposta, mediante análise e justificativa. Apresentação dos artefatos de projeto e principais referências produzidos.

**05 Resultados:** Apresentação da aplicação implementada, demonstrando suas interfaces e lógica de funcionamento. Este capítulo demonstra passo a passo a utilização do software desenvolvido, exibindo como o mesmo atua na resolução das problemáticas levantadas.

**06 Conclusão:** Fechamento e apresentação do desfecho do trabalho, pontuando como os resultados obtidos contribuem para a resolução da problemática apontada, e comentando sobre expectativas alcançadas, desafios encontrados e objetivos concluídos.

## 2 REVISÃO BIBLIOGRÁFICA

Esta sessão tem por objetivo apresentar as características que definem um jogo da modalidade RPG, e descrever as etapas que devem ser cumpridas antes e durante a execução da atividade, dessa forma é possível compreender a fundo o assunto abordado pela aplicação proposta.

### 2.1 O que é o RPG.

A sigla RPG significa *Role Playing Game*, o que em tradução livre quer dizer “Jogo de interpretação de papéis”, e se refere a uma modalidade de jogo onde os participantes interpretam personagens fictícios dentro de uma trama ficcional. Os jogadores devem tomar decisões e trabalhar em equipe para que seus personagens progridam na trama, resolvendo conflitos e atingindo objetivos ficcionais propostos na narrativa. A figura 2.1 (BONNEY, 2018) ilustra uma partida de RPG.



**Figura 2.1:** Sessão de RPG no quadro “CelebriD&D”, na imagem Marisha Ray, Ashly Burch, Terry Crews e Matt Mercer

A literatura é um bom ponto de partida para se compreender o que é um RPG, uma vez que ambos se assemelham por promover uma experiência ficcional e imaginária, utilizando eventos, cenários, situações, personagens e outros componentes narrativos (PEREIRA, 2007). Portanto, entende-se, a princípio, que o RPG e a literatura são meios de entretenimento semelhantes, imergindo e engajando pessoas em uma trama dotada de reviravoltas, objetivos, tensão, drama, etc.

## 2.1.1 Narração

Na literatura, é comum que a narração ocorra em primeira ou terceira pessoa, de forma que o leitor se mantém passivo e externo à narrativa, consumindo um relato. Na primeira pessoa, o personagem faz um relato pessoal, descrevendo suas próprias ações, declarando “eu fiz”. Já na terceira pessoa o relato é de um observador, que descreve as ações do personagem, declarando “ele fez” (OLIVEIRA, 2018).

Porém, observando trechos demonstrativos que aparecem nos próprios livros de regras de vários jogos de RPG, ou acompanhando qualquer sessão de jogo em podcasts ou plataformas de vídeo, nota-se a recorrência de termos como “você fez”, “você ouviu” ou “vocês viram”, demonstrando uma forte presença da narração em segunda pessoa no RPG, algo incomum na literatura.

De repente, vocês todos vêem as portas da cantina correndo pelos trilhos assim que um grupo de brutamontes durões entra na sala. O garçom grita, “Ei! sem blasters aqui!”. Ele mergulha atrás do bar, bem a tempo, assim que um dos brutamontes despreocupadamente dispara um tiro de blaster em sua direção. (SLAVICSEK ANDY COLLINS, 2008, p. 16).

Como cada participante deve representar algum personagem na trama do jogo, pode-se dizer que o uso de “você” sobrepõe o indivíduo real que está jogando e o personagem fictício que está dentro da trama, se referindo a ambos em simultâneo. Dessa forma, a narração em segunda pessoa convida o jogador a se projetar na situação fictícia, através do personagem.

Outro aspecto dessa sobreposição é que os jogadores comumente se referem aos personagens na primeira pessoa, como falando de si próprios. Ao que é demonstrado no trecho que simula uma partida, no livro de regras básicas do RPG *Dungeons and Dragons* (MEARLS, 2014) “Phillip (jogando com Gareth): Eu quero olhar as gárgulas. Eu tenho um sentimento que elas não são apenas estátuas.”.

O trecho citado também demonstra que, ao se projetar na trama, através de um personagem, o jogador pode interagir com a ficção. Basicamente, os jogadores decidem o que seus personagens pensam, dizem, fazem e tentam fazer, e cabe a narrativa se adaptar a essas escolhas, fluindo dinamicamente entre elas.

Logo, a narração no RPG não pode ser estática, como na escrita, motivo pelo qual ela é expressa por via oral, o que exige que um dos jogadores assuma o papel de narrador (MEARLS, 2014). Dessa forma, a narrativa se apropria da fluidez da conversação, permitindo que o narrador questione “o que você fez?”, dando espaço para que o personagem, sob controle do jogador, tome suas decisões.

## 2.1.2 Regras

Está claro que o RPG consiste em uma experiência narrativa, em que se cria uma trama ficcional que se desenrola mediante as interações de personagens (AZEVEDO, 2019). Porém, a atividade não é tida apenas como um exercício dramatúrgico ou literário, e sim como um jogo, onde existem regras para definir, viabilizar e controlar as interações.

Uma vez que o jogo consiste na criação coletiva de um enredo, é natural que a todo momento surjam desafios, conflitos e objetivos narrativos, os quais precisam ser superados, resolvidos e alcançados pelos personagens. Para alcançar essas metas, os jogadores precisam tomar decisões sujeitas as regras do jogo, que mantém a coesão do mundo ficcional imaginado.

Ou seja, as regras definem as capacidades dos personagens, restringindo o que eles podem ou não fazer, elas também estabelecem parâmetros como, quando determinada ação pode ser realizada, quando ela será concluída, quantos metros um determinado movimento pode cobrir, qual o alcance máximo de determinada arma ou magia, etc.

O trecho a seguir, extraído do livro de regras do RPG *Dungeons and Dragons* quinta edição, demonstra uma regra do jogo, aplicada quando um jogador decide que seu personagem realizará um salto em distância.

Quando você realiza um salto em distância, você cobre uma distância igual ao seu valor de Força multiplicado por 0,3 metros, se ele percorrer no mínimo 3 metros antes do salto. Quando realiza um salto em distância sem percorrer 3 metros antes de saltar, você pode saltar apenas metade dessa distância. De qualquer forma, cada metro saltado custa um metro de deslocamento. (MEARLS, 2014, p. 92).

Também é comum que as regras definam se as ações realizadas pelos personagens serão bem sucedidas ou falhas. Esse processo costuma envolver rolagens de dados multifacetados, que geralmente possuem entre 4 e 20 faces (AZEVEDO, 2019). Isso adiciona uma camada de aleatoriedade que deixa o jogo ainda mais imprevisível, uma vez que não se sabe o que o jogador decidirá fazer, e muito menos se a ação escolhida será executada com sucesso ou não.

O RPG se trata de uma modalidade para a qual existem diversos jogos, como “*Dungeons and Dragons*”, “*Tormenta20*”, “*Pathfinder*”, “*Terra Devastada*”, “*Vampiro a Máscara*”, etc. Naturalmente, o conjunto de regras varia de acordo com o jogo em questão, visando atender suas particularidades. Portanto, os sistemas de regras dos jogos da modalidade podem se diferenciar em volume, complexidade, foco e, principalmente, em relação ao cenário abordado.

### 2.1.3 Cenário

Uma vez que o jogo possui um foco narrativo, é natural que a história se passe em algum cenário, ou seja, em um mundo ficcional com características próprias. Portanto, tal como ocorre em livros, ou filmes, os jogos de RPG se diversificam imensamente em relação ao cenário ficcional onde a história pode se passar. Dessa forma, a depender do jogo em questão, os jogadores podem experientar tramas em cenários de exploração espacial, velho oeste, mitologia, era vitoriana, super-heróis, etc.

O já mencionado “*Dungeons and Dragons*”, por exemplo, se passa em mundos de fantasia medieval, em que existe magia, dragões e outros elementos místicos (MEARLS, 2014). Já o “*GURPS Cyberpunk*” tem como cenário um mundo futurista, repleto de alta tecnologia, implantes biotecnológicos, etc. (MARQUES, 2003).

Uma vez que as regras definem formas de interação com o mundo ficcional, é natural que a situação desse mundo impacte no tipo de regra necessária. Logo, o cenário é um fator determinante para estabelecer as regras do jogo, que precisam considerar o nível tecnológico do mundo, as características das raças jogáveis, existência de elementos mágicos, entre outros fatores.

Por exemplo, o RPG *Force and Destiny* é ambientado no cenário de *Star Wars*, baseado nos filmes homônimos de fantasia espacial de George Lucas. Neste universo, os personagens podem pilotar naves e utilizar armas e habilidades, como aparecem nos filmes, logo, o jogo dispõe de regras que viabilizam tais façanhas.



**Figura 2.2:** Livro de regras do RPG *Star Wars: Force and Destiny*

Além disso, existem várias opções de raças alienígenas jogáveis, como consta na figura 2.2 (BUTLER, 2015), as quais compõe o cenário específico de *Star Wars* e não aparecem em outros RPGs.

Outro exemplo é o RPG *Call of Cthulhu*, baseado nos contos de horror cósmico do escritor H.P. Lovecraft. Nessas histórias os horrores presenciados pelos personagens podem levá-los a loucura. Por esse motivo, o jogo conta com regras para gerenciar a sanidade dos personagens (PETERSEN; WILLIS; MASON, 2015).

## 2.2 Visão Geral do funcionamento de um RPG

Uma vez conceituado o RPG e seus principais componentes, se torna possível compreender como o jogo de fato funciona. Existem etapas a serem cumpridas antes do jogo começar, como a criação de personagens, elaboração da aventura, etc. Neste trabalho, essas etapas preliminares são categorizadas como “etapas pré-jogo”, detalhadas na sessão 2.3. Posteriormente, os participantes, geralmente entre 3 e 6 pessoas, podem de fato se reunir e jogar, iniciando as “Etapas de jogo”, detalhadas na sessão 2.4.

Tradicionalmente, como demonstrado na figura 2.3 (MEARLS, 2014), um dos participantes assume o papel de mestre. Ele quem planeja a trama, para que durante o jogo possa introduzi-la e narrá-la. O mestre também opera como mediador, portanto ele deve ter domínio das regras. Os jogadores, no que lhes concerne, precisam criar seus personagens e interpretá-los durante o jogo, atuando e tomando as decisões que os aproximem de seus objetivos narrativos.

Mestre: Ok, um de cada vez. Phillip, o Gareth está olhando para as gárgulas?  
 Phillip: Sim. Tem alguma dica de que elas podem ser criaturas em vez de decoração?  
 Mestre: Faça um teste de Inteligência.  
 Phillip: Por acaso minha perícia Investigaçāo se aplica?  
 Mestre: Claro!  
 Phillip (rolando um 1d20): Pô! Sete.  
 Mestre: Elas parecem como decoração para você. E Amy, Riva está checando a ponte levadiça?

**Figura 2.3:** Trecho exemplificando a atuação de mestre e jogador no livro de regras do “Dungeons and Dragons”

Como mencionado no capítulo 1, o pioneiro “Dungeons and Dragons” é um dos pilares da modalidade, motivo pelo qual é possível extrair dele uma boa noção do funcionamento dos RPGs de maneira geral. Nesse sentido, ele pontua que a dinâmica do jogo ocorre em três etapas, que se repetem de maneira cíclica, fazendo com que um novo fragmento da história se construa a cada laço (MEARLS, 2014).

O primeiro passo seria a narração de uma cena, que apresenta possibilidades de interação e situa os jogadores em relação ao ambiente e circunstâncias em que os personagens se encontram. Na segunda etapa, os jogadores determinam o que seus personagens disseram, pensaram, fizeram ou tentaram fazer na situação. E na terceira, as decisões são incorporadas na narrativa, causando reações no cenário que afetam a trama, ditando seus rumos.

Por exemplo, em uma sessão hipotética, os personagens dos jogadores estão sendo perseguidos por autoridades devido a algum delito, e decidem se esconder em um quarto de hotel. O mestre narra que os guardas que os perseguem estão batendo na porta desse quarto. Cabe aos jogadores decidir o que fazer, seja negociar, se render, combater, etc. Dada a escolha dos jogadores o mestre narra a continuidade da cena.

Vale ressaltar que, eventualmente, as ações dos personagens e reações do cenário podem estar sujeitas as regras do jogo, como mencionado na sessão 2.1.2. Outro elemento a ser pontuado, é que por mais que se trate de um jogo, onde existem regras e participantes em posições distintas, o RPG possui um forte teor cooperativo, e não competitivo, como enfatizado por Azevedo, no trecho que se segue.

O jogo promove uma interação em comum dos jogadores, incentivando a colaboração entre si, não havendo dessa forma uma competição. O mestre não compete contra os jogadores e nem os jogadores competem entre si. No final não há vencedores ou perdedores, apenas a boa experiência de jogar. (AZEVEDO, 2019, p. 22).

Quanto ao objetivo do jogo, este está contido na própria narrativa, ou seja, as motivações e propósitos partem dos próprios personagens e da trama. São exemplos, resgatar uma pessoa, explorar um local, derrotar um vilão, etc. Como mencionado no próprio livro de regras do “*Dungeons and Dragons*” (MEARLS, 2014, p. 4) “ O grupo pode não conseguir completar uma aventura, mas se todos tiveram um bom tempo juntos e criaram uma história memorável, então todos ganharam”.

## 2.3 Etapas pré-jogo

Esta sessão detalha das etapas que antecedem o jogo, abordando dos requisitos básicos aos preparativos finais para a sessão.

Etapa	Descrição
Obtenção	Deve-se ter acesso a regras e cenário de um jogo da modalidade RPG
Participantes	Deve ser formado um grupo de pessoas para jogar
Personagens	Devem ser criados os personagens do jogo
Planejamento	Deve ser definido o plano de fundo e elementos da trama
Preparativos	Os participantes definem como, quando e onde jogar

**Tabela 2.1:** Visão geral sobre as “etapas pré-jogo” de um RPG

A tabela 2.1 cita e resume as etapas que antecedem a partida, as quais serão detalhadas nas sessões seguintes.

## 2.3.1 Obtenção de um RPG

O requisito mais básico para se jogar RPG é escolher e adquirir um jogo da modalidade. Geralmente, um RPG pode ser comprado em livrarias ou sites especializados, contando com materiais obrigatórios ou opcionais, que podem vir na caixa do jogo ou ser obtidos separadamente. Naturalmente, cada jogo possui seu próprio conjunto de peças e materiais. Alguns componentes comuns serão abordados nas subseções seguintes.

### 2.3.1.1 Livro de regras

Descreve o sistema de regras e o cenário onde o jogo se passa, se tratando do componente essencial do jogo. É comum que muitos RPGs sejam comercializados apenas como livros de regras, encontrados tanto em mídia física quanto em mídia digital, sendo este o principal componente de um RPG.



**Figura 2.4:** Livros do RPG “Dungeons and Dragons”

A figura 2.4 (MTELLES, 2019) exibe alguns exemplos de livros de RPG.

### 2.3.1.2 Ficha

Se trata de um formulário com campos a serem preenchidos, como demonstrado na figura 2.5 (BINI, 2020), que exibe um exemplo de ficha de RPG. A finalidade da ficha é armazenar de forma estruturada as anotações dos jogadores a respeito de seus personagens. Dessa forma, é comum que a descrição da aparência do personagem, seus traços de personalidade, valores pré-calculados referentes a atributos e bônus, entre outras informações, sejam anotados na ficha.



Figura 2.5: Ficha do RPG “Cyberpunk”

Geralmente, os livros de regras de jogos de RPG incluem algumas cópias de fichas de jogo vazias, destinadas a serem preenchidas pelos jogadores no momento da criação de personagens. Uma vez preenchida, a ficha se torna uma representação tangível e pessoal do personagem, contendo detalhes que irão moldar as ações e decisões do jogador ao longo do jogo.

### 2.3.1.3 Kit de Dados

No RPG são utilizados dados poliédricos e multifacetados para gerar números aleatórios. Os números gerados nos dados são aplicados em cálculos durante a execução de mecânicas de jogo, determinando os desfechos de certas ações e situações, dessa forma, o uso dos dados torna o RPG mais dinâmico e imprevisível.

O dado de 6 faces é talvez o mais tradicional e conhecido dos dados utilizados em jogos de maneira geral, porém, os RPGs costumam utilizar dados com diferentes números de faces, os quais proporcionam uma ampla gama de possibilidades numéricas, permitindo a aplicação de diferentes regras e mecânicas de jogo.

Entre os dados mais comuns estão os dados de 4, 6, 8, 10, 12 e 20 faces, como consta na figura 2.6 (MAGO, 2020) que demonstra um kit tradicional de dados de RPG. Os dados físicos costumam ser adquiridos no ato da compra do RPG, mas podem ser vendidos separadamente, com diferentes cores, materiais de fabricação, tamanhos, etc.



**Figura 2.6:** Dados de 20, 10, 8, 6 e 12 faces

Também é possível encontrar versões digitais desses dados, disponíveis em aplicativos e sites especializados. Até mesmo o Google oferece dados digitais em sua caixa de pesquisa (basta digitar “d20”), permitindo aos jogadores rolar dados virtualmente.

#### 2.3.1.4 Miniaturas

Alguns jogos utilizam miniaturas, figuras de ação tridimensionais que representam personagens, monstros, construções, etc. Essas peças são posicionadas sobre mapas ilustrados ou painéis quadriculados, e servem como referência visual de posicionamento.



**Figura 2.7:** Miniaturas sobre um painel quadriculado

A figura 2.7(ARENA4G, 2017) demonstra um painel quadriculado e miniaturas, exemplificando sua disposição em uma sessão de RPG.

A presença das miniaturas podem gerar a falsa impressão de que o RPG se trata de um jogo de tabuleiro. Porém, como já evidenciado, tais elementos são apenas recursos auxiliares opcionais. Como pontuado por Mearls (2014, p. 5) “Algumas vezes, um Mestre pode dispor um mapa e usar marcadores ou miniaturas para representar cada criatura envolvida na cena e ajudar os jogadores a acompanhar onde cada um está”.

Em alguns casos podem ser usados mapas, ilustrações, trilhas sonoras, entre outros recursos aplicáveis para aprimorar ou auxiliar algum aspecto da experiência de jogo. Entretanto, o essencial para o jogo ocorrer é, na maioria dos jogos, os livros de regras, as fichas de personagem e os dados multifacetados.

### **2.3.2 Participantes**

O RPG é uma atividade fundamentalmente coletiva, portanto a presença de participantes é um fator crucial. Tradicionalmente, uma pessoa deve assumir a função de “mestre”, enquanto os demais atuam como “jogadores”.

#### **2.3.2.1 Mestre**

Mestre é uma função central na maioria dos jogos da modalidade, sendo originalmente chamado de *Dungeon Master*, mas também de “narrador” ou “mediador”. Durante a preparação do jogo, sua função é planejar a aventura, criando uma história prévia com inúmeros conflitos em aberto (FREITAS FELIPE DELLA CORTE, 2020).

Quando o jogo de fato se inicia, o mestre deve mediar as regras, narrar a aventura e interpretar os personagens que não os dos jogadores. Sobre sua função, é descrito no livro de regras do RPG de *Star Wars*:

Enquanto cada um dos jogadores controla um único herói, o Mestre narra a história, determina as regras, e interpreta todos os personagens do Mestre – o elenco de suporte que serve como aliados e oponentes para os heróis. (SLAVICSEK ANDY COLLINS, 2008, p. 15).

#### **2.3.2.2 Jogador**

Cada um dos demais participantes é um jogador, que interpreta um personagem. “o personagem serve como representante de você no jogo, seu avatar no mundo [...]” (MEARLS, 2014, p.9). Como citado anteriormente, o jogador se projeta na história através do personagem, tomando as decisões como se estivesse dentro da narrativa.

Naturalmente, a escolha ou criação de personagens ocorre antes da partida começar (MEARLS, 2014). Todo RPG tem seu próprio processo de criação de personagens, cabendo ao jogador analisar as opções dadas pelo jogo e selecionar as que melhor componham a persona que ele deseja criar.

### 2.3.3 Criação de Personagens

Os personagens dos jogadores são os protagonistas da história, sendo comum que formem uma equipe para enfrentar os desafios, combinando ideias e capacidades para progredir (MEARLS, 2014). Já os personagens do mestre são os coadjuvantes e antagonistas da trama, com o papel de desafiar, ajudar, direcionar ou atrapalhar os personagens dos jogadores, auxiliando na criação e/ou resolução de conflitos.

Um personagem é um elemento fictício da narrativa, uma pessoa que existe apenas dentro da trama imaginária do jogo, portanto, é natural que ele seja coerente com o cenário, uma vez que faz parte daquele mundo. Além disso, em um jogo de RPG os personagens possuem habilidades e capacidades expressas na forma de valores, que precisam ser compatíveis com as regras do jogo em questão. Por esses motivos, é natural que cada jogo tenha seu próprio processo para criação de personagens.

De maneira geral, para criar um personagem, é preciso defini-lo conceitualmente, imaginando uma persona que se encaixe no cenário proposto (FREITAS FELIPE DELLA CORTE, 2020). Adiante, deve-se buscar o processo de criação de personagens no livro de regras, que dispõe opções, condizentes com o cenário do jogo, para a escolha de raça, habilidades, poderes, equipamentos, etc. Já a aparência do personagem deve corresponder às características da espécie e raça a qual pertence.

É comum que ocorra a distribuição e cálculo de pontos durante o processo de criação de personagens, os quais expressam suas capacidades de forma numérica, algo que será usado em operações mediadas pelas regras durante o jogo. Porém, o processo de criação de personagens aborda, não apenas a distribuição de pontos, mas também a origem dos personagens, suas motivações, traumas, valores, objetivos e outras questões pessoais.

PERSONAGENS SÃO DEFINIDOS POR MUITO mais do que sua raça e classe. Eles são indivíduos com suas próprias histórias, interesses, conexões e capacidades que vão além do que sua raça ou classe podem definir. Esse capítulo apresenta os detalhes que distinguem os personagens uns dos outros, incluindo o básico, como os nomes e a descrição física, as regras de antecedentes e idiomas, a aqueles pontos mais refinados, como os detalhes de sua personalidade e tendência. (MEARLS, 2014, p. 48).

Esses detalhes precisam ser definidos para que mestre e jogadores possam “dar vida” a seus personagens durante a sessão, uma vez que, para fins imersivos, é desejável (e em alguns casos recompensado) que os participantes atuem, representando trejeitos, gesticulação, entonação de voz ou até sotaque do personagem. Também é importante que os jogadores, ao interpretar seus personagens, tomem decisões condizentes com a personalidade dos mesmos.

Porém, é importante pontuar que não é comum ou necessário que os jogadores se levantem da mesa, se fantasiem ou se comportem demasiadamente como atores em jogos de RPG tradicionais. Aspectos mais teatrais são típicos de outras vertentes do RPG, como pontuado por Azevedo (2019, p.29), as quais não são abordadas neste trabalho.

### 2.3.4 Planejamento da Aventura

O planejamento é feito pelo mestre, e consiste em definir elementos com os quais os personagens poderão interagir na história, como vilões, locais, cenas, itens, subtramas, etc (MEARLS, 2014). É crucial que o mestre elabore as problemáticas que vão engajar os jogadores, definindo os objetivos e desafios que os personagens devem superar.

Fica em aberto quais dos elementos planejados serão realmente explorados no jogo, afinal, como já pontuado, os jogadores decidem onde seus personagens vão e o que fazem. Inclusive, eles explorar elementos imprevistos, o que exigirá do mestre uma boa capacidade de improvisação e criatividade.

Naturalmente, todos esses elementos devem ser condizentes com o cenário do jogo, por isso os livros de regras devem dispor as informações necessárias para viabilizar o processo criativo de elaboração do plano de fundo. Para facilitar ainda mais, existem materiais extras, com aventuras pré-planejadas (SLAVICSEK ANDY COLLINS, 2008).

### 2.3.5 Preparativos

Antes do jogo, os envolvidos na atividade devem combinar o local e horário das partidas, debater ideias e expectativas para a experiência, entre outras questões. Tradicionalmente, o jogo ocorre em reuniões presenciais, geralmente em torno de uma mesa, de preferência em um local sem interferências externas.

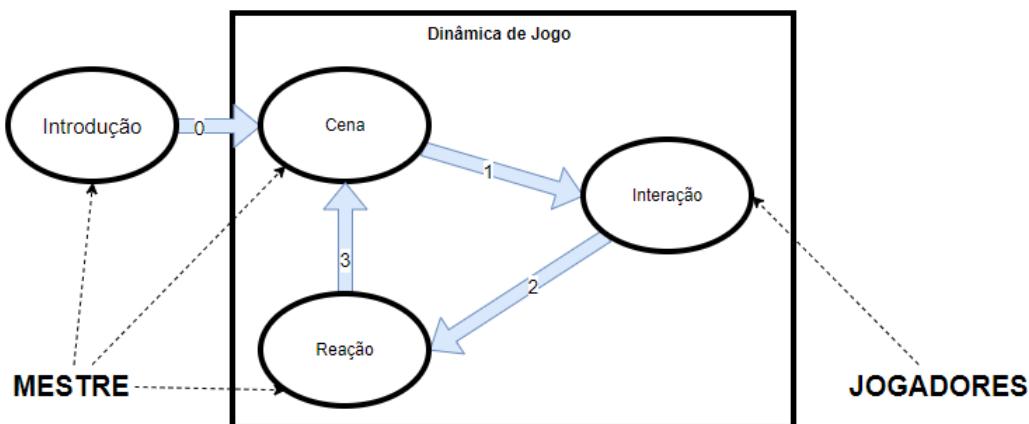


**Figura 2.8:** Sessão remota de RPG transmitida no YouTube

Entretanto, atualmente é comum que alguns ou todos os jogadores se reúnem remotamente, como demonstrado na figura 2.8 (RPGPLANET, 2020), uma vez que a comunicação oral costuma ser o suficiente para que um jogador possa participar.

## 2.4 Etapas do jogo

Uma vez finalizados os preparativos, planejamento da aventura, criação de personagens e demais etapas pré-jogo, os envolvidos podem de fato se reunir para jogar. A figura 2.9 ilustra a dinâmica cíclica do jogo, e foi baseada na sessão “Como Jogar” do livro de regras do RPG “*Dungeons and Dragons*” (MEARLS, 2014, p. 4).



**Figura 2.9:** Fluxo de jogo

As etapas do fluxo são “Introdução”, “Cena”, “Interação” e “Reação”, interligadas na direção e ordem das setas enumeradas (0, 1, 2 e 3). As setas tracejadas indicam quais etapas competem ao mestre e aos jogadores. A lista abaixo resume o fluxo ilustrado, e as sessões seguintes exploram cada uma das etapas detalhadamente.

- **Introdução:** O mestre contextualiza a trama, dispondo informações relevantes para situar os jogadores.
- **Transição 0:** A narrativa é direcionada da introdução para a cena, ajustando o foco narrativo para os personagens.
- **Cena:** O mestre descreve a situação atual dos personagens dos jogadores e abre possibilidades de interação com o ambiente ficcional.
- **Transição 1:** A partir da situação estabelecida, os jogadores avaliam o que seus personagens devem fazer.
- **Interação:** Os jogadores descrevem a ação realizada pelos personagens na cena, interagindo com a narrativa.
- **Transição 2:** O mestre avalia o que os personagens fazem e como isso afeta a narrativa.
- **Reação:** O mestre descreve a reação do cenário, pontuando as consequências das ações do personagem.
- **Transição 3:** A reação do cenário é expressa na narração de uma cena, trazendo ciclo de volta para a primeira etapa.

## 2.4.1 Introdução

A introdução é uma etapa externa ao ciclo, e costuma ocorrer antes da dinâmica de fato se iniciar. Como mencionado anteriormente, o mestre precisa criar um plano de fundo antes do jogo, portanto, é comum que algumas informações precisem ser repassadas aos jogadores antes do jogo começar, algo semelhante ao prólogo de um livro ou filme.

Portanto, a introdução é um momento de contextualização, e pode ser usado para informar eventos recentes, o período em que a história se passa, em que região do mundo fictício ela ocorre, fatores políticos ou sociais relevantes, etc.

Há muito tempo atrás em uma galáxia muito, muito distante... este é o tempo da República, aproximadamente vinte e dois anos antes dos dias do Império e da primeira estação de batalha Estrela da Morte. O Supremo Chancellor Pal-patine lidera a república, Mace Windu e Yoda lideram o conselho Jedi e o começo das Guerras Clônicas está ainda há algumas semanas distantes. Vocês estão em uma cantina de um Porto Espacial em Corellia, um mundo proeminente no núcleo da galáxia. O lugar está bastante movimentado para uma tarde de um dia útil, apesar de não estar no limite da capacidade. Vocês vêem humanos e uma variedade de outras espécies bebendo, rindo e conversando calmamente próximos ao bar, em tendas escuras e mesas livres dispersas. [...] (SLAVICSEK ANDY COLLINS, 2008, p. 15).

Como indicado no trecho citado, em algum momento da introdução, o mestre deve direcionar a narrativa para a cena. Ou seja, ele sai do momento de contextualização geral da trama e se aproxima dos personagens dos jogadores, se concentrando em descrever onde eles estão, o que estão fazendo, o que está acontecendo ao seu entorno, etc. Essa transição é o que demarca a abertura da primeira cena, iniciando o ciclo de cenas que se estende por todo o jogo.

A introdução também pode ser usada para que os jogadores apresentem seus personagens, descrevendo sua aparência, capacidades, objetivos pessoais, etc (SLAVICSEK ANDY COLLINS, 2008, p.15). Isso se justifica pelo fato de que todos os personagens compõem a mesma história e podem interagir entre si, portanto, é interessante que todos os participantes saibam como são os personagens uns dos outros.

Também é comum que o mestre use a etapa de introdução para recapitular eventos de partidas passadas, afinal, os jogadores podem se reunir várias vezes para completar uma história, e dependendo do intervalo desses encontros pode ser necessário situar os participantes em relação ao que ocorreu anteriormente. Isso é especialmente necessário quando, por algum motivo, algum dos jogadores faltou na sessão passada e não sabe o que seus companheiros de equipe fizerem em sua ausência.

## 2.4.2 Cena

O mestre descreve eventos, ambientes, objetos, pessoas, cheiros, sons, sensações e outros componentes narrativos que descrevem a situação atual dos personagens, alimentando a imaginação dos jogadores, os deixando imersos na experiência ficcional.

De repente, vocês todos vêem as portas da cantina correndo pelos trilhos assim que um grupo de brutamontes durões entra na sala. O garçom grita, "Ei! sem blasters aqui!". Ele mergulha atrás do bar, bem a tempo, assim que um dos brutamontes despreocupadamente dispara um tiro de blaster em sua direção. Ali estão quatro capangas – dois humanos, um rodiano e um gamorreano. O gamorreano empunha uma lâmina vibratória e os outros três estão armados com pistolas blasters. O gamorreano rosna alguma coisa rude e gesticula em direção à mesa onde o senador está sentado. [...] (SLAVICSEK ANDY COLLINS, 2008, p. 16).

É preciso que a cena disponha elementos de interação, expondo opções de ação para os personagens (MEARLS, 2014). Cabe os jogadores analisar a situação sob a perspectiva de seus personagens, se baseando em seus objetivos, antecedentes, capacidades, e personalidade para tomar a decisão que melhor se encaixe na situação.

## 2.4.3 Interação

Os jogadores descrevem a ação que seu personagem deve realizar, interagindo com o cenário. Por exemplo, o trecho a seguir demonstra uma possibilidade de interação para a cena citada na sessão 2.4.2.

Eu salto da barraca e me posicione entre o senador e os capangas como minha primeira ação de movimento. Como minha segunda ação de movimento eu saco meu sabre de luz. Como uma ação livre eu ativo o sabre de luz e grito, "Este homem está sob a proteção dos Cavaleiros Jedi!". [...] (SLAVICSEK ANDY COLLINS, 2008, p. 16).

Para outros jogadores, na mesma situação, a interação com a cena poderia ser diferente. Um jogador cujo personagem é um diplomata sem habilidades de combate poderia optar por se esconder ou fugir durante o conflito, já um personagem mercenário, poderia interferir quando algum dos lados já estivesse enfraquecido, garantindo suas chances de vitória para poder cobrar alguma recompensa posteriormente.

É notável que não se pode prever as escolhas dos jogadores, além disso, muitas das decisões tomadas acarretam a aplicação de regras, as quais comumente envolvem rolagens de dados que adicionam ainda mais imprevisibilidade. Afinal, se um personagem decide atacar, ninguém sabe se o mesmo será bem-sucedido ou não, pois são as regras quem definem se o alvo foi atingido.

#### **2.4.4 Reação**

A partir das decisões dos jogadores e das interações realizadas, o mestre define e improvisa as consequências narrativas da situação, se baseando em implicações lógicas, na coerência narrativa e nos elementos planejados para a aventura (MEARLS, 2014).

Os capangas são os próximos a agir. Cada um dos dois humanos atiram no Jedi que está entre eles e o senador. O capanga rodiano atira em Deel e o gamorreano avança rapidamente para retalihar o Jedi com sua lâmina vibratória. [...] (SLAVICSEK ANDY COLLINS, 2008, p. 17).

Descrevendo a reação do cenário, o mestre está narrando uma nova cena, voltando à primeira etapa da dinâmica e fechando o ciclo (MEARLS, 2014). Dessa forma, a dinâmica se repete conforme as cenas são narradas, dando continuidade ao acontecimento anterior. Assim, cena a cena, a trama se constrói, guiada pelo planejamento do mestre, mas sujeita ao improviso, acaso dos dados e a decisões dos jogadores.

#### **2.5 Fim do jogo**

É comum que as sessões de jogo durem algumas horas, o que nem sempre é o suficiente para concluir um enredo. Portanto, os jogadores podem se reunir quantas vezes forem necessárias, realizando várias sessões para concluir uma história. O jogo em si não tem um final definido, portanto, os personagens podem se envolver em novas histórias, formando uma grande “Campanha” (MEARLS, 2014).

Em conferências e eventos de RPG é comum que uma aventura seja concluída em apenas uma sessão, que dura entre 3 e 4 horas (GONZAGA, 2020). Por outro lado, aventuras causais, entre amigos, podem estender uma mesma trama por meses ou anos, intercalando sessões semanais ou mensais. Para reter o interesse dos jogadores por tanto tempo, é comum que vários jogos da modalidade possuam regras para evolução dos personagens, controlando o ganho de experiência e obtenção de novas habilidades e poderes.

#### **2.6 Observação**

É importante pontuar que o trabalho aborda o RPG “tradicional”, fortemente baseado no pioneiro *Dungeons and Dragons*, o qual serve de base para grande parte dos jogos da modalidade, incluindo o jogo específico que será trabalhado na aplicação proposta. Adicionalmente, é válido pontuar que o trabalho reconhece a existência de outras categorias de RPG, os quais subvertem alguns dos conceitos apresentados de forma significativa (AZEVEDO, 2019). Porém, estes jogos não serão abordados, uma vez que fogem ao escopo do trabalho.

### **3 APRESENTAÇÃO DO “MYSTHEN RPG”**

A aplicação sugerida neste trabalho se propõe a adaptar um jogo de RPG em específico, de codinome “Mysthen RPG”. A partir dele são definidas as “regras de negócio” que foram trabalhadas na aplicação, portanto, a finalidade deste capítulo é apresentar as regras e características do jogo citado.

#### **3.1 Visão Geral**

O “Mysthen RPG” é um jogo de autoria própria, desenvolvido e atualizado desde 2015, como um projeto pessoal ainda não publicado, sendo testado e validado apenas informalmente, em sessões recreativas e particulares. Portanto, o jogo não foi criado especificamente para este trabalho.

Se tratando de um RPG de mesa, ele opera aos moldes tradicionais dos jogos da modalidade, portanto, se aplicam ao “Mysthen RPG” todos os conceitos descritos e **referenciados** ao longo do capítulo 2. Dessa forma, o jogo em questão é voltado para a construção coletiva de narrativas em um cenário ficcional e imaginário, naturalmente, submetido a um conjunto de regras.

Para ilustrar o funcionamento prático desse jogo e evidenciar sua correlação com o conteúdo do capítulo 2, foi redigida uma simulação de uma sessão de jogo, a qual consta no apêndice A. Dessa forma, fica demonstrado como se aplicam as regras do “Mysthen RPG”, e como o funcionamento do jogo é condizente com a modalidade RPG.

Uma vez que o software desenvolvido neste trabalho implementa as regras do “Mysthen RPG” em seu funcionamento, uma validação definitiva e formal do jogo pode ocorrer através da aplicação. Entretanto, isso não será abordado neste TCC, uma vez que demandaria que as regras do jogo fossem implementadas em sua totalidade, ultrapassando o escopo do trabalho.

Apenas as regras inclusas no escopo da aplicação serão descritas neste capítulo. Para acessar a versão atual do livro de regras do “Mysthen RPG”, para consulta das regras na íntegra, basta navegar pela plataforma de gerenciamento de conteúdo *Notion*, disponível para consulta através do link: [www.notion.so/rpgleethak/Mysthen-RPG](http://www.notion.so/rpgleethak/Mysthen-RPG).

### 3.1.1 Justificativa

O objetivo posto a este trabalho é o desenvolvimento de uma aplicação que gerencie um RPG de mesa, logo, se faz necessário selecionar algum jogo específico para tal, independente de qual seja. O “Mysthen RPG” atende a essa demanda, uma vez que, como já citado, possui todos os elementos de um jogo dessa modalidade. Adicionalmente, por se tratar de um projeto próprio, evitam-se problemas relacionados a direitos autorais, interpretações equivocadas das regras, etc.

Além disso, a proposta de um RPG de mesa tradicional com elementos analógicos abstraídos para o meio digital, permitindo a interação entre jogadores em diferentes dispositivos, é um modelo híbrido pouco explorado em RPGs tradicionais. Portanto, o projeto é passível de aprimoramento e patenteamento, motivo pelo qual é necessário trabalhar com regras e cenário autorais.

Embora sempre tenha sido jogado aos moldes analógicos tradicionais, a integração com mídia digital é cogitada desde as primeiras versões do “Mysthen RPG”. Por consequência disso, o jogo é propositalmente compatível com o uso de recursos computacionais, dispensando a necessidade de modificação das regras para adaptar o jogo ao aplicativo, o que facilita e agiliza o desenvolvimento.

É importante pontuar que quanto mais numerosas, vareadas, criteriosas e frequentes forem as regras de um RPG, mais o jogo tende a favorecer o realismo, verossimilhança e imersão. Em contrapartida, isso faz com que o jogo se torne mais desafiador, potencializando as problemáticas levantadas no capítulo 1, uma vez que as mecânicas de jogo precisam ser aplicadas e gerenciadas pelos participantes, que possuem limitações humanas referentes a memorização, atenção, cálculo, etc.

Nesse sentido, o “Mysthen RPG” se permite dispor de um corpo relativamente robusto de regras, uma vez que suas mecânicas foram elaboradas cogitando a atuação de uma aplicação que as automatize. Isso nunca impediu que o jogo fosse jogado aos moldes tradicionais, entretanto, uma vez que o jogo é potencialmente mais exigente, a relevância e necessidade de se otimizar esse jogo em particular acaba sendo maior.

Essa robustez das regras visa atender a demandas que compõe a identidade do “Mysthen RPG”, que favorece o realismo e dificuldade recompensante, inspirados na literatura de Bernard Cornwell e nos jogos da franquia *Dark Souls*. Para isso, o jogo conta com mecânicas que abordam a degradação física e mental dos personagens através de redutores, gerencia o acúmulo de adversidades sofridas, reduz a efetividade de armas e magias com base em requisitos não cumpridos, entre outros aspectos desafiadores.

### 3.1.2 Considerações sobre o sistema de regras

O sistema de regras desenvolvido para o jogo “Mysthen RPG” é chamado “SIR”, sigla para “Sistema de Integridade Redutível”. É comum que os RPGs tradicionais sejam baseados no pioneiro *Dungeons and Dragons*, e o sistema “SIR” não é exceção. Os tópicos a seguir comparam ambos os sistemas em diversos aspectos, a fim de demonstrar suas semelhanças e diferenças. Vale pontuar que serão citados alguns termos referentes a regras e mecânicas de jogo, os quais serão explicados nas sessões seguintes.

**Uso de dados poliédricos** Em ambos os jogos o dado de 20 faces é o mais utilizado, se tratando do principal dado do jogo, porém, existem algumas situações onde podem ser aplicados outros tipos de dados, como o de 6 faces, 8 faces, 10 faces, 12 faces, etc.

**Características de personagem** Em ambos os jogos as características de personagens são expressas através de valores, como do deslocamento, da iniciativa, força, conhecimento, etc. Porém, a forma com que esses valores são gerados e calculados é específica de cada jogo, e em muitos casos, o significado e aplicação dos valores também difere.

**Atributos e Perícias** No “D&D” os personagens possuem valores em 6 atributos que representam aspectos fundamentais do personagem, sendo força, destreza, constituição, inteligência, sabedoria e carisma. Eles também possuem perícias, habilidades que especificam o uso dos atributos, como “Acrobacia” que é perícia de destreza, ou “atletismo” que é perícia de força. Tais conceitos não se aplicam ao “Mysthen RPG”.

**Capacidades** No “Mysthen RPG” os valores fundamentais do personagem são suas 15 capacidades, que expressam propriedades físicas e mentais do indivíduo de maneira mais específica que um atributo ou perícia. Por exemplo, dada uma situação em que um personagem busca não ser notado por outro, no “D&D” basta que sejam realizados testes da perícia “furtividade”, mas no “Mysthen RPG” a capacidade usada depende da descrição exata da cena, paços leves e silenciosos dependem da capacidade “sutilidade”, se enfiar rapidamente em um esconderijo depende de “agilidade” e assim por diante.

**Classes** No “D&D” os personagens possuem alguma classe, um arquétipo como guerreiro, mago, clérigo, etc. A classe determina qual o tipo de arma, magia e armadura que o personagem é apto a utilizar. No “Mysthen RPG” não existem classes, ao invés disso os personagens possuem um conjunto de capacidades baseadas em suas qualidades, defeitos, raça, treinamentos, etc. Portanto, não é uma classe quem determina quais armas, magias ou armadura o personagem pode utilizar, e sim, o conjunto de escolhas realizadas na criação do mesmo.

**Progressão de personagem** Ambos os jogos possuem regras para garantir a evolução dos personagens, fazendo com que obtenham novas habilidades e aperfeiçoem suas aptidões. No “D&D” isso ocorre através do acúmulo de pontos de experiência, obtidos ao derrotar inimigos. No “Mysthen RPG” são usados pontos de superação, obtidos por critério narrativo, geralmente quando os personagens expressam aprendizado, seja com seus sucessos ou falhas, geralmente treinando, estudando ou praticando.

**Obtenção de novas habilidades** No “D&D” são determinados 20 níveis de personagens, obtidos pelo acúmulo de pontos de experiência. As classes determinam qual o benefício obtido para cada novo nível alcançado. Já no “Mysthen RPG” o jogador é livre para escolher as habilidades que seu personagem obtém a cada nível, desde que o personagem cumpra os requisitos narrativos, de capacidade e de superação necessários para tal.

**Estrutura de combate em turnos** Em ambos os jogos, os combates são segmentados em turnos, de forma que cada participante tem seu momento para agir durante a batalha, que se estende por vários turnos até que seja concluída. Existe um limite para o tipo e quantidade de ações que um personagem pode realizar a cada turno, mas cada jogo tem sua própria forma de determinar essas limitações.

**Ordem de combate** Em ambos os jogos é preciso definir em que ordem cada personagem atua no combate, sendo realizado um teste de iniciativa, determinando do primeiro ao último a atacar. Porém, no “D&D” essa ordem é fixa, sendo definida apenas uma vez para cada combate, já no “Mysthen RPG” a dinâmica de combate está sempre em movimento, uma vez que esta ordem é redefinida no início de cada turno.

**Uso de Habilidades** No “D&D” é comum que as habilidades especiais dos personagens sejam utilizáveis apenas uma vez por descanso, uma limitação arbitrária que impede o uso de uma mesma habilidade repetidas vezes, o que atrapalharia o balanceamento do jogo. Já no “Mysthen RPG” as habilidades são projetadas para serem utilizadas sob condições específicas que não estão sob total controle dos jogadores, dessa forma, as habilidades especiais operam como técnicas aprendidas que podem ser executadas várias vezes, limitadas apenas pelas condições de uso da habilidade em questão.

**Diferença das armas** Ambos os jogos possuem opções de armas que podem ser utilizadas pelos personagens, elas se diferenciam em peso, letalidade, propriedades e custo, sendo exemplos as espadas, adagas, arcos, machados, alabardas, etc. Porém, enquanto no “D&D” a letalidade das armas é expressa pela quantidade de dano causado por elas, o “Mysthen RPG” trabalha a letalidade através das chances que uma arma tem de ferir um alvo a depender das capacidades do atacante.

**Degradação dos personagens** No “D&D” a degradação dos personagens é expressa pela perda de pontos de vida, que aborda o sofrer de ferimentos e perda de força de vontade, adicionalmente, existem regras específicas para algumas condições debilitantes, como “envenenado” ou “atordoado”, ainda sim, muitas condições são expressas apenas narrativamente pelos jogadores. Já no “Mysthen RPG” o sistema de redutores aborda todos os tipos de degradação física ou mental, como ferimentos, exaustão, questões emocionais, etc. Adicionalmente, os redutores acumulados geram pontos de adversidade, que reduzem a chance de sucesso para as tentativas de ação dos personagens, expressando sua perda de performance causada pela degradação sofrida, algo que o “D&D” não aborda.

**Temática de jogo** Ambos os jogos se passam em cenários de fantasia medieval, onde existem elementos como castelos, reinos, dragões, gigantes, magias, etc. No “D&D” as regras são focadas em combate e aventura, favorecendo narrativas com temáticas épicas e heroicas dentro deste cenário. Já no “Mysthen RPG” as regras possuem um foco semelhante, porém, a possibilidade de criar personagens mais “fracos” e de controlar a mecânica de redutores, permite que o jogo flua entre temáticas. Para jogar em uma narrativa com temática de suspense e horror, por exemplo, basta que os jogadores criem personagens vulneráveis e o mestre seja criterioso com redutores de estresse, causados por medo, ansiedade, etc. A mesma lógica se aplica para narrativas com temática de sobrevivência, favorecidas através da ênfase em redutores de cansaço e necessidades, como fome, sede, calor, frio, etc.

### 3.1.3 Considerações sobre o cenário

Como já mencionado, o cenário é crucial para a definição das regras, porém, esse capítulo visa apresentar apenas os aspectos mecânicos do jogo, não se aprofundando nos detalhes do cenário. Comentando superficialmente, pode-se apontar que o jogo conta com um cenário próprio, inspirado em elementos das obras de J.R.R.Tolkien e HP Lovecraft, além da construção de mundo presente nas franquias *The Elden Scrols* e *Warcraft*.

Tal cenário carrega o nome do mundo em que o jogo se passa, “Leethak”, se tratando de uma fantasia medieval, onde existe magia e criaturas místicas. Esse universo fictício possui seus próprios povos, línguas, culturas, continentes, espécies humanoides, fauna, flora, etc.

Apesar de fantasioso, o cenário busca forte referência nas ciências humanas e da natureza para atingir o máximo de verossimilhança. Portanto, é um mundo de fantasia que se passa em um planeta que surgiu pela ação de fenômenos físicos, não sendo canonicamente criado por entidades, as raças jogáveis possuem características originadas a partir da seleção natural, e mesmo a magia opera por critérios muito bem definidos.

## 3.2 Recorte de regras implementadas

Feita a introdução ao “Mysthen RPG”, suas regras poderiam começar a ser descritas, portanto, no trabalho original as sessões seguintes apresentam as mecânicas de jogo abrangidas na aplicação. Porém, como este documento é uma versão pública, e o conteúdo dessa sessão está sujeito a registro de patente, o recorte das regras e mecânicas do Mysthen RPG não serão exibidas adiante. Por esse motivo, nos capítulos seguintes, ocorrerão diversas citações a subseções do capítulo 3 que só existem no trabalho original.

## 4 FERRAMENTAS E TÉCNICAS

Este capítulo apresenta as ferramentas e técnicas utilizadas no projeto e implementação da aplicação. Foram utilizadas várias ferramentas de desenvolvimento para a realização deste trabalho, todas gratuitas e de domínio do autor. No planejamento foi realizado o levantamento de requisitos, definição do escopo, descrição de casos de uso, e modelagem do banco de dados.

Quanto a implementação, a metodologia “*Kanban*” foi aplicada no gerenciamento os trabalhos. O servidor foi implementado em “*Java*” se comunicando com um banco de dados “*postgreSQL*”. As interfaces foram desenvolvidas utilizando a linguagem “*JavaScript*”, através do *framework* “*VueJS*”.

### 4.1 Requisitos

Requisitos são, segundo o livro Sommerville de engenharia de software (SOMMERVILLE, 2010, p. 57), “as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento”, portanto, o levantamento e análise dos requisitos foram os processos inaugurais do planejamento do software proposto.

Os requisitos podem ser divididos em dois tipos principais: requisitos funcionais e requisitos não funcionais. Os requisitos funcionais referem-se às ações e comportamentos específicos que o sistema deve realizar, descrevendo as funcionalidades e comportamentos do software. Já os requisitos não funcionais dizem respeito a aspectos de qualidade, desempenho, usabilidade, segurança e outras características do sistema que não as funcionalidades em si.

Os capítulos anteriores deste trabalho apresentam as regras de negócio usadas como referência para identificação das necessidades atendidas, dessa forma o material base utilizado para o levantamento dos requisitos foram os fundamentos de um jogo da modalidade RPG (capítulo 2) e as regras do “*Mysthen RPG*” em específico (capítulo 3).

A saída deste processo é a documentação das funcionalidades pensadas para a aplicação, uma listagem de funcionalidades e características do sistema, como apresentado nas subseções seguintes. Dessa forma se atinge o objetivo da etapa de requisitos, como descrito em artigo da USP (DIAS *et al.*, 2010, p. 42) “O principal propósito das atividades de Engenharia de Requisitos é de coletar as necessidades dos usuário e documentá-las de maneira adequada.”

## 4.1.1 Requisitos Funcionais

Esta sessão descreve os atributos de qualidade referentes às funcionalidades necessárias para que o sistema cumpra sua função, pontuando o que o sistema faz.

### 4.1.1.1 Cadastro e Usuários

**RF 01 - Cadastrar usuários** O sistema deve permitir que os usuários sejam cadastrados, informando nome, data de nascimento, e-mail e senha.

**RF 02 - Login de usuário** A aplicação deve permitir que os usuários acessem seus perfis validando e-mail e senha.

### 4.1.1.2 Personagem

**RF 03 - Criação de personagem** O sistema deve permitir a criação de personagens, disponibilizando opções de espécie, qualidade, defeito, treinamentos, conhecimentos e traços de personalidade a serem selecionadas pelo usuário. O processo deve corresponder diretamente as etapas de criação para personagens do jogo “Mysthen RPG” (sessão 3).

**RF 04 - Calcular os valores que compõe o personagem** Os pontos obtidos através das escolhas dos jogadores, durante a criação de personagens, devem ser automaticamente calculados, conforme as operações descritas na etapa de finalização de personagem (sessão 3).

**RF 05 - Gerenciar lista de Personagens** A aplicação deve permitir que o usuário visualize uma listagem dos personagens de sua autoria, o permitindo duplicar ou excluir qualquer personagem selecionado.

**RF 06 - Exibição de características** A aplicação deve permitir que o usuário selecione um de seus personagens criados e visualize seu nome, espécie, valor das capacidades, conhecimentos, aptidões, integridade, base ofensiva, base de poder, iniciativa, guarda, pós-guarda e proteção.

**RF 07 - Bônus calculados** A aplicação deve calcular automaticamente os bônus das capacidades e conhecimentos do personagem para utilização em testes.

**RF 08 - Administrar redutores** A aplicação deve registrar e exibir, automaticamente, os redutores sofridos pelo personagem, para que o usuário saiba quantos pontos de cada categoria estão aflijindo o personagem no momento (sessão 3).

**RF 09 - Administrar adversidade** A aplicação deve somar os redutores e gerar o valor da adversidade do personagem, para que o usuário possa visualizar o acúmulo dos redutores sofridos (sessão 3). Sempre que o valor dos redutores for alterado, a adversidade deve ser atualizada.

**RF 10 - Impacto da adversidade** O valor da adversidade deve impactar as demais características do personagem. Portanto, os bônus das capacidades e ataques, assim como os valores de iniciativa, guarda, pós-guarda e conhecimentos devem ser subtraídos a cada ponto de adversidade acumulado.

#### 4.1.1.3 Sala

**RF 11 - Criar Sala** O sistema deve viabilizar a criação de salas, que devem conter nome e descrição. O sistema deve gerar um código de acesso na criação da sala, o qual pode ser compartilhado pelo criador da mesma, que atua como “mestre” no jogo.

**RF 12 - Inserir personagens de jogadores** Através do código de acesso, outros usuários podem vincular personagens na sala em questão, atuando como “jogadores”. Cada usuário jogador pode vincular apenas um de seus personagens criados a uma sala, também não deve ser possível que o mesmo personagem seja inserido em várias salas.

**RF 13 - Inserir personagens do mestre** A aplicação deve permitir que o criador da sala vincule vários personagens a ela, sejam os criados por ele, ou os pré-concebidos pela aplicação, que se tratam de inimigos comuns e monstros típicos do cenário “Leethak”, que serão disponibilizados para uso.

**RF 14 - Manter log da sala** O sistema deve exibir um histórico atualizado de logs da sala para o criador da mesma, exibindo os resultados de testes, solicitações de acesso, permissões de acesso, entre outras informações pertinentes.

#### 4.1.1.4 Interações

**RF 15 - Realização de testes** A aplicação deve disponibilizar várias opções de testes a serem realizados pelos personagens do “mestre” e dos “jogadores”. Uma vez solicitado um teste, a aplicação deve, automaticamente, gerar um valor aleatório entre 1 e 20 (simulando a rolagem de dados) e realizar os cálculos pertinentes ao teste em questão. Apenas personagens vinculados a salas podem realizar testes.

**RF 16 - Realização de testes de capacidade e conhecimento** A aplicação deve permitir a realização de “testes de capacidade” (sessão 3) e “testes de conhecimento” (sessão 3). Para isso, o sistema deve permitir que o usuário selecione uma das capacidades ou conhecimentos do personagem para teste. O resultado deve ser exibido ao usuário que realizou o teste e ao criador da sala.

**RF 17 - Realização de testes de resistência a redutores** A aplicação deve viabilizar “testes de resistência”(sessão 3). Para isso, o sistema deve permitir que o criador da sala selecione o redutor aplicável, a dificuldade do teste, a capacidade relacionada, e os personagens alvo. O resultado deve ser exibido, e em caso de falha no teste a aplicação deve registrar o redutor pertinente ao personagem afigido.

#### 4.1.2 Requisitos Não Funcionais

Esta sessão descreve os atributos de qualidade não diretamente relacionados às funcionalidades, abordando usabilidade, segurança, entre outras características.

**RNF 01 - Multiplataforma** A aplicação deve ser executada em diversas plataformas, atendendo a usuários em diferentes dispositivos. Naturalmente, as interfaces devem se ajustar aos diferentes tamanhos de tela, para celulares, tablets e monitores.

**RNF 02 - Segurança de senhas** O sistema não deve registrar as senhas dos usuários explicitamente no banco de dados, é necessário codificar as senhas registradas.

**RNF 03 - Autenticação de requisições** As requisições feitas ao servidor devem ser aceitas mediante validação dos dados cadastrais do usuário que tramita pelo sistema.

**RNF 04 - Mapear funções não implementadas** Nas interfaces devem ser deixados espaços e elementos visuais para mapear as funcionalidades que serão implementadas em trabalhos futuros.

**RNF 05 - Interações rápidas** Durante uma partida, os usuários devem conseguir executar qualquer tipo de teste com no máximo 3 cliques. Os “testes” em questão foram apresentados na sessão 3.

**RNF 06 - Navegação otimizada** Navegando pela aplicação o usuário deve ser capaz de, a partir de qualquer tela, exceto a de login e cadastro, acessar sua lista de personagens e salas, ou criar um novo personagem, ou sala.

## 4.2 Escopo

O escopo define os limites e a abrangência do projeto. Segundo o livro “Engenharia de Software” de Sommerville (SOMMERVILLE, 2010, p. 182), o escopo refere-se ao “alcance e propósito do sistema, incluindo quais funcionalidades e características estão dentro e fora do projeto”. Nesse sentido, apenas as funcionalidades essenciais para a demonstração da lógica da aplicação e validação da proposta foram incluídos no escopo.

No livro “Software Engineering: A Practitioner’s Approach” de Roger S. Pressman (PRESSMAN, 2015), se discute a importância do escopo como um conceito central para a engenharia de software, uma vez que definir o escopo permite que os elementos que serão inclusos no projeto sejam claramente distinguidos dos que ficarão de fora. Nesse sentido, esta sessão aborda não apenas os elementos inclusos no escopo, como também comenta e justifica elementos não inclusos.

Na delimitação do escopo são consideradas as restrições de tempo, conhecimento, recursos e tecnologias disponíveis. Nesse sentido, a ponderação de que as regras do “Mysthen RPG” não foram implementadas em sua totalidade, como citado algumas vezes no decorrer do trabalho, se trata de uma clara limitação de escopo, uma vez que tal implementação seria inviável dentro do prazo estabelecido.

Portanto, considerando os requisitos levantados e a proposta do trabalho, foi possível definir quais funcionalidades são realmente viáveis e necessárias para o sistema. A seguir o escopo definido será abordado, separado em assuntos, sendo pontuados os elementos inclusos, seguido de um comentário sobre elementos referentes ao assunto, mas que não foram incluídos.

### 4.2.1 Acesso, login e senha

Analizando as necessidades do sistema em relação ao cadastro e autenticação de usuários, considerando, inclusive, as medidas de segurança e outros requisitos, foram inclusas no escopo da aplicação as seguintes funcionalidades:

- Tela de cadastro de usuário simplificada, apenas nome, login e senha.
- Funcionalidade de acesso via login
- Segurança no registro de credenciais de usuário via UtilPassword.
- Autenticação de rotas via token JWT.

Como mencionado, no escopo também são delimitados os elementos que não serão inclusos. Sobre isso, pode-se apontar a não inclusão de recuperação de senha ou registro de dados adicionais, como data de nascimento ou endereço. Isso porque os cadastros de usuários no sistema são, até o momento, todos de teste, e apenas o básico necessário para criação de contas e login foi considerado.

## 4.2.2 Criação de personagem

Se baseando principalmente nas regras do processo de criação, que consta na sessão 3, e nos requisitos levantados a respeito desse assunto, foram inclusos no escopo do projeto os seguintes elementos:

- Interface intuitiva de seleção das opções de criação de personagem.
- Seleção de 3 opções de espécies, 4 de raças e 8 de etnias (sessão 3).
- Seleção de 12 opções de qualidade e 12 de defeito (sessão 3).
- Seleção de 12 opções de treinamentos (sessão 3).
- Seleção de 20 opções de conhecimentos (sessão 3).
- Exibição dos pontos obtidos a cada etapa, para acompanhamento.
- Cálculo automático dos valores.
- Listagem de personagens criados por usuário.

Não foi inserida no escopo do trabalho a etapa referente a escolha de traços de personalidade do personagem que está sendo criado. Como já mencionado, os traços de personalidade afetam apenas a interpretação do personagem pelo jogador, sem influência nos cálculos e testes, como já explicado no segundo parágrafo do capítulo 3.

## 4.2.3 Criação de sala

As salas de jogo são uma representação virtual de uma mesa ou sessão de jogo tradicional, uma vez que atua como elemento centralizador, viabilizando a interação entre os jogadores. Sobre este tema foram inclusas no escopo:

- Interface para criação de sala.
- Campo para definição do nome e código de acesso da sala.
- Validação da unicidade do código de acesso informado.
- Opção para criar sala ativa ou inativa.
- Inclusa listagem de salas criados por usuário.
- Mapeamento de funcionalidades não inclusas.

Sobre este tema, não foram inclusos mecanismos para descrição de uma sinopse da aventura que se passará na sala, controle de quantos usuários podem acessar a sala, qual a configuração dos itens iniciais dos personagens que entram na sala, definição de uma relação de redutores iniciais para os personagens inclusos na sala, etc.

Apenas os elementos essenciais foram considerados na criação de sala, sendo o nome e código da sala, e o controle do status de ativa ou inativa. Porém, atendendo ao requisito não funcional de número 04, os elementos não implementados, citados no parágrafo anterior, foram mapeados visualmente na interface, como mencionado através do tópico “Mapeamento de funcionalidades não inclusas”.

#### 4.2.4 Painel de jogo - Jogador/Personagem

O painel de jogo é uma das principais interfaces do sistema, usada durante as partidas para realizar testes, verificar informações e aplicar regras e mecânicas necessárias. Referente ao painel do jogador em específico, foram considerados os seguintes elementos:

- Acesso ao painel de jogo com qualquer personagem criado.
- Solicitação de acesso a uma sala de jogo via código de acesso.
- Aba de identificação do personagem (nome, sobrenome, título, etc).
- Valores calculados da Integridade, Aptidão Física e Mental.
- Exibição dos valores de capacidade obtidos na criação do personagem.
- Relação dos redutores, automaticamente atualizados quando alterados.
- Cálculo do acúmulo de adversidade a partir dos redutores.
- Mecanismo reativo de impacto da adversidade sobre outros valores.
- Cálculo do bônus de todas as 15 capacidades e conhecimentos.
- Mecanismo de realização de testes de capacidade.
- Mecanismo de realização de testes de conhecimento.
- Aba de combate, exibindo valores de guarda, pós-guarda e base ofensiva.
- Listagem de armas com valor de ataque calculado.
- Mapeamento de funcionalidades não inclusas

Esse painel aglutina a maior parte das funcionalidades do sistema, naturalmente, como parte das regras não foi implementada, várias funcionalidades do painel não puderam ser inseridas. É o que ocorre, por exemplo, com os testes de combate, não inclusos por depender do sistema de inventário, onde ocorreria a seleção de armas, armadura e itens. O mesmo ocorre com o sistema de magias e habilidades especiais.

A listagem de armas só foi inclusa no escopo porque todo personagem criado possui uma arma padrão, seu corpo, ou seja, a capacidade de combate desarmado (com socos, chutes e afins). Portanto, “desarmado” é considerado uma arma no sistema, sendo a único tipo de ataque listado, e calculado, para todo personagem criado.

Também não foram inclusas informações sobre traços de personalidade, descrição da aparência, entre outros conteúdos informativos da aba de identificação do personagem. Isso porque as etapas onde essas informações seriam geradas não foram implementadas, como a seleção de traços de personalidade na criação de personagem, ou o painel onde o jogador poderia escrever tais descrições.

Por fim é válido ressaltar que o requisito não funcional de número 04 aponta a necessidade de mapear na interface os elementos não implementados, ao que se refere o tópico “Mapeamento de funcionalidades não inclusas”. Portanto, estão demarcados visualmente, mas não implementados, as abas de controle de habilidades, magias, inventário, perfil, além dos testes de conhecimento.

#### 4.2.5 Painel de jogo - Mestre/Sala

O painel de jogo do mestre é semelhante ao painel do jogador, visualmente e em suas funcionalidades. Porém, ele é usado para gerenciar a sala em si, além de todos os personagens do mestre. Referente a este painel, foi incluso no escopo da aplicação:

- Acesso ao painel de jogo de qualquer sala criada.
- Gerenciamento de solicitações de acesso à sala.
- Exibição dos personagens aceitos na sala.
- Exibição do código de acesso à sala para compartilhamento.
- Solicitação de teste de resistência para qualquer personagem.
- Exibição de histórico de logs da sala.
- Histórico de testes e acessos da sala atualizado em tempo real.
- Mapeamento de funcionalidades não inclusas

A inserção de personagens no mestre na sala ficou de fora do escopo, entretanto, as interfaces, tanto do painel de pesquisa e inserção dos personagens quanto da listagem dos personagens já inseridos deve ser implementada visualmente.

Uma vez que o sistema ainda não deve permitir a inserção de personagens do mestre, não é possível selecionar um desses personagens para realizar testes, ataques e afins. Porém, as interfaces referentes a tais funcionalidades também devem ser mapeadas.

### 4.3 Casos de Uso

Casos de uso são representações de unidade funcionais do sistema, como dito por Cockburn no livro “Escrevendo Casos de Usos Eficazes” (COCKBURN, 2005, p. 21) “O caso de uso descreve o comportamento do sistema sob diversas condições conforme o sistema responde a uma requisição de um dos stakeholders, chamado ator primário”.

Portanto, os casos de uso permitem expressar o comportamento da aplicação proposta por meio da descrição textual da funcionalidade, podendo contar com um diagrama que ilustra seu funcionamento. A documentação e desenho dos casos de uso segue as diretrizes definidas pela *Unified Modeling Language* (UML), que estabelece o padrão internacionalmente seguido para esse tipo de documentação.

O propósito é ilustrar o fluxo de ações que produzem os resultados esperados, representando as etapas que compõe as funcionalidades do sistema, viabilizando uma melhor compreensão do mesmo. No caso da aplicação proposta, serão desenvolvidos casos de uso textuais, para auxiliar no desenvolvimento das interfaces e simular funcionalidades antes da implementação real. Permitindo que erros e equívocos sejam identificados com antecedência, evitando o retrabalho.

No contexto dos casos de uso, pelas determinações da UML, são usados “atores” como indicativos de papéis a serem desempenhados no sistema, geralmente um usuário ou outro sistema externo que interage com a aplicação, como mencionado no livro “UML - Guia do Usuário” de Grady Booch et al. (BOOCH, 2006). Nesse sentido, é relevante pontuar que os usuários são os únicos atores do sistema, não existindo outros atores com diferentes níveis de acesso ou mesmo interação com outras aplicações.

Em um RPG os participantes podem assumir o papel de jogador ou de mestre, como já citado na sessão 2.3.2, porém, isso não justifica a criação de dois tipos de usuário na aplicação, uma vez que um usuário pode atuar como mestre ou jogador de uma partida para outra, variando conforme o momento, não existindo limitações de acesso no sistema.

A seguir, serão apresentados os casos de uso desenvolvidos neste trabalho, descrevendo as principais funcionalidades da aplicação e demonstrando as interações entre os atores e o sistema. Cada caso de uso apresenta um detalhamento dos passos necessários para se realizar alguma ação ou cumprir um objetivo específico na aplicação, portanto, expressa concisamente as principais capacidades e recursos disponíveis.

### 4.3.1 Realizar Cadastro de Usuário

- **Autor Principal:** Usuário
- **Objetivo:** Criar uma conta para o usuário acessar o sistema.
- **Pré-condições:** Usuário não registrado no sistema.
- **Fluxo Principal:**
  1. Usuário acessa o aplicativo.
  2. Sistema exibe a tela de login.
  3. Usuário acessa a opção de cadastro.
  4. Sistema exibe o formulário de cadastro.
  5. Usuário preenche os campos do formulário.
  6. Usuário seleciona o botão “Cadastrar”.
  7. Sistema valida as informações cadastrais.
  8. Sistema cria uma conta para o usuário com as informações fornecidas.
  9. Sistema exibe mensagem confirmando cadastro.
  10. Caso de uso encerrado.
- **Fluxos Alternativos:**
  1. **Informações cadastrais inválidas:** Se o sistema, no passo 7 do fluxo, verifica que as credenciais do usuário são inválidas, é exibida uma mensagem correspondente e o fluxo retorna ao passo 2.
- **Pós-condições:** O usuário está registrado no sistema.

### 4.3.2 Realizar Login

- **Autor Principal:** Usuário
- **Objetivo:** Permitir que o usuário acesse o sistema mediante autenticação.

• **Pré-condições:** O usuário deve estar registrado no sistema.

- **Fluxo Principal:**

1. Usuário acessa o aplicativo.
2. Sistema exibe a tela de login.
3. Usuário informa seu login e senha.
4. Usuário seleciona o botão “Login”.
5. Sistema verifica as credenciais.
6. Sistema autentica as credenciais.
7. Sistema gera um token temporário.
8. Sistema salva o token no *sessionStorage* do navegador.
9. Sistema redireciona o usuário para a página inicial do sistema.
10. Caso de uso encerrado.

- **Fluxos Alternativos:**

1. **Credenciais inválidas:** Se o sistema, no passo 5 do fluxo, verifica que as credenciais do usuário são inválidas, seja por não atingir o mínimo de caracteres, conter informações obrigatórias em branco, ou mesmo se o login já estiver sendo utilizado por outro usuário, é exibida uma mensagem correspondente e o fluxo retorna ao passo 2.

2. **Navegação não autorizada:** Acessando uma rota do sistema sem um token válido o fluxo retorna ao passo 2. O token pode se tornar inválido por expirar, ou seja, ter vencido seu tempo de atividade, uma vez que o mesmo é temporário, além disso, pode acontecer do usuário tentar acessar uma rota diretamente pela URL, sem realizar login, geralmente por ter salvo alguma página a qual tente acessar diretamente, sem passar pela tela inicial.

- **Pós-condições:** O usuário autenticado tem acesso ao sistema.

### 4.3.3 Criar Personagem

- **Autor Principal:** Usuário (atuando como jogador)
- **Objetivo:** Criar um personagem.
- **Pré-condições:** O usuário deve estar logado no sistema.
- **Fluxo Principal:**
  1. Usuário acessa opção “Criar Personagem” (tela principal ou menu lateral).
  2. Sistema exibe tela de opções de “espécies, raças e etnias”
  3. Usuário seleciona uma opção.
  4. Sistema exibe tela de opções de “qualidade”.
  5. Usuário seleciona uma opção.
  6. Sistema exibe tela de opções de “defeitos”.
  7. Usuário seleciona uma opção.
  8. Sistema exibe tela de opções de “treinamentos”.
  9. Usuário seleciona 3 opções distintas.
  10. Sistema exibe tela de opções de “conhecimentos”.
  11. Usuário seleciona 2 opções para cada fase da história do personagem (3 fases).
  12. Sistema calcula integridade, aptidão física e mental.
  13. Sistema calcula iniciativa, deslocamento, guarda, base ofensiva e de poder.
  14. Sistema exibe tela de finalização, com os valores calculados.
  15. Usuário digita o nome, sobrenome e título/apelido do personagem.
  16. Usuário confirma criação do personagem.
  17. Sistema valida os dados do personagem criado.
  18. Sistema registra personagem criado
  19. Sistema exibe tela de lista de personagens criados.
  20. Caso de uso encerrado.
- **Fluxos Alternativos:**
  1. **Acompanhamento de pontos obtidos:** A qualquer momento, entre as etapas 2 e 29, o usuário pode selecionar uma opção para exibir a relação de pontos acumulados a cada etapa da criação de personagem, até o momento.
  2. **Validação:** Caso os dados fornecidos sejam inconsistentes de alguma forma, o sistema impede a criação do personagem, e exibe uma mensagem informando o erro.
  3. **Acessar etapa anterior:** A qualquer momento, entre as etapas 2 e 29, o usuário pode retornar para a etapa anterior, apagando as escolhas da etapa atual e posteriores.
- **Pós-condições:** Criado novo personagem.

#### 4.3.4 Criar Sala

- **Autor Principal:** Usuário (atuando como mestre)
- **Objetivo:** Criar nova sala de jogo.
- **Pré-condições:** O usuário deve estar logado no sistema.
- **Fluxo Principal:**
  1. Usuário acessa opção “Criar Sala” (tela principal ou menu lateral).
  2. Sistema exibe tela de criação de sala.
  3. Usuário insere nome da sala.
  4. Usuário insere código da sala.
  5. Usuário define se a sala será criada ativa ou inativa.
  6. Sistema valida o nome da sala.
  7. Sistema verifica se o código criado é único.
  8. Sistema registra a sala criada.
  9. Sistema exibe mensagem confirmando criação da sala
  10. Sistema exibe tela de lista das salas criadas.
  11. Caso de uso encerrado.
- **Fluxos Alternativos:**
  1. **Validação:** Na etapa 6, caso o nome fornecido seja inconsistente de alguma forma (tamanho menor que três caracteres, vazio, nulo, etc.), o sistema impede a criação da sala, e exibe uma mensagem informando o erro.
  2. **Código Incorreto:** Na etapa 7, caso o código informado já esteja em uso, tendo sido associado a outra sala, o sistema impede a criação da sala, e exibe uma mensagem informando o erro.
- **Pós-condições:** Nova sala de jogo criada.

### 4.3.5 Acessar painel de jogo com um personagem

- **Autor Principal:** Usuário (atuando como jogador)
- **Objetivo:** Permitir que o usuário acesse o painel de jogo do personagem.
- **Pré-condições:** O usuário deve ter algum personagem criado.
- **Fluxo Principal:**
  1. Usuário acessa opção “Lista Personagens” (tela principal ou menu lateral).
  2. Sistema busca os personagens criados pelo usuário.
  3. Sistema exibe tela com listagem de todos os personagens encontrados.
  4. Sistema exibe nome, sobrenome e título/apelido de cada personagem.
  5. Sistema exibe integridade, aptidão física e mental de cada personagem
  6. Sistema exibe botão para criação de novo personagem.
  7. Usuário seleciona personagem que deseja acessar.
  8. Sistema carrega as informações do personagem escolhido.
  9. Sistema calcula todos os status do personagem a partir dos dados carregados.
  10. Sistema exibe painel de jogo do personagem selecionado.
  11. Sistema registra novo receptor de mensagens assíncronas do servidor (SSE)
  12. Caso de uso encerrado.
- **Fluxos Alternativos:**
  1. **Nenhum personagem criado:** Na etapa 2 do fluxo, caso o sistema não encontre nenhum personagem criado, associado ao usuário logado, a tela exibirá uma mensagem informativa destacando a ausência de personagens criados.
  2. **Novo personagem:** Assim que a tela de listagem é carregada, é exibido um botão indicando a opção de criar um novo personagem. Portanto, caso o usuário não queira acessar um dos personagens listados, ou mesmo, caso ele não tenha um personagem criado para que seja listado, ele pode acessar a opção que o direciona para a tela de criação de personagens.
  3. **Receptor de mensagens assíncronas:** Na etapa 11 é criada uma conexão SSE entre o cliente e o servidor, através da qual, sempre que o sistema emitir uma solicitação de atualização da sala, os clientes personagens vinculados a essa sala, serão notificados, ao que o sistema reage atualizando os dados da sala e do personagem automaticamente.
  4. **Personagem ainda não vinculado a sala:** Caso o personagem em questão ainda não tenha solicitado e sido aceito em uma sala, a etapa 11 é ignorada. Isso porque não existe vínculo do personagem com uma sala, para que ele seja notificado e atualize os dados da mesma.
- **Pós-condições:** O usuário acessa o painel de jogo com o personagem selecionado.

### 4.3.6 Acessar painel de jogo de uma sala

- **Autor Principal:** Usuário (atuando como mestre)
- **Objetivo:** Permitir que o usuário acesse o painel de jogo do personagem.
- **Pré-condições:** O usuário deve ter alguma sala criada.
- **Fluxo Principal:**
  1. Usuário acessa opção “Lista Salas” (tela principal ou menu lateral).
  2. Sistema busca as salas criadas pelo usuário.
  3. Sistema exibe tela com listagem de todas as salas encontradas.
  4. Sistema exibe nome e código de cada sala.
  5. Sistema exibe botão para criação de nova sala.
  6. Usuário seleciona sala que deseja acessar.
  7. Sistema carrega informações da sala escolhida.
  8. Sistema calcula e carrega as informações da sala escolhida.
  9. Sistema exibe painel de jogo da sala selecionada.
  10. Sistema registra novo receptor de mensagens assíncronas do servidor (SSE)
  11. Caso de uso encerrado.
- **Fluxos Alternativos:**
  1. **Nenhuma sala criada:** Na etapa 2 do fluxo, caso o sistema não encontre ao menos uma sala criada pelo usuário logado, a tela deve exibir apenas uma mensagem indicando a ausência de salas criadas.
  2. **Nova sala:** Assim que a tela de listagem é carregada, é exibido um botão indicando a opção de criar um nova sala. Portanto, caso o usuário não queira acessar uma das salas listados, ou mesmo, caso ele não tenha uma sala criada para que seja listada, ele pode acessar a opção que o direciona para a tela de criação de salas.
  3. **Receptor de mensagens assíncronas:** Na etapa 10 é criada uma conexão SSE entre o cliente e o servidor, através da qual, sempre que o sistema emitir uma solicitação de atualização para determinada sala, o cliente que executa a sala em questão será notificado, ao que o sistema reage atualizando os dados da sala automaticamente.
- **Pós-condições:** O usuário acessa o painel de jogo da sala selecionada.

### 4.3.7 Solicitar acesso à sala de jogo

- **Autor Principal:** Usuário (atuando como jogador)
- **Objetivo:** Solicitar acesso de um personagem a uma sala de jogo.
- **Pré-condições:** O usuário deve acessar o painel de jogo de um personagem, e deve saber o código da sala que deseja acessar.
- **Fluxo Principal:**
  1. Sistema exibe campo para digitação do código da sala.
  2. Usuário insere o código no campo correspondente.
  3. Sistema verifica a existência da sala pelo código.
  4. Sistema registra solicitação, status “solicitado”.
  5. Sistema notifica atualização da sala (via SSE).
  6. Sistema registra solicitação de acesso no log da sala.
  7. Sistema exibe mensagem confirmando solicitação.
  8. Caso de uso encerrado.
- **Fluxos Alternativos:**
  1. **Código Incorreto:** Na etapa 3, caso seja verificada a inexistência da sala, o sistema exibe mensagem notificando o erro por código incorreto.
  2. **Aguardando Confirmação:** Após a etapa 6, a tela com campo para digitação do código da sala é substituída por uma tela de espera, informando que a solicitação foi realizada, e que o sistema está aguardando a confirmação do acesso.
  3. **Sala inativa:** Caso seja verificado que a sala para a qual o acesso foi solicitado existe, mas seu status é “inativo”, a tela de espera informa que a solicitação foi realizada, mas que a sala em questão se encontra inativada no momento.
  4. **Solicitação Aceita:** Caso o usuário criador da sala aceite o acesso, a tela de espera é substituída pela tela da sala, que exibe o nome da sala, a lista de inimigos e a lista de personagens de outros usuários que também possuem acesso à sala.
  5. **Solicitação Negada:** Caso o usuário criador da sala negue o acesso, a tela de espera é substituída pela tela inicial, exibindo campo para digitação de código de uma sala.
- **Pós-condições:** O usuário acessa o painel de jogo com o personagem selecionado.

### 4.3.8 Conceder acesso à sala de jogo

- **Autor Principal:** Usuário (atuando como mestre)
- **Objetivo:** Vincular personagem de outro usuário a uma sala de jogo.
- **Pré-condições:** O usuário deve acessar o painel de jogo da sala, e ter compartilhado o código da sala com outros usuários.
- **Fluxo Principal:**
  1. Usuário abre painel de gerenciamento de acessos da sala.
  2. Sistema exibe lista de personagens aceitos na sala.
  3. Sistema exibe lista de solicitações de acesso.
  4. Usuário concede acesso a um personagem.
  5. Sistema muda o status da solicitação para “Aceito”.
  6. Sistema notifica atualização da sala (via SSE).
  7. Sistema registra concessão de acesso no log da sala.
  8. Sistema exibe mensagem confirmando acesso do usuário.
  9. Sistema remove o personagem da lista de solicitações.
  10. Sistema insere o personagem na lista de personagens aceitos.
  11. Caso de uso encerrado.
- **Fluxos Alternativos:**
  1. **Compartilhamento do código:** O sistema não se encarrega do compartilhamento do código, portanto, o usuário criador da sala (mestre) precisa ter algum canal de comunicação com seus jogadores para lhes enviar o código de acesso. O Código é criado pelo próprio usuário durante a criação da sala, e pode ser consultado na listagem de salas, e no painel de gerenciamento de acessos.
  2. **Negar acesso:** Na etapa 4, caso o usuário criador da sala negue o acesso de um personagem, o sistema altera o status da solicitação para “negado” e o personagem em questão é removido da lista de solicitações.
- **Pós-condições:** O usuário acessa o painel de jogo com o personagem selecionado.

### 4.3.9 Realizar teste de capacidade

- **Autor Principal:** Usuário (atuando como jogador)
- **Objetivo:** Obter o valor de um teste de capacidade.
- **Pré-condições:** O usuário deve acessar o painel de jogo de um de seus personagens, este deve estar vinculado a uma sala.
- **Fluxo Principal:**
  1. Sistema exibe o valor do bônus calculado de todas as 15 capacidades.
  2. Usuário seleciona a capacidade que deseja testar.
  3. Sistema simula rolagem de dados, gerando valor aleatório entre 1 e 20.
  4. Sistema obtém valor da capacidade selecionada para o personagem.
  5. Sistema calcula acúmulo de adversidade do personagem.
  6. Sistema soma rolagem e capacidade, depois subtrai adversidade.
  7. Sistema notifica atualização da sala (via SSE).
  8. Sistema registra resultado do teste no log da sala.
  9. Sistema exibe valores da rolagem, capacidade, adversidade e resultado.
  10. Caso de uso encerrado.
- **Fluxos Alternativos:**
  1. **Acerto Crítico:** Caso o resultado da rolagem de dados seja exatamente igual a 20, o sistema registra rolagem crítica no log da sala, além de destacar esta informação ao exibir os resultados ao jogador realizou o teste.
  2. **Erro Crítico:** Caso o resultado da rolagem de dados seja exatamente igual a 1, o sistema registra rolagem de erro crítico no log da sala, além de destacar esta informação ao exibir os resultados ao jogador realizou o teste.
- **Pós-condições:** O usuário acessa o painel de jogo com o personagem selecionado.

#### 4.3.10 Solicitar teste de resistência

- **Autor Principal:** Usuário (atuando como mestre)
- **Objetivo:** Realizar teste de resistência para personagens selecionados.
- **Pré-condições:** O usuário deve acessar o painel de jogo de uma de suas salas, nele ele deve ter aceitado o acesso de personagens dos jogadores.
- **Fluxo Principal:**
  1. Usuário acessa aba de solicitação de teses.
  2. Sistema exibe painel de solicitação de testes.
  3. Usuário determina valor da dificuldade do teste.
  4. Usuário seleciona o redutor testado.
  5. Usuário seleciona a capacidade utilizada para resistir ao redutor.
  6. Sistema exibe aba para seleção dos personagens alvo do teste.
  7. Usuário seleciona os personagens alvo do teste.
  8. Usuário seleciona opção de solicitar teste.
  9. Sistema obtém relação de personagens alvo.
  10. Sistema realiza, para cada personagens alvo:
    - (a) Simulação de rolagem de dados.
    - (b) Obtenção do valor da capacidade definida para resistir ao redutor.
    - (c) Cálculo do acúmulo de adversidades
    - (d) Soma rolagem e capacidade, subtraindo adversidade.
    - (e) Comparação do resultado com nível de dificuldade estabelecida
    - (f) Registro do resultado do teste no log da sala.
  11. Sistema notifica atualização da sala (via SSE).
  12. Sistema exibe valor e resultado do teste de todos os personagens alvo.
  13. Caso de uso encerrado.
- **Fluxos Alternativos:**
  1. **Acerto Crítico:** Na etapa 10, subitem “e”, caso o resultado seja inferior ao valor da dificuldade estabelecida, é atestada falha no teste, nesse caso é incrementado ‘‘+1’’ no redutor em questão, o que será acumulado na adversidade do personagem que falhou no teste.
  2. **Acerto Crítico:** Caso o resultado da rolagem de dados seja exatamente igual a 20, o sistema registra rolagem crítica no log da sala, além de destacar esta informação ao exibir os resultados ao jogador realizou o teste.
  3. **Erro Crítico:** Caso o resultado da rolagem de dados seja exatamente igual a 1, o sistema registra rolagem de erro crítico no log da sala, além de destacar esta informação ao exibir os resultados ao jogador realizou o teste.
- **Pós-condições:** O usuário acessa o painel de jogo com o personagem selecionado.

## 4.4 Metodologia Ágil

A metodologia de trabalho foi definida com base nas seguintes características do projeto: o tamanho reduzido da equipe (composta por apenas uma pessoa), a necessidade de fluidez e dinamismo do processo de desenvolvimento para cumprimento do prazo, e a possibilidade de redução ou expansão do escopo.

Portanto, a implementação do sistema será gerenciada através de uma metodologia ágil, como é recomendado para projetos que possuam estas, entre outras, características (SOARES, 2004), sendo selecionado o “Kanban”, que opera na administração de demandas, para definição e administração dos objetivos de implementação.

Desenvolvido em linhas de montagem japonesas, o Kanban foi capaz otimizar a produção através do controle de tarefas utilizando recursos visuais (PEINADO; AGUIAR, 2007). As etapas do desenvolvimento são dispostas na forma de cartões, movidos entre colunas customizáveis, que indicam estados como “pronto” ou “em desenvolvimento”.

Neste trabalho, esses cartões foram gerenciados virtualmente, através da plataforma “Jira”, voltada ao desenvolvimento de software. Essa plataforma foi selecionada por ser integrada ao “BitBucket”, a aplicação que opera através do sistema “Git” como repositório de código, e que será usada para o controle de versões.

NÃO INICIADO 1 ITEM	INICIADO 2 ITENS	EM ESPERA 2 ITENS	CONCLUÍDO 4 ITENS
Ajustes Finais da criação de personagem <input checked="" type="checkbox"/> TCC-29	Definir rotas de erro 500 e 404 <input checked="" type="checkbox"/> TCC-12	Módulo de acesso do Usuário <input checked="" type="checkbox"/> TCC-10	Tela do mestre <input checked="" type="checkbox"/> TCC-34
+ Criar item	Erro: Ao recarregar dashboard sempre encaminha usuário para tela de login <input checked="" type="checkbox"/> TCC-14	Implementar tabelas da Game Data <input checked="" type="checkbox"/> TCC-19	implementar comunicação server -> client <input checked="" type="checkbox"/> TCC-35
			Definir Módulos do Sistema <input checked="" type="checkbox"/> TCC-2
			Cronograma <input checked="" type="checkbox"/> TCC-31

**Figura 4.1:** *Quadro Kaban da aplicação no Jira Software*

A figura 4.1 demonstra o painel de quadros da plataforma “Jira”, é possível observar as 4 colunas de controle utilizadas, intituladas como “Não Iniciado”, “Iniciado”, “Em espera” e “Concluído”. Tais colunas correspondem ao estado de resolução das tarefas, expressas na forma de cartões que são movidos entre as colunas, como também é visível na referida figura.

É válido comentar que a plataforma utilizada para construção dos quadros, o “Jira”, oculta os quadros mais antigos da coluna de tarefas concluídas, sendo necessário acessar a opção “Ver todos os itens concluídos”, para ter acesso à lista completa. A figura 4.2 destaca, na legenda **A**, o painel de exibição de itens concluídos.

The screenshot shows the Jira interface with the following elements:

- Top Bar:** Includes buttons for 'Compartilhar', 'Exportar itens', 'Ir para pesquisa avançada', 'VISUALIZAÇÃO DE LISTA' (selected), 'VISUALIZAÇÃO DETALHADA', 'Mais +', 'Redefinir', 'Salvar filtro', 'BÁSICA', and 'JQL'.
- Left Panel (Kanban View):** Shows a list of items categorized by status: 'Resolvido'. Item 'TCC-33' is highlighted with a red arrow labeled **A**. Other items include 'Gameplay - Design da Tela', 'Seleção de personagem', 'Personagens - Concluir criação' (highlighted with a red arrow labeled **C**), 'Personagens - Conhecimentos', 'Cria Personagem - Abas utilitárias', and 'Homepage,'.
- Middle Panel (Card View):** Displays the details for the selected item 'Personagens - Concluir criação'. It includes sections for 'Descrição' (with bullet points: 'Exibir status finais do personagem' and 'Salvar personagem'), 'Atividade' (with tabs: 'Tudo' (selected), 'Comentários', 'Histórico'), and a 'Informações' sidebar. The sidebar shows fields like 'Responsável' (Não definido), 'Categorias' (Nenhum), 'Versões corrigidas' (Nenhum), 'Desenvolvimento' (15 commits, Há 1 mês), and 'Relator' (Chris). A red arrow labeled **B** points to the status bar at the top right showing '10 de 30'. Another red arrow labeled **D** points to the 'Informações' sidebar.
- Bottom Panel:** Shows creation and update details: 'Criado 1 de maio de 2023 às 20:35', 'Atualizado 9 de maio de 2023 às 17:55', and 'Resolvido 9 de maio de 2023 às 17:55'.

**Figura 4.2: Quadros concluídos**

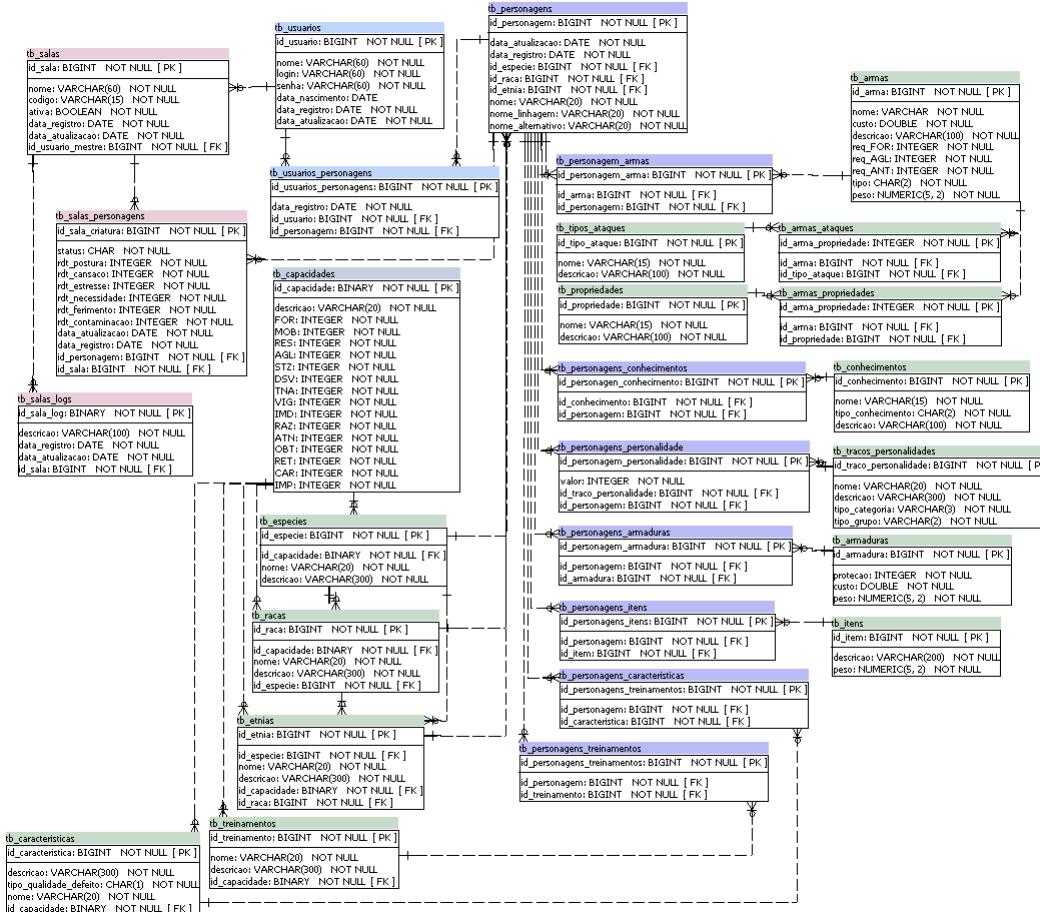
A figura 4.2 também indica, na legenda **B**, o número de quadros concluídos, sendo 30 ao final da implementação. Já a legenda **C** indica a possibilidade de inserir, para cada quadro, um título e descrição do trabalho que precisa ser realizado, o que auxilia no gerenciamento das atividades referentes a cada quadro. Nesse mesmo espaço, é possível inserir comentários, imagens, tabelas, anexos, entre outros recursos auxiliares.

Ainda na figura 4.2, a legenda **D** indica algumas informações sobre o cartão, incluindo a mencionada relação entre o “Jira” e o repositório de código “BitBucket”. Após a implementação de um novo trecho de código, a alteração é enviada ao repositório (processo chamado “commit”), nesse momento, o quadro referente a tarefa que está sendo cumprida pode ser referenciado, vinculando o “commit” ao cartão do “Kanban”.

Levando em consideração a praticidade e versatilidade da metodologia “Kanban” em conjunto com a facilidade de uso da plataforma “Jira” e sua integração com o “BitBucket”, é possível concluir que a escolha dessa abordagem metodológica se mostrou altamente satisfatória, uma vez que não apenas atenderam às necessidades de gerenciamento do projeto, mas também foram fáceis e ágeis de utilizar.

## 4.5 Banco de Dados

Antes de implementar o banco de dados em si, foi necessário projeta-lo, através da construção do diagrama de entidade-relacionamento (DER). De acordo com Elmasri e Navathe, no livro “Fundamentals of Database Systems”(ELMASRI; NAVATHE, 2016), gerar esse diagrama permite representar atributos, entidades e relacionamentos entre entidades, fornecendo uma visão clara e estruturada do modelo.

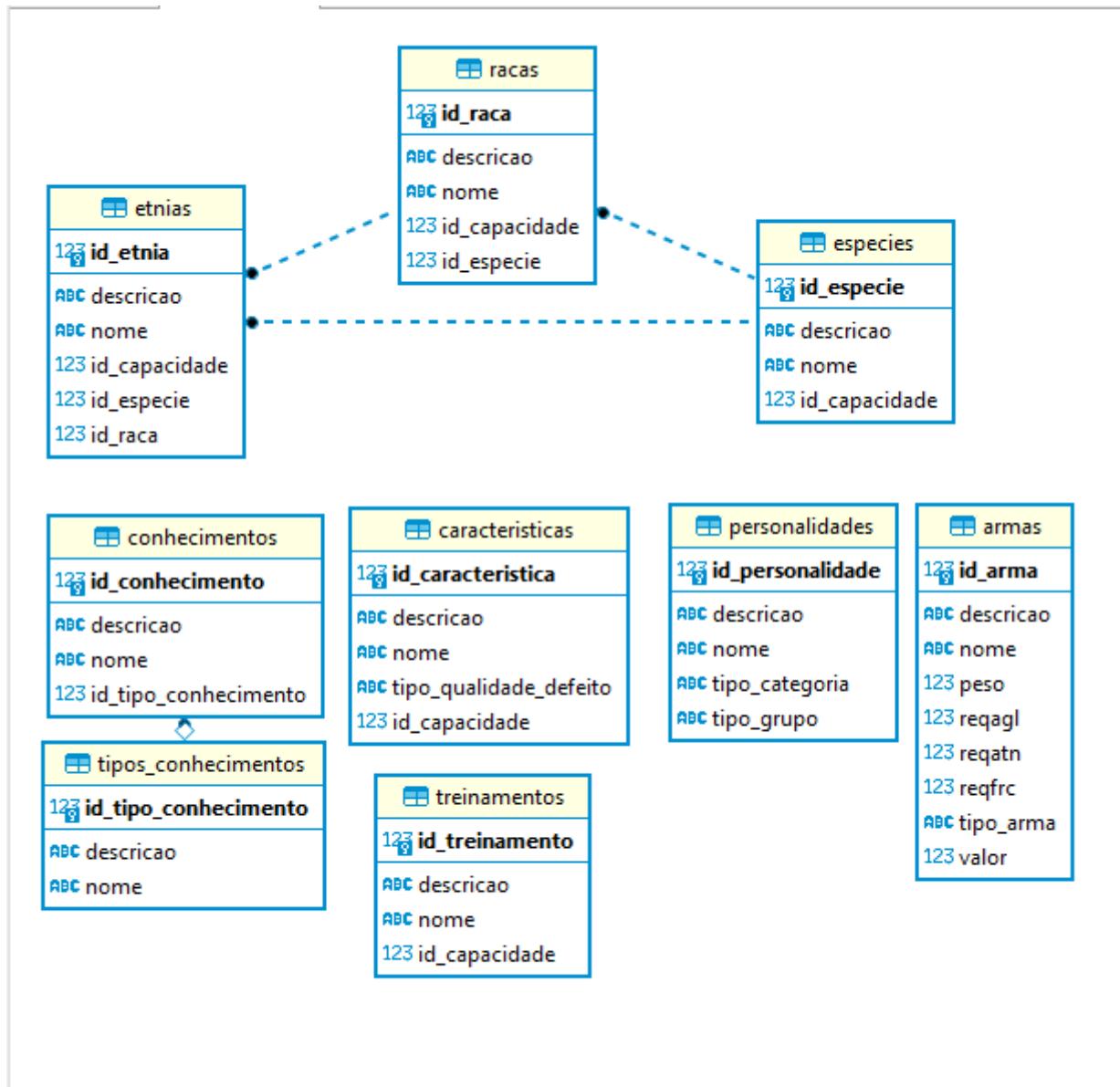


**Figura 4.3:** Diagrama Entidade-Relacionamento da aplicação

Nesse trabalho, a plataforma *SQL Power Architect* foi utilizada para “desenhar” o modelo de bando de dados, como consta na figura 4.3 (em tela cheia no apêndice B). O diagrama foi subdividido em quatro esquemas, visando facilitar a organização e compreensão do modelo. Devido ao tamanho do diagrama em relação à página, a legibilidade do artefato fica comprometida. Portanto, os esquemas serão abordados individualmente nas sessões seguintes.

### 4.5.1 Esquema Game Data

O esquema “Game Data” é o que contém mais tabelas no sistema, as quais foram indicadas em cor verde no diagrama 4.3. Ele agrupa as tabelas de dados referentes a informações do próprio jogo, isto é, as opções de escolha que o sistema disponibiliza aos usuários, como as opções de armas, magias, raças, conhecimentos, etc.

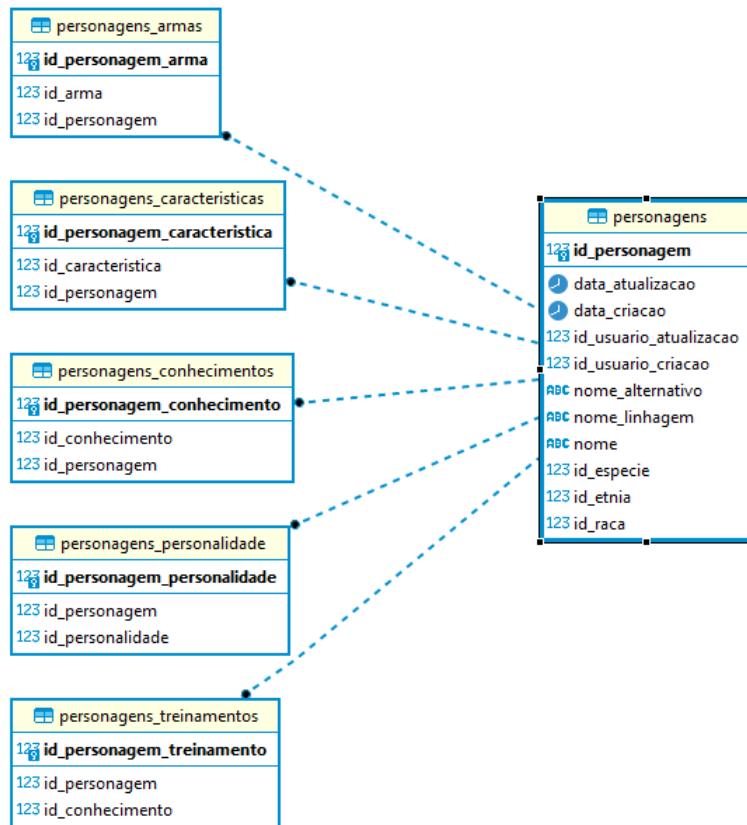


**Figura 4.4:** Tabelas implementadas do esquema “Game Data”

O usuário jamais insere, altera ou remove itens nessas tabelas, ele apenas os visualiza e seleciona, sendo a combinação desses dados que gera os personagens, e possibilita a obtenção de itens. Também é válido comentar que esse é o esquema que mais deve receber tabelas novas a medida que novos módulos do sistema são implementados.

## 4.5.2 Esquema Personagem

O esquema “Personagem” tem suas tabelas indicadas em roxo no diagrama 4.3. São tabelas referentes aos dados de personagens criados no sistema, armazenando todas as informações cruciais a respeito de personagens criados, como nome ou identificação, conforme indicado na figura 4.5.



**Figura 4.5:** Tabelas implementadas do esquema “Personagem”

Nesse esquema também existem tabelas que vinculam um personagem aos dados das tabelas do esquema “Game Data”. Por exemplo, para vincular o personagem a um treinamento, considerando que ele sempre terá mais de um treinamento, o esquema personagem conta com a tabela “*personagens\_treinamentos*”.

Os dados das tabelas desse esquema são recorrentemente alterados pelos usuários, uma vez que os mesmos podem criar e editar vários personagens. Adicionalmente, os personagens-base oferecidos pelo próprio jogo para uso durante as aventuras, como monstros, feras e outros tipos de inimigos, também teriam seus dados registrados (via script) nessas tabelas.

### 4.5.3 Esquema Capacidade

O esquema capacidade possui uma única tabela, indicada em cor cinza no diagrama 4.3, e representa um elemento central tanto para as tabelas do esquema de personagens, quanto do esquema “Game Data”.

	capacidades_game_data
id_capacidade	
123	agilidade
123	atencao
123	carisma
123	desenvoltura
123	forca
123	imunidade
123	imposicao
123	mobilidade
123	obstinacao
123	razao
123	resistencia
123	retorica
123	sutileza
123	tenacidade
123	vigor
ABC	descricao

**Figura 4.6:** Tabelas implementadas do esquema “Capacidades”

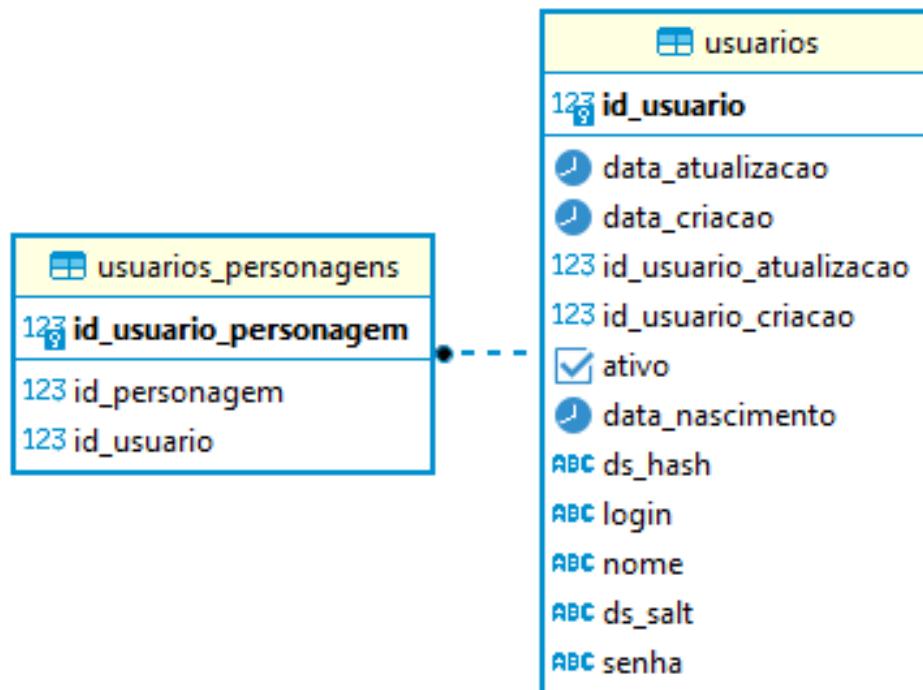
Isso porque, é na tabela de capacidades que são registrados os valores de cada uma das 15 capacidades (como demonstra a figura 4.6, que exibe seus campos), uma vez que, é a partir desses valores que são calculados a integridade, aptidão física, iniciativa, guarda, etc.

Esse esquema é crucial para o funcionamento da aplicação, uma vez que várias tabelas de outros esquemas possuem uma relação de valores de capacidades. Por exemplo, todas as raças do jogo possuem uma relação de pontos distribuídos entre as 15 capacidades existentes, por isso, ao selecionar a raça, o personagem criado também obtém os pontos de capacidade relacionados, e o mesmo ocorre com treinamentos, características (qualidades e defeitos), etc.

É estudada a viabilidade de criação de duas tabelas de capacidade, uma dedicada apenas aos valores de capacidades de registros do próprio sistema, ou seja, vinculados a itens das tabelas do esquema “Game Data”, e a outra voltada apenas para as capacidades de personagens criados por usuários.

#### 4.5.4 Esquema Cadastro

O esquema de cadastro, indicado em azul no diagrama 4.3, possui as tabelas que armazenam os dados dos usuários, mantendo informações cadastrais e de login. Nesse esquema também consta a tabela que armazena a relação entre os usuários e os personagens criados por eles.



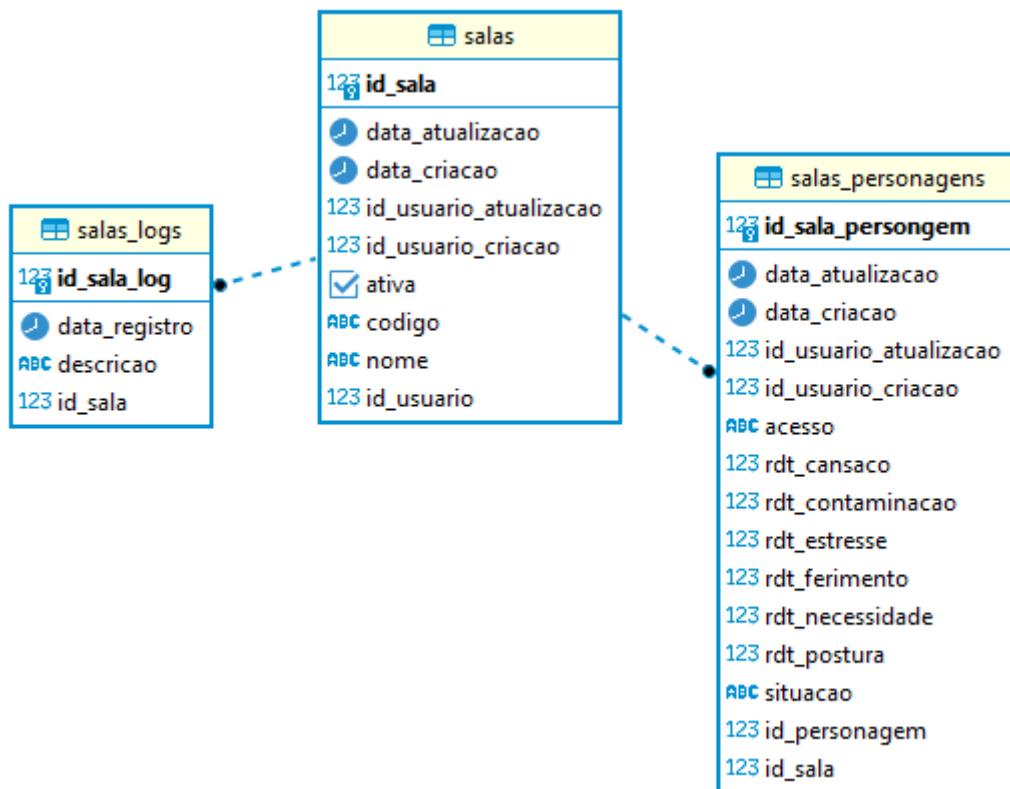
**Figura 4.7:** Tabelas implementadas do esquema “Capacidades”

Observando a figura 4.7 é possível notar os campos “senha”, “hash” e “salt”, usados na codificação de senhas, conforme mencionado na sessão 4.6, sobre a classe UtilPassowrd. Basicamente, o uso de salt e hash em conjunto para codificar senhas é uma medida básica de segurança, implementada apenas para adicionar alguma camada de ocultação dos dados.

Também é possível observar a inexistência de um campo que diferencie tipos de usuário, entre mestre e jogador. Como já mencionado, um usuário pode desempenhar tanto o papel de mestre quanto de jogador em diferentes partidas, logo, na primeira versão do sistema, não há necessidade de categorização dos tipos de usuário.

#### 4.5.5 Esquema Sala

O esquema sala, indicado pela cor rosa do diagrama 4.3, contém as tabelas referentes as salas de jogo, armazenando as informações necessárias para criação e acesso às salas. Eventualmente os usuários podem criar novas salas, portanto, esta é uma tabela cujos dados são eventualmente manipulados por eles, como ocorre no caso das tabelas de personagem.



**Figura 4.8:** Tabelas implementadas do esquema “Sala”

A tabela “salas\_personagens” mantém o vínculo entre um personagem e a sala a qual foi inserido. Quando o usuário solicita a entrada em uma sala, é gerado um registro com situação “SO” (solicitado), que posteriormente será alterado para “AC”(aceito) ou “NG”(negado), indicando assim qual a situação do vínculo entre sala e personagem.

Nessa mesma tabela, é realizado o controle dos redutores do personagem, cujo somatório resulta no “acúmulo de adversidade”. Isso permite que cada instância de um personagem presente em uma sala tenha seus redutores controlados individualmente.

Por exemplo, nas tabelas de personagem consta o registro de um “Goblin”, uma criatura do próprio jogo. O mestre pode inserir 3 Goblins em uma sala como inimigos, sendo cada um deles um registro na tabela “salas\_personagens”. Dessa forma, em meio ao combate, um dos Goblins pode ser atacado, e apenas sua instância de sofrerá o redutor de ferimento.

## 4.5.6 Esquema Auditoria

O esquema auditoria não está indicado no diagrama de entidade-relacionamento, uma vez que nenhuma de suas tabelas é relacionada a funcionalidades do sistema.

<b>usuarios_personagens_auditoria</b>	<b>personagens_auditoria</b>	<b>usuarios_auditoria</b>	<b>salas_personagens_auditoria</b>
123 <b>id_usuario_personagem</b> 123 <b>rev</b>	123 <b>id_personagem</b> 123 <b>rev</b>	123 <b>id_usuario</b> 123 <b>rev</b>	123 <b>id_sala_persongem</b> 123 <b>rev</b>
123 <b>revtype</b> ⌚ <b>data_atualizacao</b> ⌚ <b>data_criacao</b> 123 <b>id_usuario_atualizacao</b> 123 <b>id_usuario_criacao</b> 123 <b>id_personagem</b> 123 <b>id_usuario</b>	123 <b>revtype</b> ⌚ <b>data_atualizacao</b> ⌚ <b>data_criacao</b> 123 <b>id_usuario_atualizacao</b> 123 <b>id_usuario_criacao</b> abc <b>nome_alternativo</b> abc <b>nome_linhagem</b> abc <b>nome</b> 123 <b>id_capacidade</b> 123 <b>id especie</b> 123 <b>id_etnia</b> 123 <b>id_raca</b>	123 <b>revtype</b> ⌚ <b>data_atualizacao</b> ⌚ <b>data_criacao</b> 123 <b>id_usuario_atualizacao</b> 123 <b>id_usuario_criacao</b> <input checked="" type="checkbox"/> <b>ativo</b> ⌚ <b>data_nascimento</b> abc <b>ds_hash</b> abc <b>login</b> abc <b>nome</b> abc <b>ds_salt</b> abc <b>senha</b>	123 <b>revtype</b> ⌚ <b>data_atualizacao</b> ⌚ <b>data_criacao</b> 123 <b>id_usuario_atualizacao</b> 123 <b>id_usuario_criacao</b> 123 <b>rdt_cansaco</b> 123 <b>rdt_contaminacao</b> 123 <b>rdt_estresse</b> 123 <b>rdt_ferimento</b> 123 <b>rdt_necessidade</b> 123 <b>rdt_postura</b> abc <b>situacao</b> 123 <b>id_sala</b> abc <b>acesso</b>
<b>salas_auditoria</b>	<b>capacidades_auditoria</b>	<b>capacidades_personagens_auditoria</b>	
123 <b>id_sala</b> 123 <b>rev</b>	123 <b>id_capacidade</b> 123 <b>rev</b>	123 <b>id_capacidade</b> 123 <b>rev</b>	
123 <b>revtype</b> ⌚ <b>data_atualizacao</b> ⌚ <b>data_criacao</b> 123 <b>id_usuario_atualizacao</b> 123 <b>id_usuario_criacao</b> <input checked="" type="checkbox"/> <b>ativa</b> abc <b>codigo</b> abc <b>nome</b> 123 <b>id_usuario</b> 123 <b>rdt_cansaco</b> 123 <b>rdt_contaminacao</b> 123 <b>rdt_estresse</b> 123 <b>rdt_ferimento</b> 123 <b>rdt_necessidade</b> 123 <b>rdt_postura</b> abc <b>situacao</b> 123 <b>id_personagem</b>	123 <b>revtype</b> ⌚ <b>data_atualizacao</b> ⌚ <b>data_criacao</b> 123 <b>id_usuario_atualizacao</b> 123 <b>id_usuario_criacao</b> 123 <b>agilidade</b> 123 <b>atencao</b> 123 <b>carisma</b> 123 <b>desenvoltura</b> 123 <b>forca</b> 123 <b>imunidade</b> 123 <b>imposicao</b> 123 <b>mobilidade</b> 123 <b>obstinacao</b> 123 <b>razao</b> 123 <b>resistencia</b> 123 <b>retorica</b> 123 <b>sutileza</b> 123 <b>tenacidade</b> 123 <b>vigor</b> abc <b>descricao</b>	123 <b>revtype</b> ⌚ <b>data_atualizacao</b> ⌚ <b>data_criacao</b> 123 <b>id_usuario_atualizacao</b> 123 <b>id_usuario_criacao</b> 123 <b>agilidade</b> 123 <b>atencao</b> 123 <b>carisma</b> 123 <b>desenvoltura</b> 123 <b>forca</b> 123 <b>imunidade</b> 123 <b>imposicao</b> 123 <b>mobilidade</b> 123 <b>obstinacao</b> 123 <b>razao</b> 123 <b>resistencia</b> 123 <b>retorica</b> 123 <b>sutileza</b> 123 <b>tenacidade</b> 123 <b>vigor</b> abc <b>descricao</b>	

**Figura 4.9:** Tabelas implementadas do esquema “Auditoria”

As tabelas desse esquema operam apenas como um registro do histórico de operações de algumas tabelas específicas, permitindo que, caso necessário, sejam consultadas alterações e remoções de registros nas mesmas, auxiliando na resolução de possíveis problemas referentes a inconsistências de dados.

## 4.5.7 Considerações

Relembrando que o diagrama está disponível em tela cheia no apêndice B deste trabalho, o mesmo foi desenvolvido antes da implementação, portanto, constam algumas tabelas que não foram de fato implementadas. Já as imagens de cada esquema foram coletadas após o desenvolvimento, ilustrando as tabelas efetivamente implementadas.

## 4.6 Implementação

Esta sessão descreve as principais ferramentas utilizadas na implementação da aplicação, incluindo linguagens de programação, *frameworks* entre outras tecnologias combinadas no processo de desenvolvimento. A arquitetura do sistema é a tradicional “cliente/servidor”, utilizada para aplicações web de maneira geral.

Nessa estrutura, os “clientes” são os dispositivos de usuários, executando a interface de comunicação com o “servidor” central, que executa os serviços disponibilizados pelo sistema. Portanto, existem duas camadas distintas, “Back-End”, onde foi implementado o banco de dados e o servidor em si, e o “Front-End” onde foram implementadas as interfaces executadas nos computadores clientes, cada camada e suas respectivas ferramentas serão abordadas nas subseções seguintes.

### 4.6.1 Back-End

É considerado “back end” a parcela do sistema responsável pelo processamento e armazenamento de dados, bem como pela lógica de negócios da aplicação. Segundo Tanenbaum e Van Steen, no livro “Distributed Systems: Principles and Paradigms” (TANENBAUM; STEEN, 2009), esta camada pode incluir diferentes componentes, como servidores, bancos de dados, Application Programming Interfaces (APIs) e serviços web, além de tratar da segurança, escalabilidade e desempenho do sistema.

#### 4.6.1.1 Banco de dados

Por Elmasri e Navathe, no livro “Fundamentals of Database Systems” (2016) um banco de dados é uma “coleção de dados” inter-relacionados, podendo ser textos, valores numéricos, imagens, etc. Naturalmente, a aplicação implementada conta com um banco de dados para armazenar informações, como dados cadastrais de usuários, personagens criados, conteúdos referentes ao jogo, etc.

O Sistema de Gerenciamento de Banco de Dados (SGBD) relacional de código aberto *PostgreSQL*, foi utilizado para implementação do banco de dados. Ele foi manipulado com o auxílio da plataforma *DBeaver*, um gerenciador de banco de dados gratuito, também de código aberto, que oferece uma interface intuitiva e abrangente para trabalhar com diversos SGBDs.

A estrutura do banco de dados implementado foi amplamente abordada na sessão 4.5, principalmente no decorrer das subseções que abrangem os esquemas que subdividem o banco. Ressalta-se que o banco de dados implementado atende as necessidades atuais do sistema, mas existe a possibilidade de modificação e acréscimos no modelo de dados em versões futuras conforme necessário.

#### 4.6.1.2 Servidor

O servidor é quem fornece os serviços e recursos da aplicação, atendendo as solicitações dos “dispositivos clientes”, conforme descrito por Pressman (2015). Neste trabalho os serviços são o registro de personagens, o cadastro de um usuário, o fornecimento de uma lista de defeitos selecionáveis, etc.

O servidor foi implementado através da linguagem “*Java*”, na versão 13, utilizando o ambiente “*Eclipse*”. Essa é uma das formas de se trabalhar com o paradigma de Programação Orientada a Objetos (POO), que permite organizar e estruturar o código de maneira eficiente, facilitando seu desenvolvimento, manutenção e reutilização.

No livro “Engenharia de Software” (SOMMERVILLE, 2010) é descrito que a POO trabalha com classes para abstrair entidades importantes, no contexto deste trabalho, essas entidades são usuários, salas, treinamentos, raças, espécies, etc. As classes são estruturas constituídas de métodos e atributos a serem preenchidos, os quais descrevem, respectivamente, comportamentos e características.

Por exemplo, uma das classes implementadas é a chamada “Personagem”, uma entidade que, como o nome sugere, representa um personagem criado por um jogador. Os atributos da classe são as características que todo personagem possui, como seu nome ou espécie. Já os métodos são os comportamentos da classe, através dos quais é possível, por exemplo, inserir uma espécie para o personagem ou obter o nome do mesmo.

Com base em uma classe é possível criar vários objetos, que são instâncias individuais da classe, com atributos preenchidos. Dessa forma, a classe “Personagem” serve como um molde para criar vários personagens individuais, cada um contendo seus próprios valores de atributo, e podendo executar os métodos definidos na classe.

As classes implementadas correspondem diretamente às tabelas exibidas na sessão 4.5, portanto, para saber quais as classes instanciadas no sistema basta observar as tabelas descritas na modelagem do banco de dados. Inclusive, os campos das tabelas correspondem aos atributos de suas respectivas classes.

Essa correspondência se deve ao mapeamento objeto-relacional utilizado nas entidades, feito através do framework *Hibernate*, que implementa o padrão *Java Persistence API* (JPA). Essa ferramenta promove uma integração entre o banco e o servidor, utilizando anotações para mapear as classes, para representarem as tabelas, inclusive especificando relacionamentos e propriedades dos campos das tabelas diretamente nas classes Java.

Dessa forma, é possível, a partir do servidor, acessar a tabela de “Raças” no banco de dados, obter a relação de todas as raças disponíveis no jogo, e gerar, automaticamente, uma lista de objetos da classe “Raça”. Outro exemplo, seria o jogador gerar um objeto “Usuário” na criação de sua conta, e os dados dessa instância serem inseridos como um novo registro na tabela correspondente.

#### 4.6.1.3 Dependências

As dependências são bibliotecas ou módulos externos, inseridos no projeto para estender as funcionalidades do mesmo. Um exemplo de dependência externa utilizada é o *Hibernate*, que foi discutido na subseção anterior. Porém, este não foi o único *framework* externo utilizado na implementação do servidor.

Para gerenciar e padronizar as dependências inseridas, foi utilizada a ferramenta *Maven* (versão 4.0.0), que permite a fácil inclusão e remoção de dependências a qualquer momento através do arquivo POM.XML, como consta no artigo “Introdução ao Maven” (DEVMEDIA, 2012).

Um *framework* que possui várias dependências gerenciadas via *Maven* é o *Spring*, utilizado, na versão 2.2.4, para auxiliar na criação e inicialização do projeto, uma vez que, automaticamente, importa e configura várias dependências úteis, como descrito no livro “Spring Boot: Acelere o desenvolvimento de microsserviços” (BOAGLIO, 2017). Algumas dessas dependências instaladas pelo *Spring*, são:

**spring-boot-starter-data-jpa** Inclui o suporte do *Spring* para integração com o *JPA*, que facilita a persistência de objetos Java em um banco de dados relacional, como comentado na sessão anterior.

**spring-boot-starter-jdbc** Suporte para acesso a banco de dados por meio do Java Database Connectivity (JDBC), permitindo a conexão e execução de consultas em bancos de dados relacionais.

**spring-boot-starter-web** Conjunto de bibliotecas para desenvolvimento de aplicativos da web com o *Spring*, incluindo suporte para o framework *Spring Model-View-Controller* (MVC).

**spring-boot-starter-tomcat** Inclui o servidor web Apache Tomcat incorporado ao Spring Boot, permitindo a execução e implantação do aplicativo como um arquivo WAR.

**spring-boot-starter-test** Bibliotecas e ferramentas úteis para testar aplicativos Spring Boot, como JUnit, Mockito e, o utilizado na aplicação, Spring Test.

**spring-boot-starter-security** : Adiciona recursos de segurança ao aplicativo Spring Boot, como autenticação e autorização de usuários.

#### 4.6.1.4 JWT

Como descrito no início da sessão 4.6.1.2, o servidor é quem oferece os serviços disponíveis na aplicação, portanto, ele recebe várias requisições, ou seja, usuários remotos solicitando serviços. Para garantir que esses acessos sejam apenas de usuários devidamente cadastrados e autorizados, é preciso verificar se o solicitante tem acesso ao serviço que deseja.

Para tratar dessa questão foi implementada a dependência externa “jjwt”, que adiciona ao projeto a possibilidade de trabalhar com JSON Web Token (JWT), uma ferramenta promove o trâmite seguro de credenciais nas requisições que chegam no servidor, garantindo que os dados sejam manipulados apenas mediante autenticação, como mencionado no artigo “Utilização de json web token na autenticação de usuários em apis rest”(MONTANHEIRO; CARVALHO; RODRIGUES, 2017).

Através dessa tecnologia é possível definir quais as solicitações seriam aceitas pelo servidor mediante autenticação e quais não requerem tal verificação. A figura 4.10 apresenta um trecho da classe “WebSecurityConfig”, onde são definidas as tratativas dadas pela segurança do JWT às requisições recebidas no sistema.

```
@Override
protected void configure(HttpSecurity httpSecurity) throws Exception {
    httpSecurity
        .cors().and()
        .csrf().disable()
        .authorizeRequests()
            .antMatchers(HttpMethod.POST, "/acesso/**").permitAll()
            .antMatchers(HttpMethod.GET, "/sse/**").permitAll()
            .antMatchers(HttpMethod.POST, "/sse/**").permitAll()
            .antMatchers(HttpMethod.GET, "/api/**").authenticated()
            .antMatchers(HttpMethod.POST, "/api/**").authenticated()
            .antMatchers(HttpMethod.PUT, "/api/**").authenticated()
            .antMatchers(HttpMethod.DELETE, "/api/**").authenticated()
            .antMatchers(HttpMethod.OPTIONS, "/api/**").permitAll()
            .anyRequest().authenticated().and().exceptionHandling()
            .authenticationEntryPoint(jwtAuthenticationEntryPoint).and().sessionManagement()
            .sessionCreationPolicy(SessionCreationPolicy.STATELESS);

    httpSecurity.addFilterBefore(jwtRequestFilter, UsernamePasswordAuthenticationFilter.class);
}
```

**Figura 4.10:** Trecho da classe WebSecurityConfig no servidor

Ao acessar um serviço cuja rota seja assinalada como “authenticated()”, verifica-se se o requerente informou um token de acesso válido, o qual só pode ser obtido mediante a validação dos dados cadastrais, no momento do login. Praticamente todos os serviços fornecidos pelo servidor decorrem da rota “/api”, que demanda autenticação.

Apenas as requisições indicadas pelo método “permitAll()” são exceções a esse mecanismo. A rota “/acesso” não é autenticada por ser utilizada no login do usuário, momento em que é gerado o token que será usado nas rotas autenticadas. Essa mesma rota também leva ao serviço de cadastro, ou seja, o ato de criação da conta, no qual o usuário ainda não possui credenciais para validar.

Já a rota “/sse” é utilizada no mecanismo de comunicação em tempo real entre usuários, que será abordado em detalhes na sessão 4.6.1.7. Essa rota é usada para estabelecer uma conexão que permita o envio de eventos assíncronos do servidor para o cliente. Como a rota transmite apenas eventos, que precisam ser tramitados rapidamente e sem conteúdo em dados, não demanda autenticação.

#### 4.6.1.5 UtilPassword

Uma medida de segurança implementada é a encriptação de senhas, para que elas não sejam salvas de forma explícita no banco de dados. Para este fim, foi utilizado o Password-Based Key Derivation Function 2 (PBKDF2), sigla para “Função de Derivação de Chave Baseada em Senha 2”, é um algoritmo amplamente utilizado para criptografia, que aumenta a segurança das senhas, tornando-as mais resistentes a ataques como “força bruta” e “dicionário”.

O algoritmo foi implementado através de uma classe utilitária chamada *Util-Password*, que utiliza técnicas de “hashing” e “salting” para codificar a senha antes de armazená-la. Dessa forma, ao se cadastrar, a senha informada pelo usuário é codificada com base em códigos *hash* e *salt* aleatórios.

ABC nome	ABC login	ABC ds_hash	ABC ds_salt	ABC senha
chris	chrisley619@gmail.com	Jf6WbcZM9q	PWf58hFbjT	9f9BXvwsE1K0Ird0B3mZWYwpBhgGu2h06hPrrq4OBx4=
teste1	teste1@gmail.com	UjapHGktZn	TqHdnG7Fd1	yJ3Tu4ujkHXmaKXIdQzuhA9PRd6jAzqjYpya0JoeFjw=
teste2	userteste@gmail.com	XhuUhOraMAI	ejpAK4xdSS	jHTvbTA0C97Ac+PQPmrzV+gCfoAUFFFclN5a48FtMdw=
Mariana	mari4567@gmail.com	igsNVEn744	8MkFY7kDGT	8dXJWb4x2nY9gvH0rcZID23TVhRfLdpGoMlj33YyEno=
Marika	marikaTheEternal@hotmail.com	6JbhYYJhhx	h2Dzrl2M1K	0zdtErSvN95HJukFxDuZcR6+R41RBXOFJAJUU3605g=
Cleia Rodrigues	cleiarodriguesdossantos@ninnTKfbbeD	ow0yHROgb3	r28E9gtoclzLDjfuheGvwjBVwkkM5blIK4cXtfReDzbc=	

Figura 4.11: Registros de usuários salvos na tabela “usuários”

A figura 4.11 exibe como as credenciais e as chaves ficam salvas no banco. É perceptível que cada usuário possui um código *hash* e um *salt* únicos para codificação da senha, o que significa que o padrão de criptografia de cada senha também é único. Dessa forma, caso uma senha seja de alguma forma decodificada, as outras senhas permanecem protegidas, uma vez que cada senha teria de ser decodificada individualmente.

#### 4.6.1.6 Convenções de código

Serão consideradas algumas convenções que contribuem para a organização da aplicação em módulos de funcionalidade, se tratando de boas práticas de implementação Java, como a criação de um pacote *controller*, *entities*, *repository*, *utils*, *filter*, *dto* entre outras. As principais convenções utilizadas serão abordadas brevemente adiante.

**entities** Pacote das classes abordadas nos últimos parágrafos da sessão 4.6.1.2, as quais encapsulam dados e comportamentos de entidades, como usuários, personagens, raças, etc. As entidades, como mencionado, também são mapeadas em relação às suas respectivas tabelas no banco de dados.

**controller** Alinhado ao modelo Model-View-Controller (MVC), as classes desse tipo são responsáveis pelo recebimento das requisições HyperText Transfer Protocol (HTTP), realizando os processamentos e dando os retornos necessários. Por exemplo, recebendo uma requisição pela rota “api/salas/list” a classe “SalaController” ativará um método para consultar as salas e enviar a listagem resultante como resposta ao usuário.

**repository** Contém classes dedicadas às interações do sistema com o banco de dados, realizando operações de inserção, atualização, exclusão e consulta de dados. Separando responsabilidades a manutenção e testabilidade do código são favorecidos. Portanto, pelo exemplo anterior, quando “SalaController” recebe e trata a requisição é a classe “SalaRepository” quem efetivamente realiza a consulta no banco para obter a lista de salas.

**DTO** O pacote Data Transfer Object (DTO) é usado na transferência de dados entre camadas da aplicação, encapsulando e estruturando os dados que serão transportados conforme a necessidade. Quando, por exemplo, o usuário solicita os dados do seu personagem, ele os recebe em uma instância da classe “PersonagemDTO”, estruturada para combinar os dados das várias entidades que formam o personagem, como raça, treinamentos, etnias, etc.

**Exceptions** Implementação de exceções personalizadas, referentes a erros específicos que merecem mapeamento na aplicação. Por exemplo, a classe “RecordNotFoundException” indica que um registro não foi encontrado com os parâmetros fornecidos, já a “RecordDuplicatedException” é ativada pela verificação de uma duplicidade. Dessa forma, o uso das “exceptions” agiliza, principalmente, a identificação e tratamento de erros.

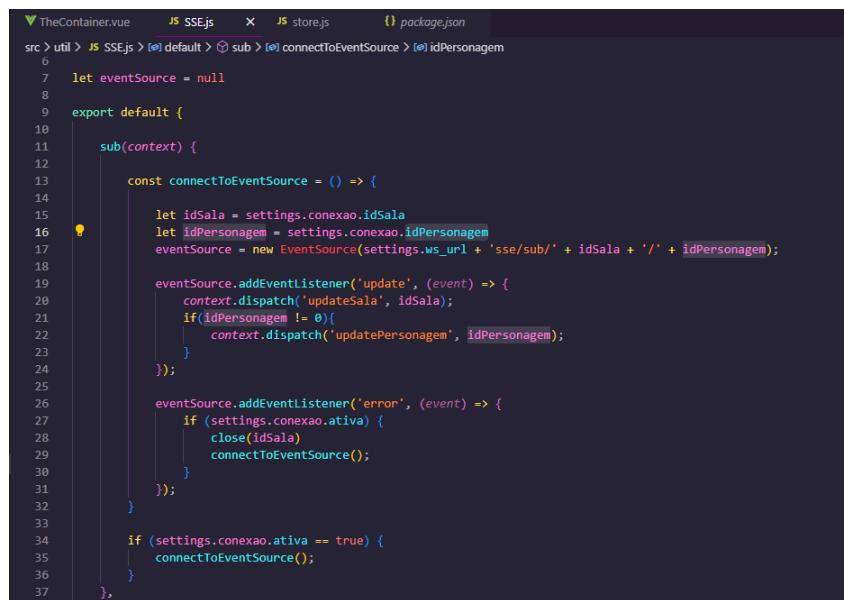
Essas e outras convenções são debatidas pela literatura especializada em livros como “Software Engineering” de Ian Sommerville (2010) e “Software Engineering: A Practitioner’s Approach” de Roger S. Pressman (2015). Naturalmente, tais convenções são amplamente adotadas por desenvolvedores Java, uma vez que contribuem para a clareza, manutenibilidade e escalabilidade do código.

#### 4.6.1.7 Server-Sent Events

Pelo procedimento de comunicação “cliente/servidor” convencional o servidor envia dados ao cliente apenas em resposta a alguma solicitação. Dessa forma, o servidor pode aprovar ou reprovar um acesso, listar salas ou personagens, registrar uma escolha, entre outras operações, porém, isso sempre ocorre mediante ação do usuário, do lado do “cliente”, o servidor apenas responde ao que foi solicitado.

Porém, neste trabalho, existem casos em que o servidor deve enviar mensagens a clientes, mesmo que eles não as tenham solicitado, o que se deve principalmente a possibilidade de interação entre jogadores. Por exemplo, se um jogador realiza um teste de capacidade com o seu personagem, ele realiza uma requisição ao servidor, que processa o teste e retorna o resultado ao solicitante, porém, o mestre também deveria ser notificado desse resultado, mesmo que a requisição tenha partido de outro usuário.

Para atender a essa demanda foi utilizado o mecanismo de comunicação assíncrona Server-Sent Events (SSE), que compõe o módulo *Spring MVC* (BAELDUNG, 2022). Essa tecnologia permite que o servidor envie eventos para o cliente sem a necessidade de uma solicitação, de forma que, quando alguma informação importante for alterada, um método de emissão é disparado e captado pelos dispositivos clientes.



```

    <?xml version="1.0" encoding="UTF-8"?>
<Workspace>
  <FileTree>
    <FileItem path="TheContainer.vue" type="Vue" selected="true" />
    <FileItem path="src/util/SSE.js" type="JavaScript" selected="true" />
    <FileItem path="src/store.js" type="JavaScript" />
    <FileItem path="package.json" type="JSON" />
  </FileTree>
  <CodeEditor>
    <Text>
      6
      7   let eventSource = null
      8
      9   export default {
     10
     11     sub(context) {
     12
     13       const connectToEventSource = () => {
     14
     15         let idSala = settings.conexao.idSala
     16         let idPersonagem = settings.conexao.idPersonagem
     17         eventSource = new EventSource(settings.ws_url + 'sse/sub/' + idSala + '/' + idPersonagem);
     18
     19         eventSource.addEventListener('update', (event) => {
     20           context.dispatch('updateSala', idSala);
     21           if(idPersonagem != 0){
     22             context.dispatch('updatePersonagem', idPersonagem);
     23           }
     24         });
     25
     26         eventSource.addEventListener('error', (event) => {
     27           if (settings.conexao.ativa) {
     28             close(idSala)
     29             connectToEventSource();
     30           }
     31         });
     32
     33         if (settings.conexao.ativa == true) {
     34           connectToEventSource();
     35         }
     36       },
     37     },
    </Text>
  </CodeEditor>
</Workspace>

```

**Figura 4.12:** Código de abertura da conexão SSE.

A primeira etapa desse mecanismo é a abertura da conexão, realizada no momento em que o usuário acessa um painel de jogo no “front-end” da aplicação, que será abordado na sessão 4.6.2. O método “sub()” é disparado mirando a rota “/sse/sub/” e passando a identificação da sala e do personagem como parâmetros, como demonstrado na imagem 4.12.

Quando o servidor recebe essa requisição ele dispara o método “subscribe()”, da classe “SSEController”, responsável pela abertura da conexão SSE. Porém, ao invés de enviar uma resposta imediata ao cliente e fechar a conexão, esse método fica pendente, criando uma conexão em aberto, através da qual o servidor pode enviar eventos ao cliente sempre que necessário.

```

    @RequestMapping( value = {"/sub/{idSala}/{idPersonagem}"}, method = RequestMethod.GET, produces = MediaType.TEXT_EVENT_STREAM_VALUE)
    public SseEmitter subscribe(@PathVariable Long idSala, @PathVariable Long idPersonagem){

        int idSalaCode = idSala.intValue();
        int idPersonagemCode = idPersonagem.intValue(); // id 0 para sala
        List<SseEmitter> listSalaEmitters = null;

        listSalaEmitters = emittersList.get(idSalaCode);
        if(listSalaEmitters == null) {
            emittersList.set(idSalaCode, new ArrayList<SseEmitter>(10));
            listSalaEmitters = emittersList.get(idSalaCode);
        }

        SseEmitter emitter = listSalaEmitters.get(idPersonagemCode);
        if(emitter == null) {
            emitter = new SseEmitter();

            emitter.onCompletion(() -> {
                emittersList.get(idSalaCode).set(idPersonagemCode, null);
            });
        }

        emittersList.get(idSalaCode).set(idPersonagemCode, emitter);

        return emitter;
    }
}

```

**Figura 4.13:** Código do controller SSE.

A imagem 4.13 exibe o *controller* que recebe a requisição realizada pelo método “sub()”. Um objeto do tipo “SseEmitter” contém os dados necessários para que o servidor possa enviar as mensagens, portanto, cada usuário que acessa um painel de sala ou de personagem gera um “emitter” no servidor.

Dessa forma, quando qualquer trâmite no servidor alterar informações de uma sala ou personagem, o servidor pode usar a conexão aberta para notificar os “emitters” interessados. Essas mensagens assíncronas são captadas no “front-end” por métodos *addEventListener*, apresentados na imagem 4.12, que disparam solicitações de atualização ao servidor, captando as alterações e exibindo os dados atualizados.

Como cada partida possui uma sala e vários personagens, a aplicação utiliza um “Array” chamado “listSalaEmitters” para armazenar os “emitter” da partida, dessa forma, ao enviar uma mensagem para todas as conexões de uma lista, o servidor se comunica com uma sala e seus personagens. Naturalmente, outra sala com outros personagens, devem ter seus “emitters” salvos em outra lista.

Para identificar cada partida o servidor mantém uma lista de listas de conexões, ou seja, um “Array” chamado “emittersList”, em que cada elemento é um “listSalaEmitters” armazenando objetos “SseEmitter”. Para que seja possível identificar as partidas na “emittersList”, o “index” de cada lista de “emitter” corresponde ao “id” da sala em que a partida ocorre.

A imagem 4.14 exibe o método responsável pelo envio das mensagens. Ele recebe o “idSala” como parâmetro, busca na “emittersList” o “index” correspondente, obtém a “listSalaEmitters” da partida, e envia uma mensagem para todos os “SseEmitter” da lista encontrada. Dessa forma, a sala e os personagens vinculados a ela recebem a mensagem via conexão SSE.

```
public void updateSala(Long idSala) {

    int idSalaCode = idSala.intValue();
    if(emittersList.get(idSalaCode) != null) {
        for (SseEmitter emitter : emittersList.get(idSalaCode)) {
            if(emitter != null) {
                try {
                    emitter.send(SseEmitter.event()
                        .data("")
                        .name("update"));
                } catch (IOException e) {
                    System.out.println("ERRO SSendMessage: " + e.getMessage());
                    //throw new BadRequestException(e.getMessage());
                }
            }
        }
    }
}
```

**Figura 4.14:** Método *updateSala* no servidor.

O método “updateSala” é disparado pelo servidor em diferentes momentos, como quando uma solicitação de acesso é registrada, ou quando um personagem realiza um teste. Como demonstrado na imagem ??, que demonstra o método “sub()”, assim que um evento é recebido no “cliente” os métodos de atualização de sala e personagens são disparados, atualizando os dados logo após as alterações no servidor.

A comunicação SSE é mais instável que as requisições “HTTP” convencionais, e os objetos “sala” e “personagem” possuem certo volume de dados. Além disso, para assegurar a velocidade da resposta, a rota “/sse”, não é autenticada no servidor, como consta na imagem 4.10 que mostra a tratativa das rotas pelo “JWT”. Por esses motivos, a mensagem é apenas uma notificação de atualização, e não os dados atualizados em si.

No contexto da aplicação, uma “notificação de atualização” é uma mensagem sem conteúdo, tendo assim o menor consumo de banda, e nenhuma dependência ou utilidade para os dados tramitados. Para a aplicação interessa apenas a recepção do evento em si, e não os dados contidos nele. Uma vez detectado o evento, é solicitada a atualização dos dados por meio de rotas autenticadas convencionais.

Sempre que a conexão SSE tem seu tempo limite esgotado, uma nova conexão é realizada automaticamente, mantendo a conexão ativa até que o usuário feche a aplicação ou saia do painel de jogo. Também vale comentar que os dados atualizados são salvos no estado central do sistema, implementado com “Vuex”, conforme citado na sessão 4.6. Todos os componentes utilizam o estado central do sistema como fonte de dados, portanto, todas as interfaces que formam o painel estão sempre atualizadas.

## 4.6.2 Front-End

Nas sessões anteriores foram discutidas as funcionalidades que compõe a camada “dos bastidores” do sistema, o “Back-End”, composto pelo servidor e banco de dados. Agora, será abordada a camada que opera do lado do cliente, o “Front-End”, a camada de interface, responsável pela interação com o usuário final, como mencionado em “Distributed Systems: Principles and Paradigms” (TANENBAUM; STEEN, 2009).

O “Front-End” é uma parte da aplicação que será executada em dispositivos clientes, ou seja, no próprio computador, celular ou tablet do usuário. Manipulando esta interface os usuários podem interagir com a aplicação, gerando os dados de entrada que serão processados pelo servidor e exibidos como resultado na tela.

Portanto, é nessa camada que são projetados elementos visuais que compõe as telas do sistema, como as páginas, formulários, botões, menus e outros componentes de interface. Se tratando de uma interface WEB, executada em navegadores, são combinadas 3 tecnologias principais para elaborar esses elementos, o Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) e JavaScript.

O HTML é usado para estruturar e organizar o conteúdo da página, os textos, botões, etc. O CSS atua na definição do estilo e a aparência de tais elementos, como a cor dos botões, tipo de fonte, etc. Já o JavaScript é a linguagem de programação executada pelo navegador, responsável pelas operações lógicas da interface.

Combinando esses elementos, a interface do sistema pode ser executada em navegadores, operando em vários dispositivos. Entretanto, foram necessárias mais ferramentas para que essa camada do sistema atenda às demandas da aplicação, como a criação de interfaces mais complexas, melhor estilizadas e com funcionalidades adicionais. As sessões seguintes abordam as tecnologias implementadas nesse sentido.

### 4.6.2.1 VueJS

O “*VueJS*”, em sua versão 2.6.12, foi implementado como o principal *framework* usado no “Front-End”. O livro “*Vue.js - Up and Running*” de Callum Macrae 2018, dedicado a apresentar e ensinar este *framework*, e menciona ainda no prefácio que o “*VueJS*” vem ganhado popularidade entre os desenvolvedores, possui uma comunidade ativa e se trata de uma das tecnologias mais bem avaliadas do “GitHub”.

O *VueJS* permite o desenvolvimento das interfaces através de componentes, nos quais a lógica e os elementos visuais são encapsulados, podendo ser reutilizados e combinados para criar interfaces complexas. Dessa forma, o *framework* se mostra flexível o suficiente para se adequar a diferentes tipos de projetos, desde pequenas aplicações até grandes sistemas empresariais.

Com o *VueJS* é possível vincular os elementos visuais da interface diretamente com os dados tratados na aplicação, de forma que as modificações computadas refletem automaticamente nos elementos visuais, sem a necessidade de controladores de estado para todas as variáveis. Isso é de suma importância nessa aplicação, uma vez que as informações da sala e dos personagens estão sempre mudando no decorrer da partida.

Nos casos em que o gerenciamento de estado é necessário, será utilizada a biblioteca *Vuex*, cuja documentação pode ser encontrada na página <https://vuex.vuejs.org>. Através dessa ferramenta é criado um estado centralizado acessível a qualquer componente “*VueJS*”, dessa forma, uma única requisição pode trazer um objeto atualizado, fornecendo dados para vários componentes.

Outra ferramenta utilizada é o “*CoreUI Free Admin Panel*”, um template código aberto de um painel administrativo, baseado em *VueJS* e *BootstrapVue*. O template foi utilizado apenas uma estrutura base, para agilizar o processo de desenvolvimento das interfaces, um ponto de partida para a implementação dos trabalhos.

O template foi totalmente customizado, ao que pode ser verificado comparando sua interface original, disponível na página <https://coreui.io/#live-preview>) e a interface final da aplicação, no capítulo 5. Foram mantidos apenas alguns componentes, como o menu lateral, e a base da estrutura de diretórios.

#### 4.6.2.2 Estilização

O “*Bootstrap*” foi a principal ferramenta utilizada na estilização das interfaces do sistema. Esse *framework* é amplamente utilizado por facilitar a criação de interfaces belas e responsivas, oferecendo componentes CSS responsivos pré-estilizados, como grupos de botões, formulários, barras de navegação, tabelas, barras de progresso, etc.

Uma das principais características do “*Bootstrap*” é o seu sistema de grade, que subdivide a tela em 12 colunas que auxiliam na criação da responsividade das interfaces. Por exemplo, em uma tela com largura a partir de 960px, um determinado elemento pode ser configurado para ocupar 4 das 12 colunas da grade, porém, em uma tela de 540px de largura, o mesmo elemento pode ocupar 6 das 12 colunas.

Uma vez que as interfaces foram desenvolvidas com base no “*VueJS*”, o “*Bootstrap*” foi implementado através do *framework* “*BootstrapVue*”, uma versão do “*Bootstrap*” adaptada aos recursos do “*VueJS*”, promovendo a combinação entre as duas tecnologias para criar interfaces dinâmicas e interativas.

A documentação do “*BootstrapVue*” é acessível na página <https://bootstrap-vue.org/docs/components>, que apresenta as suas características e funcionalidades na íntegra. Nesse trabalho, essa tecnologia foi utilizada para a criação de telas responsivas, com elementos complexos e customizáveis, as quais serão exibidas em detalhes no capítulo 5.

#### 4.6.2.3 Elementos temporários

No desenvolvimento das interfaces foram priorizadas as funcionalidades, portanto, através das ferramentas apresentadas no decorrer das sessões anteriores, o “Front-End” da aplicação foi desenvolvido de maneira funcional como esperado. Porém, existem vários conteúdos visuais que compõe as telas de um sistema, como textos informativos, ícones que favorecem a navegação intuitiva e/ou economizam espaço de tela, imagens que atuam na composição da identidade visual, etc.

A produção ou aquisição de tais conteúdos visuais não são prioridade neste trabalho, portanto, foram utilizados elementos visuais gratuitos e temporários, os quais serão revisados e substituídos em trabalhos futuros. A inserção de tais elementos foi necessária para que suas posições sejam mapeadas na estrutura das telas de antemão, permitindo a correta definição das escalas e posicionamento dos elementos que contém, ou dividem espaço, com essas imagens, ícones, textos, etc.

Os ícones utilizados na aplicação são todos gratuitos, tendo sido disponibilizados pelo próprio *BootstrapVue* e pelo banco de ícones *flaticon*, acessível na página <https://www.flaticon.com.br/>. Em alguns casos específicos, foi necessário customizar algum ícone, para isso foi utilizado o “software” *photoshop* da Adobe.

O mesmo ocorre com as imagens, obtidas nos bancos de imagens *picsum* e *pinterest*, acessíveis, respectivamente, nas páginas <https://picsum.photos> e <https://br.pinterest.com>). Além da construção das interfaces, as imagens também facilitam a compreensão de certas telas, melhorando a didática deste trabalho e sua apresentação, o que se aplica principalmente no que se refere a certas opções na criação de personagens.

Porém, é válido pontuar que as imagens que ilustram, por exemplo, as opções de etnias, nem mesmo indicam à real aparência das etnias a qual são atribuídas, uma vez que as espécies e raças do jogo são autorais, não sendo possível encontrar ilustrações que representem sua anatomia corretamente.

A única imagem de produção autoral é a ilustração do mapa do jogo na tela principal, que será, assim como todas as imagens mencionadas, exibida no decorrer do capítulo 5. A referida ilustração é de autoria própria, e mostra um mapa geográfico do mundo onde o jogo se passa. As ferramentas utilizadas na concepção dessa imagem são o software *photoshop* e a plataforma *inkarnate*, disponível em <https://inkarnate.com>).

Outro elemento muito importante são os textos dispostos na aplicação, como o texto de boas-vindas, as descrições das opções na criação de personagens, dispondo informações sobre a anatomia, origem, cultura e outras características que descrevem às espécies, raças ou etnias do jogo, ou mesmo as definições das opções de qualidade, defeito, treinamento, etc. De maneira geral, os textos inseridos são apenas “placeholders”, usados para demarcação de espaço.

# 5 RESULTADOS

Este capítulo apresenta os resultados da utilização das ferramentas e técnicas abordados no capítulo 4, demonstrando a aplicação implementada e seu funcionamento. Como mencionado na sessão 2.3, é necessário que os usuários definam, antecipadamente, algum canal ativo de comunicação, seja presencial ou remoto. Eles também devem decidir qual deles deve assumir o papel de mestre, enquanto os demais participantes assumem o papel de jogadores.

## 5.1 Acesso ao sistema

Decididos os preparativos, cada usuário deve se cadastrar na aplicação. Como mencionado na sessão 4.2.1 o sistema conta com uma tela de cadastro simplificada, apenas com os campos necessários, como demonstrado na imagem 5.1.

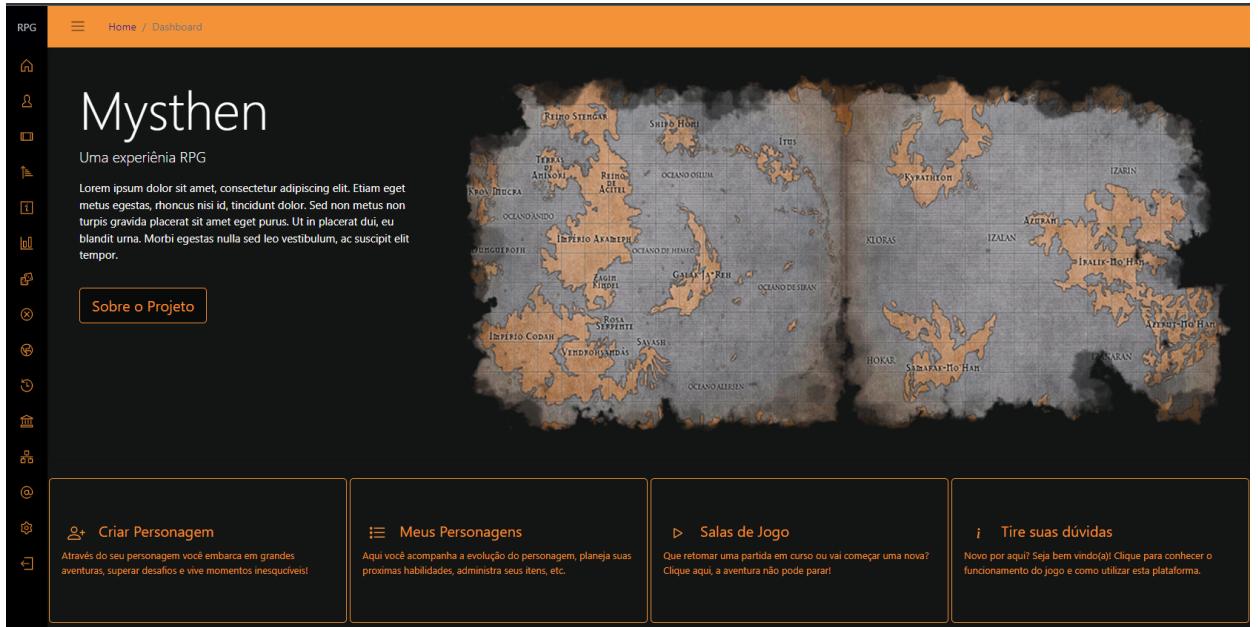


**Figura 5.1:** Tela de cadastro

Ao preencher os dados o usuário confirma a criação da conta, e pode realizar o login para acessar o sistema. Os mecanismos de segurança operantes nesse processo foram abordados nas sessões 4.6.1.4 e 4.6.1.5.

## 5.2 Tela Principal

Uma vez logado, o usuário tem acesso à tela principal da aplicação, a qual está sendo exibida na figura 5.2. Conforme mencionado ao longo da sessão 4.2 e 4.6.2.3, as funcionalidades foram priorizadas em detrimento de conteúdos informativos e visuais, portanto, é notável a presença de textos “placeholders” nessa e em outras telas que serão exibidas no decorrer deste capítulo.



**Figura 5.2:** Tela principal da aplicação

Os usuários podem navegar pelo menu lateral, ou pelas opções destacadas na tela principal, para serem direcionados para a funcionalidade desejada. Utilizando as ferramentas de estilização mencionadas na sessão 4.6.2, foi implementado um menu lateral na forma de uma coluna de ícones, sobre os quais, ao posicionar o cursor do mouse, apenas o submenu correspondente é exibido.

Por se tratar de uma aplicação responsiva, esse e outros elementos podem ser visualizados de diferentes maneiras, a depender das dimensões do dispositivo e preferências do usuário. No caso do menu lateral, o mesmo pode ser exibido como coluna de ícones (ocupando pouco espaço na tela), ou pode ser expandido para mostrar as opções do menu por escrito, ou ainda, pode ser ocultado completamente.

Outro elemento da interface notável já na tela principal, é a barra superior, onde foi implementado um elemento “Breadcrumb”, que permite ao usuário saber o caminho percorrido até a página atual. Naturalmente, este elemento é fixo, sendo exibido ao usuário independentemente da tela onde ele esteja inserido. Excepcionalmente os painéis de jogo contam com uma barra superior personalizada, como será abordado adiante.

Demonstrando a responsividade, implementada através das tecnologias discutidas na sessão 4.6.2.2, a figura 5.3 exibe a mesma tela principal da figura 5.2, porém, na tela reduzida de um dispositivo móvel.



**Figura 5.3:** Tela principal da aplicação em dispositivo mobile

É nítida a capacidade de reajuste dos elementos da interface a qualquer tamanho de tela, alternando proporções e posicionamentos automaticamente conforme configurado. Nesse sentido, ao detectar uma tela de proporções reduzidas o menu lateral fica totalmente oculto, para exibi-lo o usuário deve clicar o botão  $\equiv$  que expande o menu para viabilizar a navegação.

## 5.3 Opções de navegação

Analizando a figura 5.4, que demonstra o menu lateral expandido, é possível verificar as opções de navegação disponibilizadas pelo sistema. Dentre as opções, o item “Home” redireciona o usuário para a tela principal, já “Personagem” e “Sala” encaminham para a criação e listagem de personagens e salas.



**Figura 5.4:** Exibição das opções do menu lateral

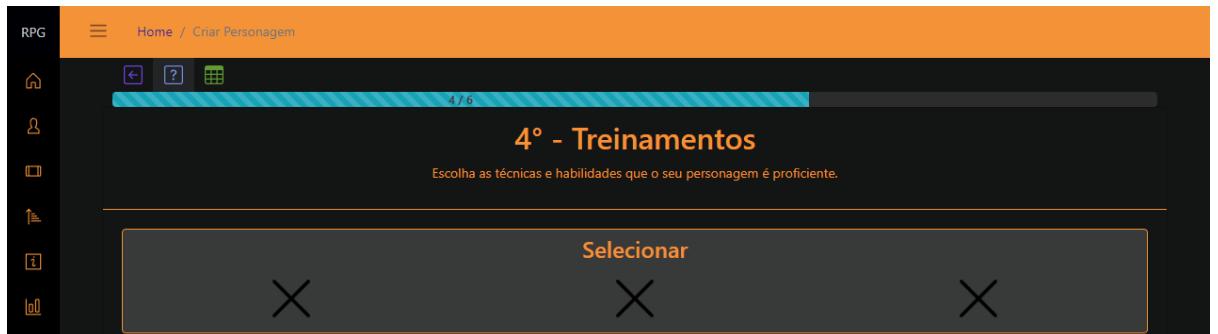
As funcionalidades dos demais itens do menu fogem ao escopo definido na sessão 4.2, porém, como foi determinado que mesmo tais funções devem ser mapeadas na interface, seus respectivos ícones, nomes e posições no menu foram inseridos.

Nesse sentido, o item “Progressão” seria referente a obtenção de habilidades especiais, magias, e outros meios de evolução dos personagens, sendo implementado em trabalhos futuros. O mesmo ocorre com os itens “Regras” e “Cenário”, por onde o usuário poderá navegar por páginas de texto estático de “conteúdo informativo”.

O conteúdo do item “Regras” equivaleria ao “Livro de Regras” no sistema, onde o usuário poderia ler sobre o funcionamento do jogo. Da mesma forma, os menus de “Cenário” trariam informações sobre o mundo onde o jogo se passa, permitindo que os usuários o conheçam, para que possam criar suas histórias, personagens e outros elementos narrativos que se passam nesse mundo.

## 5.4 Criação de Personagem

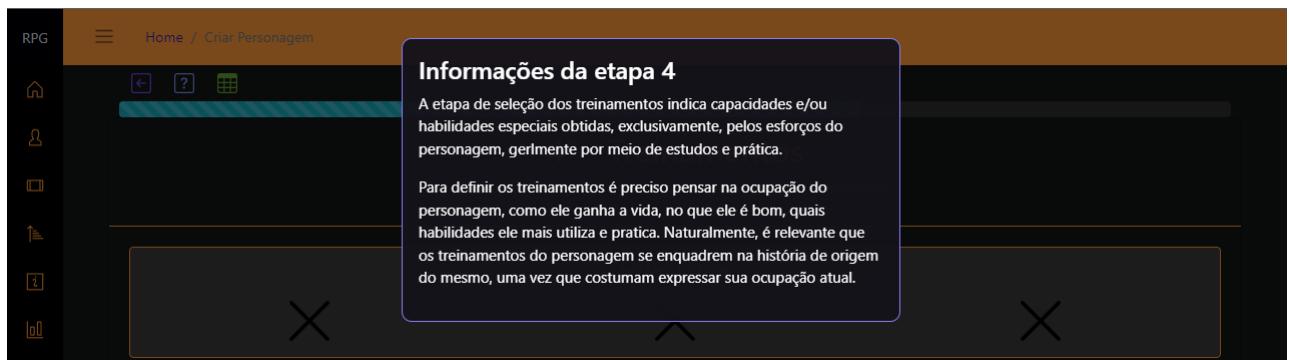
Acessando a opção para criar personagem, o usuário é direcionado à página onde ele pode iniciar o processo de criação. Como demonstrado na sessão 4.3.3, o processo de criação de personagens consiste na análise das opções oferecidas pelo sistema, e na seleção daquela que mais agrade o usuário.



**Figura 5.5:** Cabeçalho da tela de criação de personagens

Na imagem 5.5 é exibido o cabeçalho padrão para todas as telas da criação de personagem. Nele, logo acima do título da etapa, foi inserida uma barra de progresso, a qual indica em qual etapa da criação o usuário se encontra em relação ao total de etapas.

O primeiro botão acima da aba, indicado pelo ícone de uma “seta”, possui a função de retorno, permitindo ao usuário navegar para a etapa anterior, substituindo a opção selecionada, e limpando todas as escolhas seguintes, conforme explicado no fluxo alternativo “Acessar etapa anterior” do caso de uso 4.3.3.



**Figura 5.6:** Aba informativa da tela de criação de personagens

A imagem 5.6 demonstra a implementação do segundo botão da aba, indicado por um sinal de interrogação. Sua função é abrir uma janela que exibe informações sobre a etapa atual, como demonstrado na referida imagem. O objetivo é que o usuário saiba o significado de cada etapa, e tome as decisões que façam melhor sentido estratégico e narrativo. Naturalmente, em versões futuras podem ser adicionadas dicas e exemplos mais completos nessa janela, para facilitar a compreensão e instruir melhor o usuário.

A imagem 5.7 demonstra a implementação botão com ícone de tabela, que exibe a relação de capacidades obtidas e os pontos acumulados até o momento.

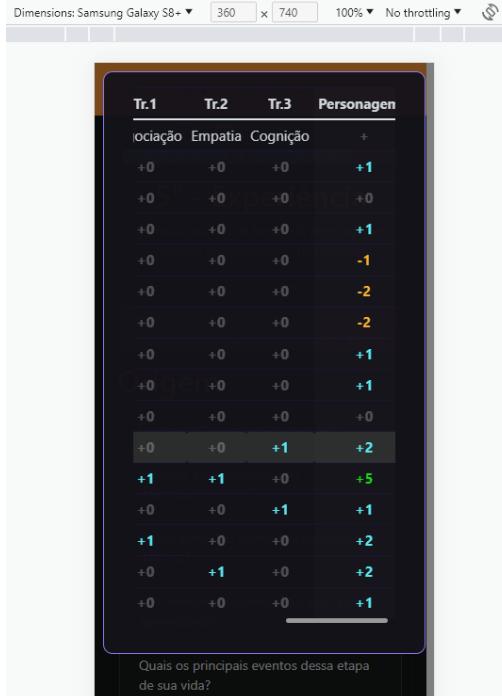


The screenshot shows a mobile RPG game's character creation screen. On the left is a vertical toolbar with icons for home, profile, settings, and other game functions. The main area features a table titled "Relação de capacidades do personagem em criação". The columns are: Capacidade, Especie, Raça, Qualidade, Defeito, Tr.1, Tr.2, Tr.3, and Personagem. The rows list various abilities like FORÇA, MOBILIDADE, RESISTÊNCIA, etc., with their respective values and points. A green +3 is visible in the Tr.3 column for AGILIDADE.

Capacidade	Especie	Raça	Qualidade	Defeito	Tr.1	Tr.2	Tr.3	Personagem
	Igneray	Sarok	Sagacidade	Antipatia	Negociação	Investida	Acrobacia	+1
FORÇA	-1	+1	+0	+0	+0	+1	+0	+1
MOBILIDADE	+1	+0	+0	+0	+0	+0	+0	+1
RESISTÊNCIA	+0	+1	+0	+0	+0	+0	+0	+1
AGILIDADE	+1	+0	+0	+0	+0	+1	+1	+3
SUTILEZA	+1	+0	+0	+0	+0	+0	+0	+1
DESENVOLTURA	+0	+0	+0	+0	+0	+0	+1	+1
TENACIDADE	+0	+0	+0	+0	+0	+0	+0	+0
VIGOR	+1	+0	+0	+0	+0	+0	+0	+1
IMUNIDADE	+0	+0	+0	+0	+0	+0	+0	+0
RAZÃO	+0	+0	+1	+0	+0	+0	+0	+1
OBSTINAÇÃO	+0	+0	+1	-1	+1	+0	+0	+1
ATENÇÃO	+1	+0	+2	-1	+0	+0	+0	+2
RETÓRICA	+0	+0	+1	+0	+1	+0	+0	+2
CARISMA	+0	+0	+0	-3	+0	+0	+0	-3
IMPOSIÇÃO	+0	+0	+0	+0	+0	+0	+0	+0

**Figura 5.7:** Relação de capacidades do personagem em criação

Ressaltando a responsividade, a imagem 5.8 demonstra a versão mobile da tela. A barra de rolagem horizontal permite a navegação por todas as colunas da tabela.



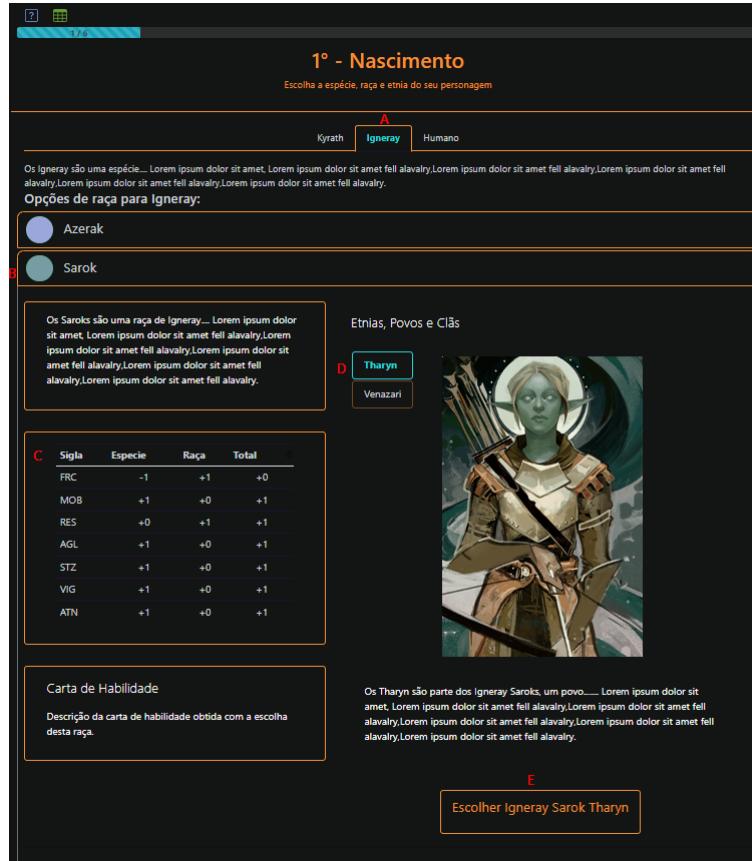
This screenshot shows the mobile version of the character creation screen. It includes a header with device dimensions (Samsung Galaxy S8+, 360x740 pixels, 100% zoom, no throttling), a toolbar, and a scrollable table. The table has columns for Tr.1, Tr.2, Tr.3, and Personagem. The rows list various abilities with their values and points. A horizontal scrollbar at the bottom allows navigation through the columns. A footer asks, "Quais os principais eventos dessa etapa de sua vida?"

Tr.1	Tr.2	Tr.3	Personagem
Negociação	Empatia	Cognição	+1
+0	+0	+0	+1
+0	+0	+0	+0
+0	+0	+0	+1
+0	+0	+0	-1
+0	+0	+0	-2
+0	+0	+0	-2
+0	+0	+0	+1
+0	+0	+0	+1
+0	+0	+0	+0
+0	+0	+1	+2
+1	+1	+0	+5
+0	+0	+1	+1
+1	+0	+0	+2
+0	+1	+0	+2
+0	+0	+0	+1

**Figura 5.8:** Relação de capacidades do personagem em criação em dispositivo móvel

O restante do conteúdo varia conforme a etapa vigente, dispondo os dados das tabelas do esquema “Game Data” nas interfaces de seleção.

**Etapa 1 - Nascimento** Na primeira etapa o usuário seleciona a espécie, raça e etnia de seu personagem, a imagem 5.9 exibe essa interface de seleção. É importante ressaltar que apenas algumas das opções de escolha do jogo foram implementadas, além disso, a arte e os textos informativos que aparecem nessa tela, são apenas *placeholders*, e serão obrigatoriamente substituídos pelos conteúdos definitivos em versões futuras.



**Figura 5.9:** Primeira etapa da criação de personagem

Manipulando as abas indicadas pela legenda **A** o usuário alterna as opções de espécie selecionáveis, e nas abas indicadas pela legenda **B**, o usuário alterna as opções de raça. O usuário pode analisar a relação de pontos de capacidade obtidos na escolha de determinada espécie e raça na tabela indicada pela legenda **C**.

Para selecionar a etnia do personagem, o usuário manipula as abas indicadas pela legenda **D**, o que também altera a imagem (e o texto descritivo) que ilustra a aparência física de um personagem da espécie, raça e etnia definidos até o momento.

Obviamente, cada espécie possui suas próprias raças, e uma raça possui suas próprias etnias, não existindo a possibilidade de, por exemplo, uma raça ser comum a duas espécies diferentes. Portanto, escolher uma etnia é, necessariamente, escolher uma raça e uma espécie, motivo pelo qual as 3 escolhas podem ser facilmente expressas em um único botão, como indicado pela legenda **E**.

**Etapa 2 - Qualidade** Na segunda etapa o usuário seleciona uma qualidade para o seu personagem, ou seja, uma característica vantajosa, natural ou adquirida. O sistema exibe uma lista com 12 opções de qualidade.



**Figura 5.10:** Segunda etapa da criação de personagem

Cada opção compõe um “card”, o qual exibe o nome da qualidade, sua descrição (textos *placeholders* na versão atual), a relação de pontos de capacidade obtidos com a escolha, e uma opção para seleção, como demonstrado na figura 5.10, que foi ampliada para exibir todas as opções.

**Etapa 3 - Defeito** Na terceira etapa o usuário seleciona um defeito para o seu personagem, ou seja, uma característica negativa, natural ou adquirida. O sistema exibe uma lista com 12 opções de defeito.



Figura 5.11: Terceira etapa da criação de personagem

Como o VueJs favorece a criação de componentes reutilizáveis, foi possível utilizar a mesma estrutura de “cards” da lista de qualidades na interface de defeitos, como demonstrado na figura 5.11, alterando apenas a fonte dos dados exibidos, sem repetição de código. A imagem exibe a interface em tamanho real, não ampliada, sendo possível notar a barra de rolagem vertical que viabiliza a navegação pelas opções.

```

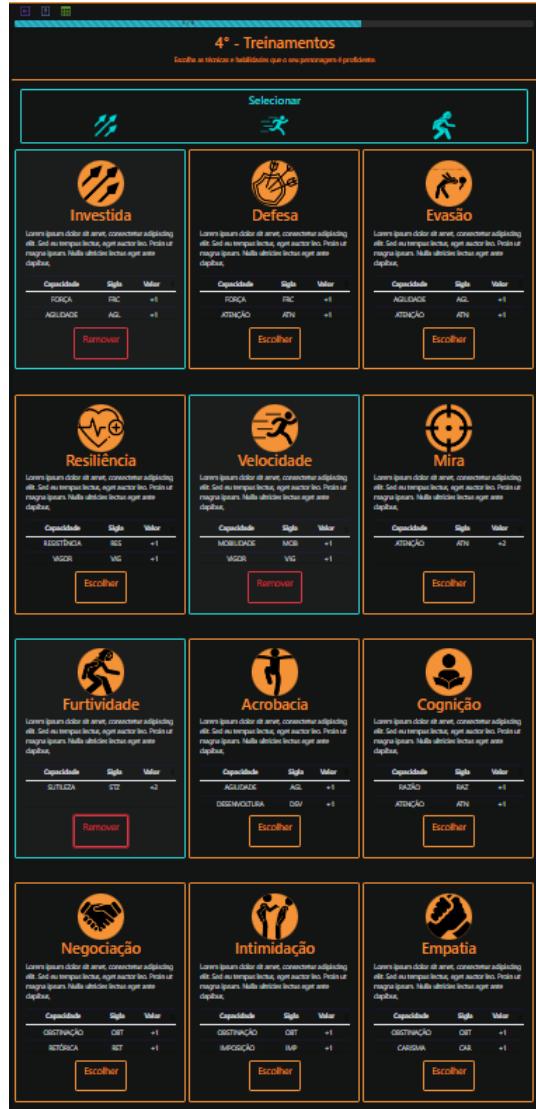
120
121@RequestMapping(method = RequestMethod.GET, value = "/caracteristica/{tipo}", produces = MediaType.APPLICATION_JSON_VALUE)
122public List<GameCaracteristica> getCaracteristicas(@PathVariable("tipo") TipQualidadeDefeito tipo) throws Exception {
123
124    List<GameCaracteristica> listCaracteristicas = new ArrayList<GameCaracteristica>();
125    try {
126        listCaracteristicas = caracteristicaRepository.findByTipo(tipo);
127    } catch (Exception e) {
128        throw new RecordNotFoundException("Erro ao obter a lista de características");
129    }
130
131    return listCaracteristicas;
132}
133
134@RequestMapping(method = RequestMethod.GET, value = "/treinamentos", produces = MediaType.APPLICATION_JSON_VALUE)
135public List<GameTreinamento> getTreinamentos() throws Exception {
136
137    List<GameTreinamento> listTreinamentos = new ArrayList<GameTreinamento>();
138    try {
139        listTreinamentos = treinamentoRepository.findAll(Sort.by("id"));
140    } catch (Exception e) {
141        throw new RecordNotFoundException("Erro ao obter a lista de treinamentos");
142    }
143
144    return listTreinamentos;
145}
146
147@RequestMapping(method = RequestMethod.GET, value = "/conhecimentos", produces = MediaType.APPLICATION_JSON_VALUE)
148public List<GameConhecimento> getConhecimentos() throws Exception {
149
150    List<GameConhecimento> listConhecimentos = new ArrayList<GameConhecimento>();
151    try {
152        listConhecimentos = conhecimentoRepository.findAll();
153    } catch (Exception e) {
154        throw new RecordNotFoundException("Erro ao obter tipos de conhecimentos");
155    }
156
157    return listConhecimentos;
158}

```

Figura 5.12: Trecho do controller CriaPersonagem, no servidor

Vale comentar que, para cada etapa, uma requisição busca os dados no servidor, como demonstrado na figura 5.12. Os dados das tabelas do esquema “Game Data” não podem ser criados, alterados ou removidos por usuários, logo, não foi necessária a criação de *controllers* dedicados para tratar esse tipo de requisição, ao invés disso, um *controller* “CriaPersonagem”, oferece os métodos de listagem dos dados de cada etapa.

**Etapa 4 - Treinamento** Na quarta etapa o usuário deve selecionar 3 treinamentos, ou seja, práticas que o personagem desenvolveu por prática, experiência, estudos, etc.

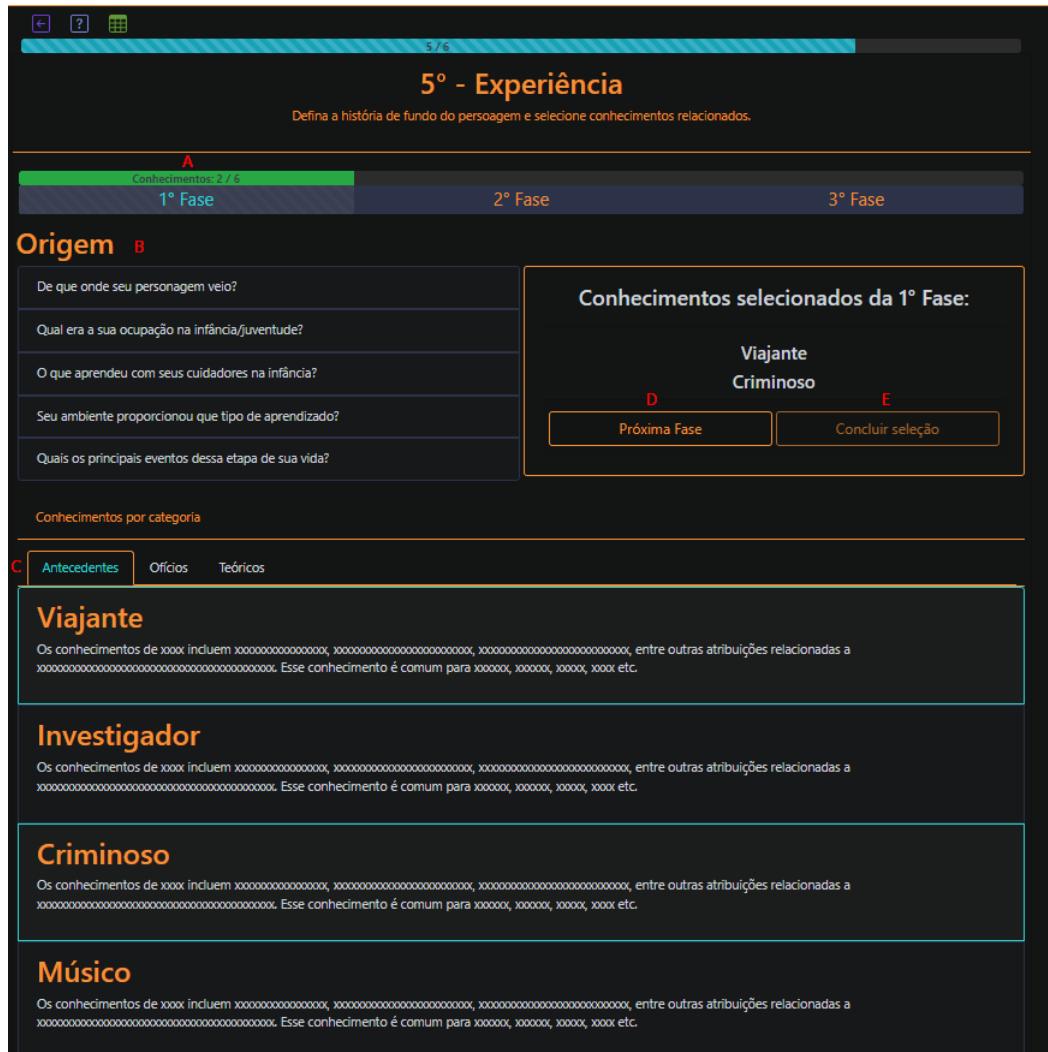


**Figura 5.13:** Quarta etapa da criação de personagem

A figura 5.13 exibe o mecanismo de seleção de treinamentos. A cada “card” selecionado, o ícone correspondente é exibido na barra superior (abaixo do título), confirmando a seleção. Além disso, a cor da borda do card selecionado é modificada, juntamente com o botão de seleção, que dá lugar a um botão de remoção da opção selecionada.

O usuário deve escolher exatamente 3 treinamentos, uma vez selecionados, a barra de rolagem ajusta a navegação para o topo da tela, para que o usuário possa confirmar sua escolha e concluir a etapa, selecionando a barra com os ícones referentes as opções de treinamento selecionadas.

**Etapa 5 - Conhecimentos** Na quinta etapa o usuário deve selecionar 2 conhecimentos para cada fase da história do personagem.



**Figura 5.14:** Quinta etapa da criação de personagem

A imagem 5.14 demonstra a interface de seleção de conhecimentos. A barra de progresso navegável, indicada pela legenda **A**, indica a fase para a qual os conhecimentos serão selecionados, sendo “Origem”, “Desenvolvimento” e “Atual”.

O sistema exibe, além da janela de informações no cabeçalho da tela (como explicado no início da sessão 5.4), uma lista de questionamentos, como indicado pela legenda **B**, que o convida o usuário a refletir a respeito das origens narrativas de suas escolhas anteriores, imaginando uma história para o personagem.

A história é expressa no sistema pelos conhecimentos obtidos em cada fase. Manipulando as abas indicadas pela legenda **C** o usuário alterna a categoria de conhecimentos, trocando a lista de opções de conhecimentos exibidos. O botão indicado pela legenda **D** permite ao usuário alternar entre as fases, e o botão indicado pela legenda **E** conclui a etapa de seleção dos conhecimentos.

**Finalização** Ao fim da criação do personagem, o usuário pode inserir um nome, sobrenome e um nome alternativo (como um apelido, título, etc), em seguida, o usuário pode confirmar a criação do personagem, como indicado pela imagem 5.15.

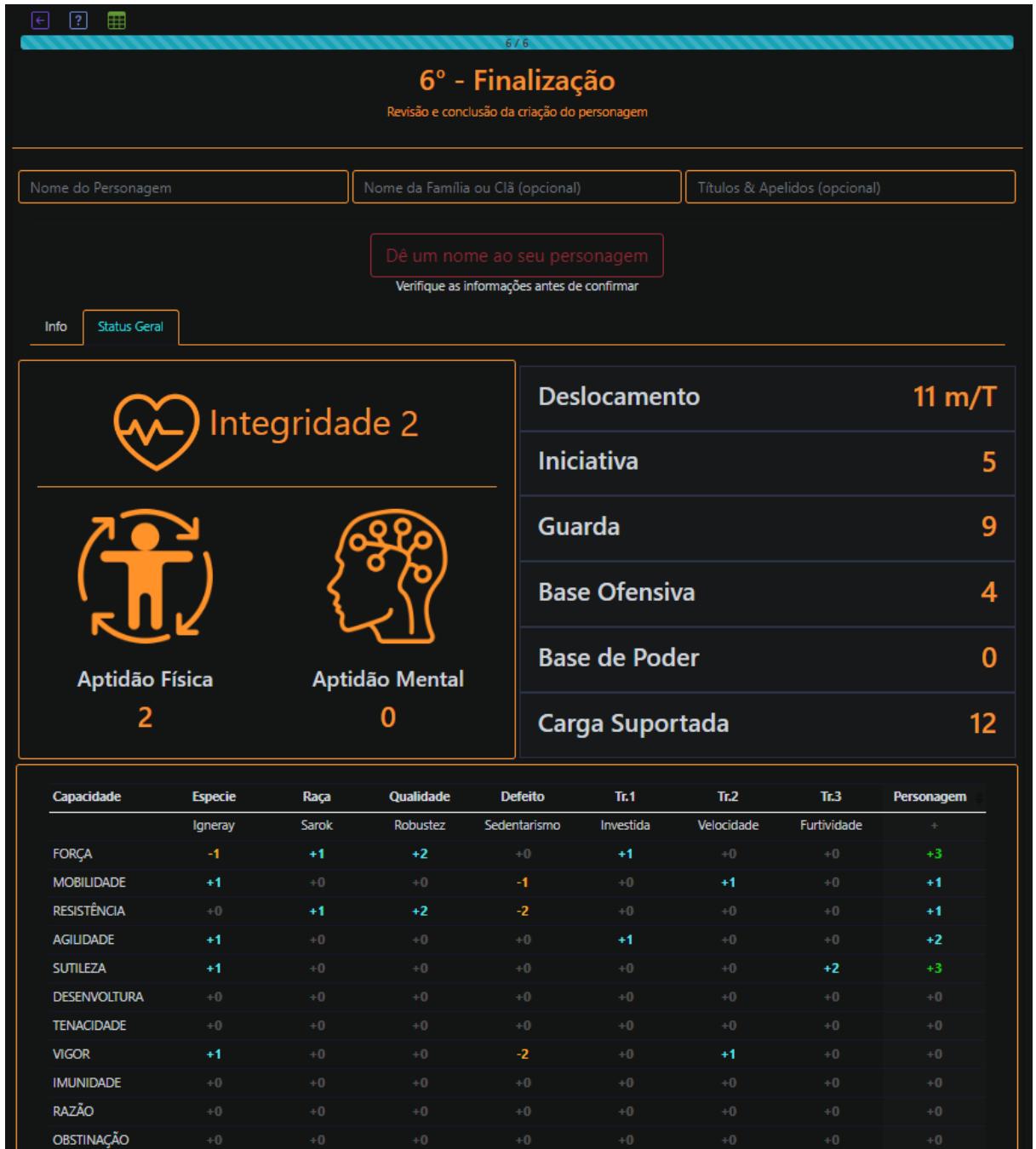


Figura 5.15: Quinta etapa da criação de personagem

A aplicação utiliza os pontos distribuídos a cada etapa para calcular os valores dos atributos de jogo, como integridade, aptidão física e mental, deslocamento, etc. Como visível na imagem 5.15, a interface exibe esses valores calculados, tal como a relação de pontos obtidos a cada etapa (reaproveitando o já mencionado componente ativado pelo ícone de tabela no cabeçalho das telas de criação de personagem).

Em seguida, o sistema valida os dados do usuário e de cada etapa selecionada, como demonstrado na figura 5.16, que exibe um trecho do método de criação do personagem no “ControllerPersonagem”.

```
@RequestMapping(method = RequestMethod.POST, value = "/novo", produces = MediaType.APPLICATION_JSON_VALUE,
public ResponseEntity<Object> create(@RequestBody PersonagemFilter personagem, BindingResult result)
throws Exception {
    Usuario usuarioLogado = new Usuario();
    try {
        usuarioLogado = usuarioRepository.findByLogin(personagem.getUsuario());
    } catch (Exception e) {
        throw new RecordNotFoundException("Erro ao verificar usuário");
    }

    GameEspecie especie = new GameEspecie();
    try {
        especie = especieRepository.findById(personagem.getIdEspecie()).get();
    } catch (Exception e) {
        throw new RecordNotFoundException("Erro ao verificar espécie");
    }

    GameRaca raca = new GameRaca();
    try {
        raca = racaRepository.findById(personagem.getIdRaca()).get();
    } catch (Exception e) {
        throw new RecordNotFoundException("Erro ao verificar raça");
    }

    GameEtnia etnia = new GameEtnia();
    try {
        etnia = etniaRepository.findById(personagem.getIdEtnia()).get();
    } catch (Exception e) {
        throw new RecordNotFoundException("Erro ao verificar etnia");
    }
}
```

**Figura 5.16:** Trecho do ControllerPersonagem

Após as validações, os dados do personagem são salvos na tabelas do esquema “Personagem”. Também ocorre, na tabela “usuarios\_personagens”, o registro do vínculo entre o personagem e o usuário criador.

## 5.5 Listagem de Personagem

Personagem	Race	Strength	Health	Intelligence
Pluma Jade	Ventralina	2	2	0
Klaus Kyevn	-	0	1	1
Misin Hakar	o sábio	0	2	2
Jeren Eleonor	Vibora do deserto	1	2	1

**Figura 5.17:** Interface de listagem de personagens

A relação dos personagens criados fica disposta na listagem de personagens, conforme demonstrado pela imagem 5.17. Naturalmente, são listados apenas os personagens de autoria do usuário logado. Ao selecionar uma opção, o usuário tem acesso ao painel de jogo do personagem em questão.

## 5.6 Criação de Salas

Enquanto cada jogador cria seu personagem, o mestre planeja a aventura em que consiste o jogo, uma função criativa não gerenciada pelo sistema. Como gerenciador do jogo, o mestre deve criar a sala que conecta os personagens dos jogadores, através da interface exibida na imagem 5.18.



**Figura 5.18:** Interface de criação de salas

Nessa tela o mestre define o nome, código de acesso e disponibilidade da sala. Antes de permitir a criação da sala, a aplicação valida se o código de acesso criado é único. Como mencionado na sessão 4.2.3, não foram incorporados alguns recursos da criação de salas, portanto, na primeira versão, o sistema conta apenas com a estrutura destinada a essas finalidades, que serão de fato implementadas futuramente.

## 5.7 Lista de Salas

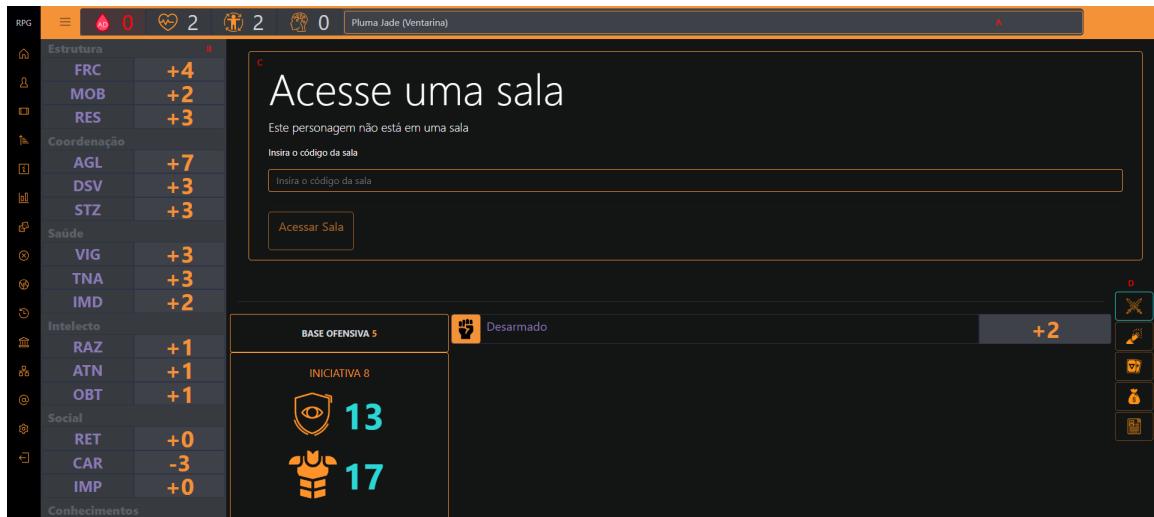
Lista de Salas		
Selecione uma sala para jogar		
Nome da Sala	Código de acesso	Selecionar
A guerra da rosa	TESTE	<button>Jogar</button>
O diamante de Evergreen	DIAMANTE	<button>Jogar</button>
Cerco ao Forte Rocha Negra	ROCHANEGRA	<button>Jogar</button>
O mistério do pantano	MISTPANTANO	<button>Jogar</button>
Neve e sangue	NEVESANGUE	<button>Jogar</button>

**Figura 5.19:** Interface de listagem de salas

Uma vez criada a sala de jogo, a mesma, tal como as demais salas que o mestre eventualmente tenha criado, será listada na interface de listagem das salas, a partir da qual o mestre pode visualizar o código de acesso, ou acessar o painel de jogo da sala selecionada.

## 5.8 Painel de jogo do Personagem

Cada jogador que criou seu personagem pode acessar o painel de jogo a partir da lista de personagens, nessa tela, ele tem acesso a todas as funcionalidades necessárias para jogar, durante todo o tempo da sessão, como demonstrado na imagem 5.20.



**Figura 5.20:** Painel do jogador

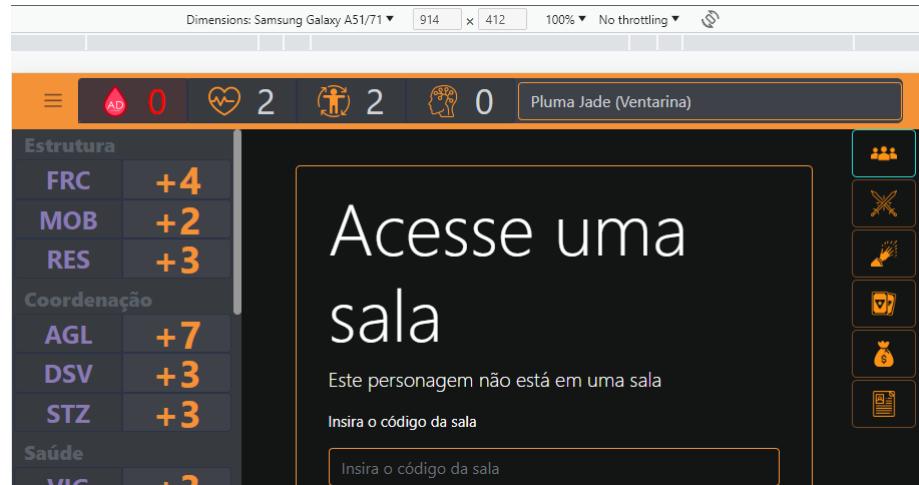
A primeira zona do painel de jogo é o cabeçalho, indicado pela legenda A, composto por 3 elementos. O primeiro elemento (fundo vermelho) exibe o valor da soma dos redutores sofridos pelo personagem, o segundo elemento (grupo de 3 botões) exibe os valores de sua integridade, aptidão física e mental. O último elemento (barra de texto) exibe o nome completo do personagem, com sobrenome e nome alternativo.

A segunda zona do painel é a coluna de capacidades e conhecimentos, indicada pela legenda B, se tratando de uma listagem vertical das 15 capacidades padrão do jogo e de todos os conhecimentos que o personagem possui, juntamente com o bônus calculado de cada capacidade e conhecimento listados.

A terceira zona do painel, indicada pela legenda C, é a “Mesa”, que exibe a interação entre os usuários que fazem parte do jogo. Por padrão, da primeira vez que o usuário abrir o painel, a mesa será apenas uma janela de texto solicitando o código de uma sala, porém, uma vez aceito, será exibida a lista dos personagens aliados e inimigos inseridos na sala.

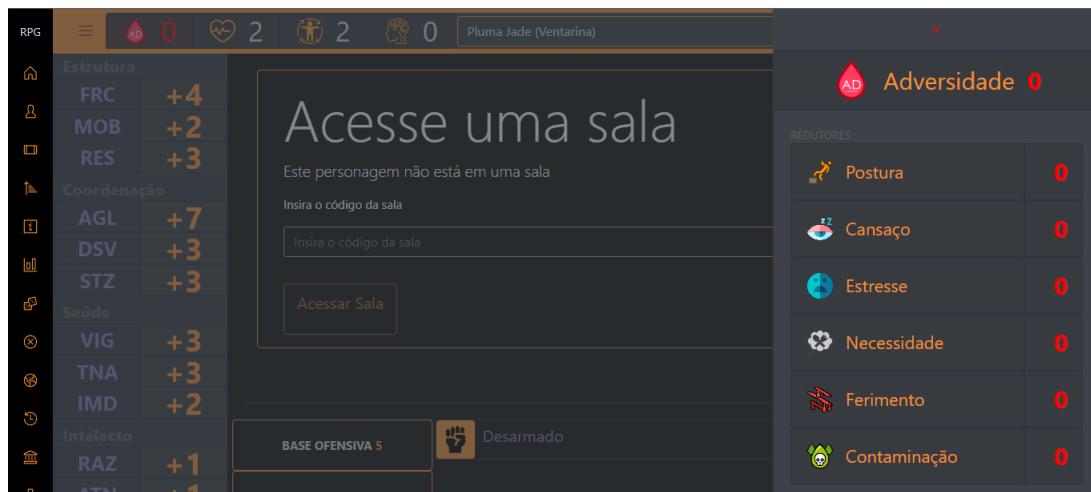
A última zona do painel é composta por várias abas de controle, como indicado pela legenda D. Essas abas correspondem, respectivamente, ao controle de combate, poder, habilidades, inventário e perfil do personagem. Na versão atual do sistema, apenas a aba de combate foi implementada, exibindo os valores calculados da guarda, pós-guarda e base ofensiva, além da lista de armas que o bônus de ataque calculado de cada uma.

Ressaltando novamente a responsividade, em dispositivos móveis a interface foi configurada para que se ajuste ao tamanho da tela em modo paisagem, como exibido na imagem 5.21, adicionando barras de rolagem verticais para viabilizar a navegação.



**Figura 5.21:** Painel do jogador em dispositivo Mobile

Também é notável que a terceira zona do painel, a Mesa, foi configurada para que seja inserida como uma das abas de controle, para que ambas as zonas não fiquem sobrepostas, economizando espaço de tela.



**Figura 5.22:** Aba lateral de redutores

Vários dos elementos que compõe as zonas do painel foram configuradas para abrir uma aba informativa na lateral direita da tela, detalhando informações relevantes. A figura 5.22 demonstra a abertura da aba lateral de redutores, aberta caso o usuário clique no primeiro botão da primeira zona do painel, cujo ícone se assemelha a uma gota de sangue. Essa aba detalha o acúmulo de cada tipo de redutor que resulta na adversidade atual do personagem.

A imagem 5.23 exibe a aba lateral aberta caso o usuário clique no segundo botão da primeira zona do painel, o grupo com os ícones e valores da integridade, aptidão física e mental. Quando aberta, a aba exibe os valores dos atributos mencionados, mas com os nomes escritos por extenso, servindo como legenda para os ícones.



**Figura 5.23:** Aba lateral de status

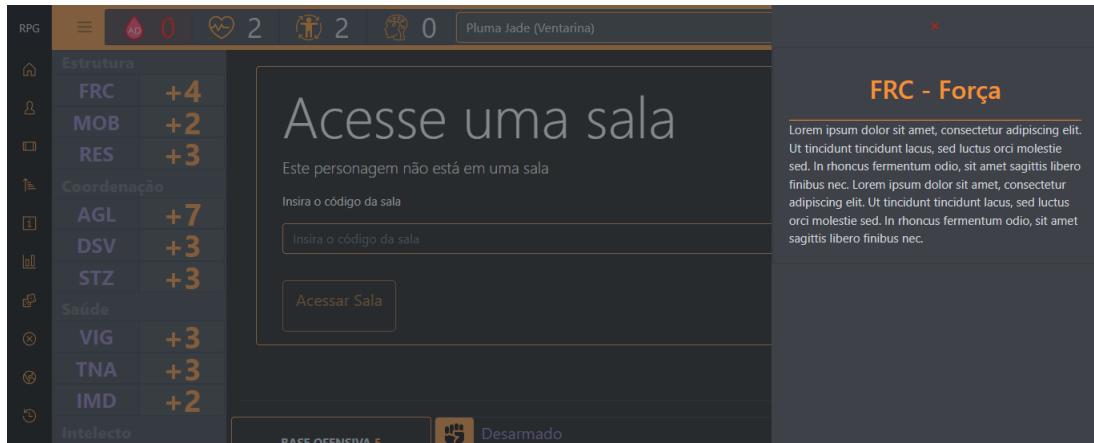
Adicionalmente, essa aba também exibe uma relação dos valores das capacidades do personagem obtidos na criação do mesmo, permitindo que o usuário veja o valor original de uma capacidade, e não apenas o valor do bônus calculado, exibido na zona de capacidades e conhecimentos.



**Figura 5.24:** Demonstração da aba de perfil

A aba exibida na imagem 5.24, aberta ao clicar no nome do personagem no cabeçalho do painel, é uma das menos desenvolvidas na primeira versão do sistema. Atualmente, ela exibe separadamente o nome, sobrenome e nome alternativo do personagem, mas em versões futuras essa aba deve permitir a troca do ícone do personagem por uma imagem da preferência do usuário, e descrições escritas pelo próprio usuário a respeito do personagem.

A zona de capacidades e conhecimentos também pode abrir abas laterais, basta que o usuário clique no nome de alguma das capacidades ou conhecimentos listados.



**Figura 5.25:** Demonstração da aba descriptiva de capacidades

A aba exibe o nome, sigla e descrição do elemento correspondente, como demonstrado na figura 5.25. Vale mencionar que, para a versão atual, este recurso foi implementado apenas visualmente.

## 5.9 Painel de jogo da Sala



**Figura 5.26:** Painel do Mestre

O mestre também deve acessar seu painel de jogo, demonstrado na figura 5.26, de onde ele poderá administrar a partida. Esse painel também pode ser dividido em zonas, e possui elementos que podem abrir abas laterais.

A zona de cabeçalho é indicada pela legenda A na figura na figura 5.26. O primeiro elemento dessa zona (ícone de fotografia) abre uma aba de galeria, como demonstra a figura 5.27. Essa funcionalidade foi implementada apenas visualmente, com imagens *placeholders*, mas espera-se que em versões futuras ela sirva para exibir imagens mapeadas de algum diretório do dispositivo, permitindo que o mestre as utilize para auxiliar na narração descritiva de cenários, itens, criaturas, etc.



**Figura 5.27:** Aba lateral da galeria

O segundo elemento do cabeçalho, com ícone de pergaminho, dá acesso a uma aba de anotações, como demonstrado na imagem 5.28. Essa aba foi implementada apenas visualmente, e seu propósito para as versões futuras é permitir que o mestre possa digitar e manter salvo um registro de anotações, para que possa consultar durante a partida.



**Figura 5.28:** Aba lateral de anotações

O terceiro elemento do cabeçalho (barra de pesquisa) serve apenas para que o mestre possa digitar uma consulta e receber “links” de acesso a outras páginas do sistema, o permitindo fazer consultas rápidas de regras, criaturas, magias, etc. Esse recurso também não é funcional na primeira versão do sistema, uma vez que as telas de conteúdo escrito não foram implementadas.

É importante ressaltar que, conforme mencionado algumas vezes no decorrer da sessão 4.2, algumas funcionalidades não seriam implementadas na primeira versão da aplicação, porém, as interfaces referentes a tais funcionalidades podem, dentro do possível, ser mapeadas visualmente, para que sejam de implementadas de maneira funcional nas versões seguintes da aplicação.

O quarto elemento do cabeçalho (escrito NPC, na figura 5.26), é referente a inserção de personagens do mestre na mesa, um recurso também não implementado de maneira funcional para a primeira versão. A proposta dessa aba é permitir que o mestre digite o nome do personagem que deseja inserir na sala (geralmente inimigos que serão combatidos pelos jogadores) para que ele seja listado e selecionado, conforme demonstrado a figura 5.29.



**Figura 5.29:** Aba lateral de inserção de personagens do mestre

O quinto elemento do cabeçalho (ícone de um dado de 20 lados na figura 5.26) é referente a aba de solicitação de testes, como demonstrado pela imagem 5.30. Nessa aba, o mestre define o nível de dificuldade do teste, o redutor que será testado, a capacidade utilizada na tentativa de resistência, e os personagens que serão alvos do teste.



**Figura 5.30:** Aba lateral de inserção de personagens do mestre

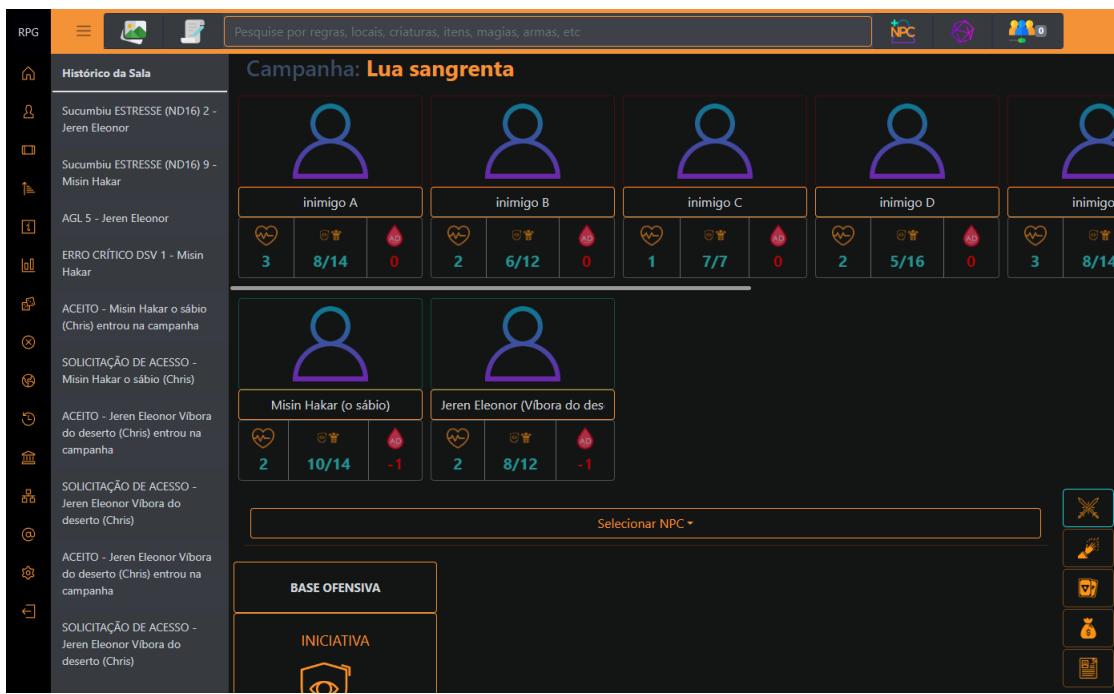
A funcionalidade de solicitação de testes foi implementada, sendo totalmente funcional nessa primeira versão do sistema. Ela será abordada em detalhes na sessão 5.10.3, uma vez que envolve o uso da conexão SSE entre cliente e servidor, tema da sessão ??, que ainda será abordada. Pode-se adiantar que a funcionalidade de gerenciamento de acessos, abordada no próximo tópico, compartilha da mesma situação.

O sexto elemento (último ícone no cabeçalho da tela exibida na figura 5.26) é o menu de gerenciamento de acessos dos personagens, exibido na imagem 5.31. Manipulando essa aba, o mestre pode aceitar ou negar as solicitações de acesso, verificar quais os personagens adicionados na sala, e visualizar o código de acesso da sala.



**Figura 5.31:** Aba lateral de inserção de personagens do mestre

Abordados todos os elementos oriundos da zona de cabeçalho do painel do mestre é possível descrever os demais elementos dessa interface. O elemento demarcado pela legenda **B** na referida imagem 5.26, nomeada “Histórico da Sala”, apresenta uma coluna de logs que exibe ao mestre as últimas 50 atualizações que ocorreram na sala.



**Figura 5.32:** Painel do mestre com histórico da sala preenchido.

A imagem 5.32 exibe o painel do Mestre com o “Histórico da Sala” preenchido, sendo possível verificar as informações recebidas. Esse histórico permite ao mestre acompanhar os eventos que ocorrem na sala, para que possa incorporá-los em sua narração.

O elemento demarcado pela legenda **C** na referida imagem 5.26 é o mesmo elemento “Mesa” presente no painel do jogador e serve a mesma finalidade. A diferença é que, na sala, esse elemento sempre exibe as listas de personagens dos jogadores e do mestre, inseridos na sala, diferentemente do painel de jogador, onde a mesa também possui a funcionalidade de inserção do código de acesso da sala.

Por fim, o elemento demarcado pela legenda **D** na referida imagem 5.26 é o mesmo grupo de abas de controle presente no painel do jogador, viabilizando o acesso a aba de combate, poder, habilidades, inventário e perfil de personagem. Entretanto, o mestre tem acesso a vários personagens em uma mesma sessão de jogo, diferentemente do jogador, logo, ele tem a opção de selecionar o personagem ao qual as abas de controle estarão vinculadas.

## 5.10 Interações em tempo real

É notável que existem várias interações entre usuários distintos através dos painéis de jogo, afinal, um jogador pode, por exemplo, solicitar acesso a uma sala, e o mestre é notificado da solicitação imediatamente. Ou então, ao realizar um teste de capacidade, tanto o mestre quanto o jogador devem ser notificados.

Para criar esse tipo de comportamento foi utilizada a tecnologia de comunicação assíncrona “*Server-Sent Events*”, detalhadamente abordada na sessão 4.6.1.7. Em suma, esta tecnologia permite ao servidor enviar eventos aos clientes registrados como “emitters”, permitindo que os mesmos sejam notificados sempre que pertinente, para que atualizem os dados da sala ou personagem automaticamente.

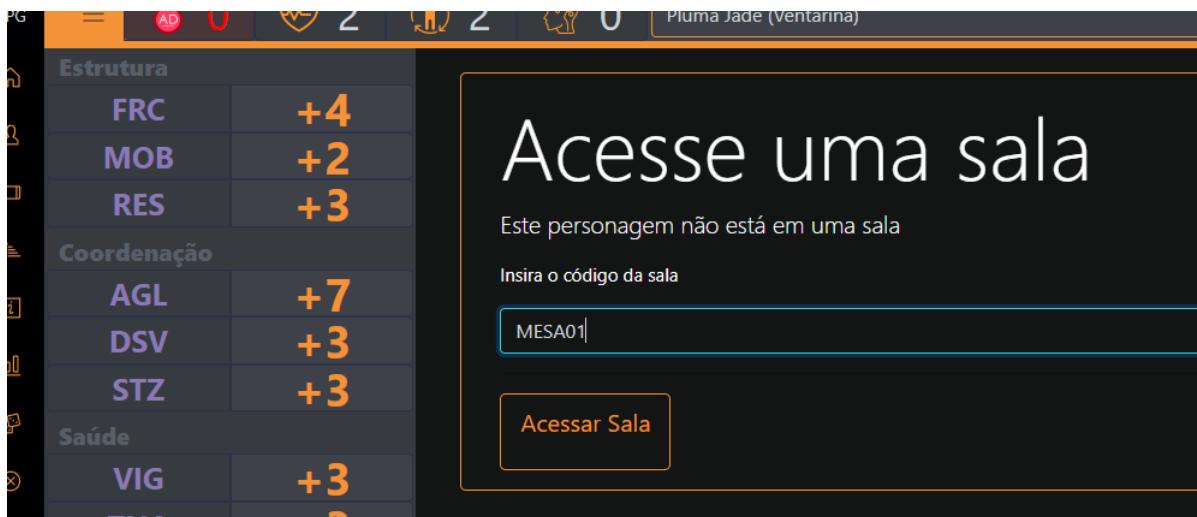
Com base nesse mecanismo foi possível implementar várias funcionalidade que se beneficiam desse tipo de interação. Portanto, no decorrer desta sessão serão descritas as funcionalidades do sistema que foram implementadas com base na comunicação em tempo real proporcionada pelo SSE.

### 5.10.1 Gerenciamento de solicitações de acesso

Começando pela solicitação de acesso a uma sala de jogo, temos que, para solicitar acesso a uma sala o jogador deve, primeiramente, saber qual o código de acesso da referida sala. Para isso, o mestre, que criou a sala, deve compartilhar o código com seus jogadores, para que eles possam solicitar o acesso.

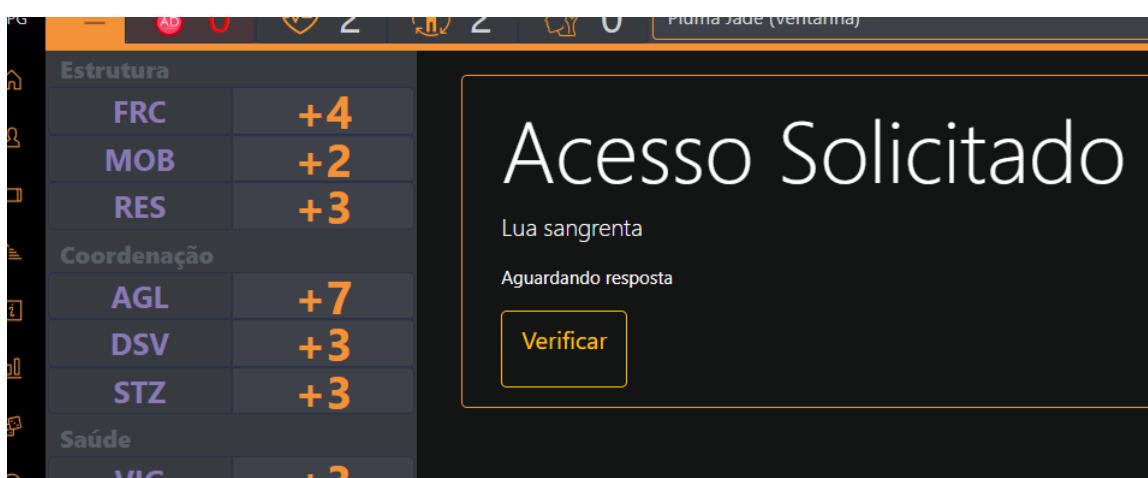
Como já mencionado, é necessário que os participantes mantenham um canal de comunicação ativo entre si, como um aplicativo de troca de mensagem, redes sociais, encontro presencial, video chamadas, etc. Dessa forma, eles podem se comunicar antes e durante as sessões de jogo.

Conhecendo o código da sala, o jogador pode se dirigir ao painel do personagem selecionado e solicitar o acesso. Como demonstrado na figura 5.33, o painel do personagem, na zona “Mesa”, conta com um campo onde o usuário insere o código da sala que deseja acessar.



**Figura 5.33:** Demonstração de uma solicitação de acesso.

Uma vez que o código informado é verificado e a solicitação é registrada, o sistema exibe uma confirmação da solicitação, informando inclusive o nome da sala encontrada, como exibido na imagem 5.34.



**Figura 5.34:** Solicitação de acesso em espera.

A solicitação de acesso parte do dispositivo de um jogador sendo processada pelo servidor. Uma vez registrada a solicitação, ela passa a constar na lista de solicitações da sala. Entretanto, o dispositivo do mestre ainda exibe os dados anteriores a esse processamento, quando a lista de solicitações estava vazia. Portanto, a menos que o mestre atualize os dados em seu dispositivo, ele não será capaz de visualizar a notificação de acesso recebida.

É nesse momento que o mecanismo de comunicação SSE cumpre sua função. O método que processa e registra a solicitação de acesso chama pelo “updateSala”, demonstrado na figura 4.14. Logo, todos os dispositivos clientes registrados como “emitters” da sala em questão recebem um evento que dispara a atualização automática dos dados.

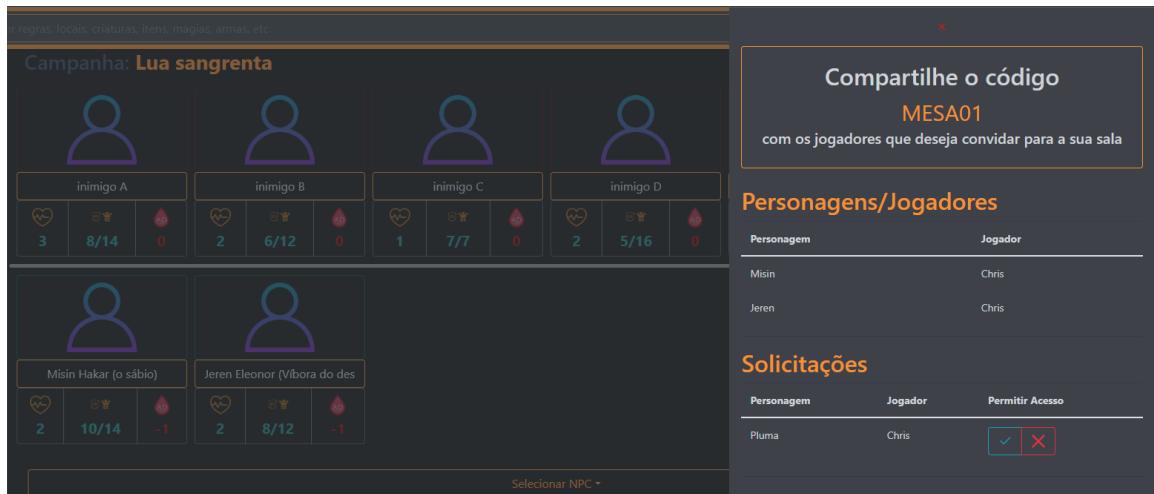


Figura 5.35: Solicitação de acesso recebida.

Dessa forma, os dados atualizados da sala são renderizados no dispositivo do mestre, exibindo automaticamente a nova solicitação de acesso no painel, como consta na imagem 5.35. A partir disso, cabe ao mestre aprovar ou negar o acesso do personagem/jogador solicitante.



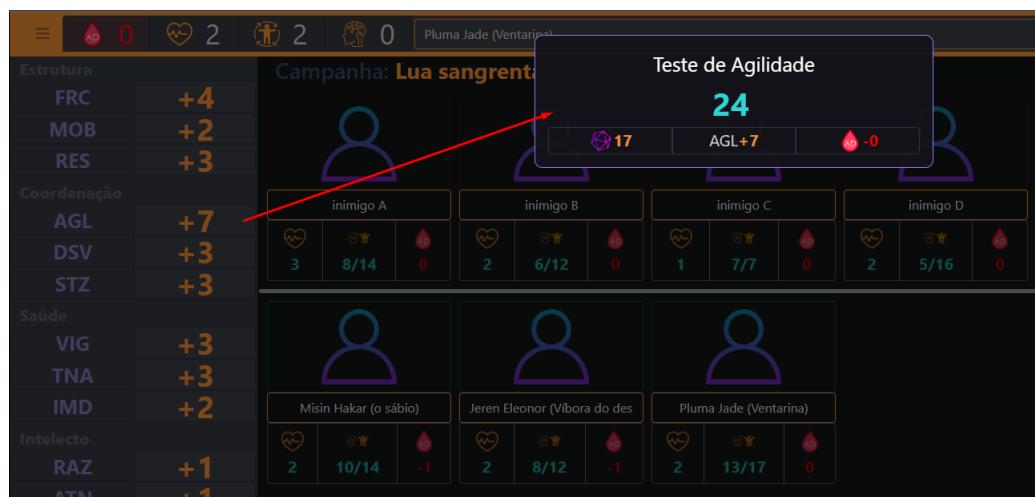
Figura 5.36: Novo personagem adicionado.

O personagem adicionado passa a constar na zona “Mesa”, que agora exibe os três integrantes atuais da partida, conforme consta na imagem 5.36. Como já mencionado, a “Mesa” aparece tanto para o mestre, quanto para os jogadores. Outro detalhe, é que a aba “Histórico da Sala” exibe mensagens notificando a chegada da solicitação e seu aceite.

## 5.10.2 Realização de testes Capacidade

Outra funcionalidade que depende do mecanismo de comunicação SSE é a realização de testes. Durante o jogo, os testes são utilizados para definir o resultado de eventos narrativos, como explicado na sessão ???. O mestre define uma dificuldade, e se o valor do teste se iguala ou supera o nível de dificuldade estabelecido, o teste foi bem-sucedido, caso contrário, o teste resulta em falha.

Os testes de capacidades são os mais corriqueiros, através de um teste da capacidade Mobilidade (MOB) é possível determinar se um personagem consegue saltar determinada distância, com um teste de Agilidade (AGL) pode-se determinar se um personagem foi rápido o suficiente para agarrar um item antes que ele caia, um teste de Retórica (RET) pode servir para negociar uma informação, e assim por diante.



**Figura 5.37:** Exemplo de teste de capacidade.

A imagem 5.37 ilustra a realização de um teste da capacidade “Agilidade”. Basicamente, o método que realiza o teste é disparado após um “clique” no botão correspondente (de onde parte a seta em vermelho). O sistema gera um valor aleatório que representa a rolagem de dados, realiza os cálculos e retorna os resultados, seguindo o processo descrito no caso de uso 4.3.9.

Embora o nível de dificuldade do teste seja definido pelo mestre, o sistema não exige que esse valor seja informado, portanto, a aplicação não se encarrega de comparar o resultado do teste com a dificuldade alvo, uma vez que isso serviria apenas para exibir se o teste foi bem-sucedido ou falho.

No caso de testes de capacidade o resultado não afeta os dados do personagem ou da sala, uma vez que o seu resultado gera uma consequência totalmente narrativa. Portanto, basta que o mestre decida o valor da dificuldade e peça para que o jogador realize o teste, uma vez dado o resultado, o mestre saberá se correu sucesso ou falha e poderá narrar a cena correspondente.

No que se refere ao mecanismo de comunicação SSE, ele se aplica, neste caso, apenas no que se refere ao controle do mestre em relação à realização de testes por parte dos jogadores, uma vez que o mestre precisa saber o resultado de todos os testes realizados.

```

190     String prefix = "";
191     if(d20 == 1) {
192         prefix = "ERRO CRÍTICO ";
193     }
194     if(d20 == 20) {
195         prefix = "CRÍTICO ";
196     }
197
198     SalaLog log = new SalaLog(prefix + tipo + " " + result + " - " + personagemCarregado.getNome() + " " + personagemCar
199     try {
200         salaLogRepository.save(log);
201     } catch (Exception e) {
202         throw new BadRequestException("Erro ao registrar operação");
203     }
204
205     SSEController sse = new SSEController();
206     sse.updateSala(idSala);
207
208     return ResponseEntity.ok(testeCapacidade);
209
210 }
```

**Figura 5.38:** Trecho do ControllerTeste.

Portanto, sempre que um jogador realiza um teste de capacidade, o servidor dispara o método “updateSala”, como demonstrado na imagem 5.38, linha 206. Dessa forma, o painel da sala será atualizado imediatamente após o registro do teste, permitindo que o mestre seja notificado do resultado quase que instantaneamente.

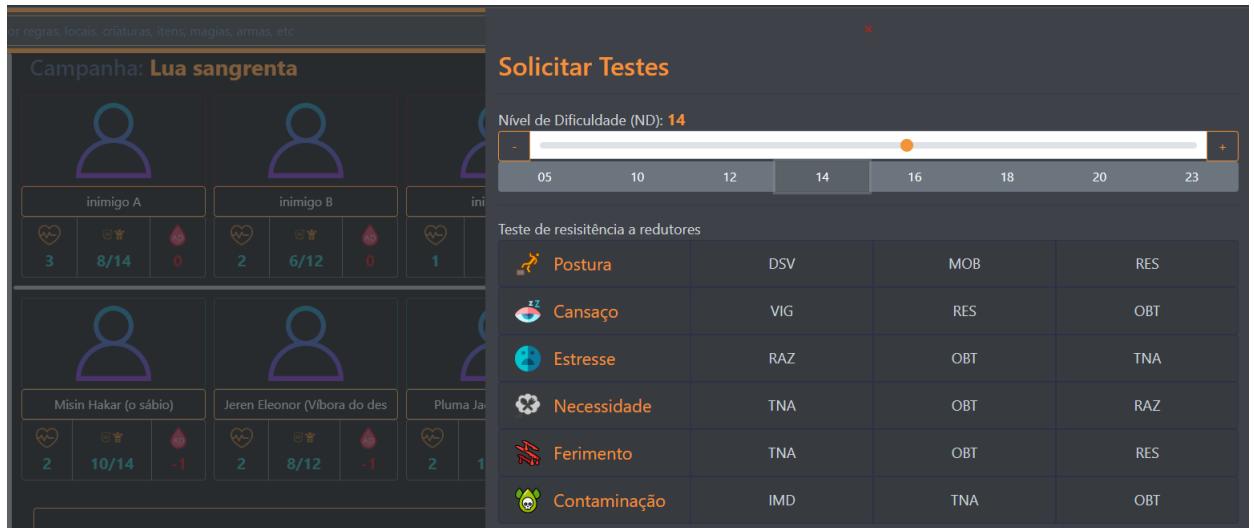


**Figura 5.39:** Registro do teste recebido.

A imagem 5.39 ilustra o painel de jogo do mestre, onde, no “Histórico da Sala” foi exibido o teste de capacidade realizado pelo jogador, incluindo o tipo de capacidade testada (AGL), seu valor (24) e o nome do personagem que realizou o teste (Pluma Jade).

### 5.10.3 Solicitação de testes de Resistência

O sistema também permite a realização de testes de resistência a redutores, como descrito na sessão 4.3.10. Esse tipo de teste é realizado quando os personagens dos jogadores estão sujeitos, dentro da narrativa, a alguma condição negativa, expressa através dos redutores, como “cansaço”, “estresse”, “contaminação”, etc.



**Figura 5.40:** Interface de solicitação de teste.

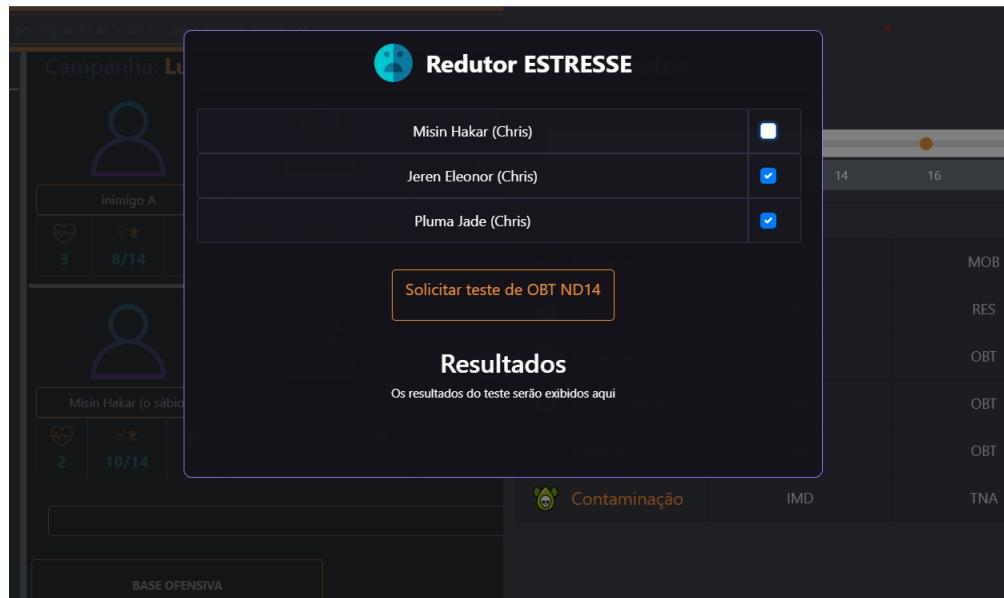
É o mestre quem solicita testes de resistência, como visível na imagem 5.40, a interface o permite manipular uma barra horizontal para definir o valor da dificuldade, e dispõe botões com valores predefinidos para agilizar o processo. Usar esse componente é mais amigável e ágil do que abrir um teclado numérico para digitação do valor, principalmente em dispositivos móveis.

Diferente de um teste de capacidade, o teste de resistência acarreta alterações em dados importantes do personagem, motivo pelo qual a dificuldade do teste deve ser informada pelo mestre antes que o teste ocorra. Dessa forma, o próprio sistema pode comparar o resultado do teste com a dificuldade estabelecida, e realizar automaticamente as tratativas decorrentes do sucesso ou falha do teste.

Em seguida, o mestre deve definir para qual dos 6 redutores o teste deve ser realizado, o que depende totalmente da ocasião narrativa que origina o teste. Por exemplo, um teste de resistência ao redutor “Estresse” pode ser pertinente caso os personagens estejam enfrentando uma criatura intimidadora que possa amedrontá-los.

Para cada redutor a interface dá três opções de capacidades que podem ser usadas no teste de resistência, algo que também depende totalmente das circunstâncias narrativas do teste. Ou seja, o estresse de adentrar um local escuro e inóspito pode ser combatido pela capacidade “Obstinação (OBT)” do personagem, que representa sua coragem, ou pela capacidade “Razão (RAZ)”, caso o personagem se apegue a racionalidade no momento de tensão para se tranquilizar.

Ou seja, para realizar uma solicitação de teste o mestre manipula o componente numérico para definir o nível de dificuldade do teste, depois ele escolhe a linha da tabela de opções que corresponde ao redutor que será aplicado, e por fim, seleciona o botão correspondente a sigla da capacidade usada na tentativa de resistência.



**Figura 5.41:** Confirmação da solicitação de teste.

A imagem 5.41 exibe a janela aberta no momento em que o mestre realiza um “clique” em uma das opções, abrindo a janela de confirmação da solicitação de teste. Na tela que se abre, o mestre verifica os dados do teste e define quais personagens serão testados, inclusive, é notável que apenas dois dos três personagens da sala estarão sujeitos ao teste exibido no exemplo.



**Figura 5.42:** Resultado da solicitação de teste.

Ao selecionar a opção “Solicitar teste” o método correspondente é disparado no servidor. Nele, é realizado um teste individual para cada personagem alvo, com rolagem de dados e cálculos distintos. Os resultados são exibidos na tela do mestre, e no “Histórico da Sala”, como demonstrado na imagem 5.42.

Caso o personagem falhe no teste o sistema deve registrar o redutor sofrido e disparar o método “sub()”, demonstrado na imagem ??, que atualiza os dados da sala e dos personagens nos dispositivos “clientes”. Dessa forma, o painel do Personagem será atualizado, permitindo que o jogador veja o valor atualizado da adversidade acumulada.

Observando o exemplo, na imagem 5.42, é possível notar que o personagem “Jeren” falhou no teste, uma vez que não atingiu o valor da dificuldade do mesmo. Narrativamente, ele sucumbiu ao estresse.

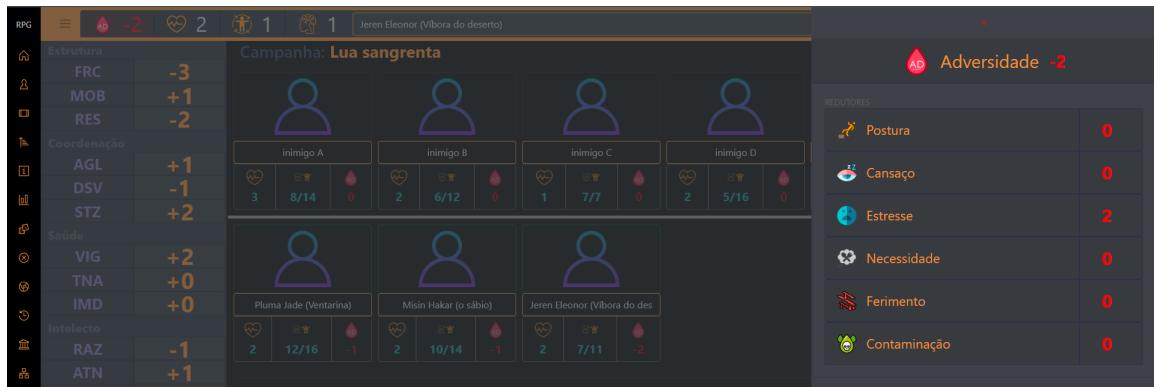


Figura 5.43: Adversidade acumulada.

A imagem 5.43 demonstra a adversidade acumulada pelo personagem citado, exibida no dispositivo do jogador após a atualização automática dos dados, decorrente da conexão SSE. O personagem em questão já possuía um ponto no redutor de “Estresse” devido a testes anteriores, portanto, mediante o novo teste, o personagem acumulou dois pontos nesse redutor.

A “Adversidade” também é elevada automaticamente, uma vez que este valor é calculado a partir do acúmulo de todos os redutores. Portanto, o sistema se encarrega de registrar o redutor e acumular a adversidade, um valor que será aplicado em todos os cálculos, para qualquer teste realizado.

A aplicação da adversidade nos cálculos é sempre negativa, motivo pelo qual seu valor é exibido como negativo. Esse comportamento foi descrito nas regras do jogo, especificamente na sessão ??, e é garantido pela aplicação, de forma que, quanto mais redutores sofridos, maior o efeito adverso, o que reduz cada vez mais as chances de sucesso do personagem para realizar ações.

## **6 CONCLUSÃO**

Neste trabalho, foi elaborado, planejado e implementado um sistema de informação, cujo principal objetivo é atestar a capacidade de desenvolvedor do autor, atuando no projeto e desenvolvimento de software, tal como no atendimento de requisitos propostos, apresentando soluções para os problemas estabelecidos.

Vale ressaltar que o projeto é, desde sua concepção, consciente da limitação de prazo e do volume de funcionalidades que precisariam ser desenvolvidas para que a totalidade do jogo fosse implementada. Motivo pelo qual, o escopo definido para o trabalho abrange uma parcela das regras do “Mysthen RPG”, e nesse sentido, o escopo definido foi implementado em sua totalidade.

Pode-se apontar que o projeto do sistema aplica vários dos principais conceitos de planejamento e engenharia de software, como o uso de boas práticas, modelagem do banco, normalização e produção de artefatos, como os levantamentos de requisitos, casos de uso, etc.

Nesse mesmo sentido, é possível afirmar, quanto a implementação em si, que foi desenvolvido um sistema web funcional, multiplataforma e responsivo, gerenciado via metologia ágil, com métodos de autenticação aplicados, trabalhando com comunicação assíncrona, apresentando interfaces complexas entre outros recursos. Além de ter sido desenvolvido a partir da combinação de diversas ferramentas em voga no mercado.

Dessa forma, os objetivos específicos definidos no trabalho foram concluídos como planejado, e no que se refere ao objetivo geral, relacionado a funcionalidade do sistema como gerenciador de um jogo de RPG, mediante as problemáticas levantadas, tem-se que a aplicação cumpre sua função conforme esperado.

No que se refere, por exemplo, a necessidade de interrupção do fluxo narrativo para realização de consultas e cálculos pelos jogadores, pode-se concluir que a aplicação consegue otimizar o processo conforme esperado, realizando os cálculos automaticamente, retornando resultados praticamente instantâneos e permitindo o acesso a qualquer funcionalidade do jogo a poucos cliques de distância.

Em relação à quantidade de regras a serem memorizadas pelos jogadores, a aplicação se mostra capaz de reduzir esse volume significativamente. O usuário ainda precisa saber como o jogo funciona, porém, a complexidade se resume a saber para onde navegar e qual opção selecionar, sem a necessidade de saber qual cálculo deve ser realizado em cada situação, e quais os elementos que o compõe.

Já a criação de personagens na aplicação se resume a navegar pelas telas analisando e selecionando opções, o que é consideravelmente mais simples e prático do que fazê-lo anotando relações de valores de capacidade, calculando e relacionando atributos manualmente e coordenando pessoalmente o processo de criação, tarefas essas totalmente coordenadas pelo sistema.

Naturalmente, existem desafios e vulnerabilidades a serem apontados e corrigidos. Como já mencionado, o projeto foi concebido consciente de que o jogo não poderia ser implementado em sua totalidade. Porém, recursos como o gerenciamento de itens, a realização de testes de combate e a implementação no modelo PWA, eram cotados para constar no trabalho, mas tiveram que ser removidos do escopo, unicamente por limitação de prazo.

Considerando a continuidade do desenvolvimento da aplicação, esses e outros recursos poderiam ser implementados sem grandes dificuldades em trabalhos futuros, afinal, nenhuma dessas funcionalidades apresenta um grau de complexidade consideravelmente maior do que os recursos já inseridos.

A única limitação técnica realmente considerável é referente ao sistema de comunicação assíncrona. Originalmente, foi estudada a possibilidade de implementação desse mecanismo via “WebSockets”, mas devido aos excessivos problemas e erros, essa tecnologia foi descartada, dando lugar aos “Server-Send Events”. Embora funcional, esse mecanismo de comunicação ainda poderia ser melhor trabalhado, no que se refere a otimização e estabilidade, algo que também poderia ser facilmente implementado em versões futuras.

Portanto, conclui-se que, dentro do que foi definido, a aplicação desenvolvida obedece à premissa do trabalho e atende aos objetivos estabelecidos, uma vez que atua de fato como gerenciador virtual de um jogo de RPG. Além disso, analisando o software implementado, pode-se atestar as capacidades do autor como um desenvolvedor de software.

# Referências Bibliográficas

- ARENA4G. **Encontre sua aventura através de jogos de mesa.** 2017. <https://arena4g.com/encontre-sua-aventura-atraves-de-jogos-de-mesa/>. 30
- AZEVEDO, Nill Chesther Nunes. Uso do role playing game (rpg) no ensino de geografia. 2019. 24, 27, 37
- BAELDUNG. **Streaming Spring MVC e processamento de solicitação SSE.** 2022. Acesso em: Junho de 2023. Disponível em: <https://www.baeldung.com/spring-mvc-sse-streams>. 77
- BINI, Marco. **Como construir sua ficha de personagem em Cyberpunk 2020.** 2020. Disponível em: <https://nuckturm.com.br/como-construir-sua-ficha-de-personagem-em-cyberpunk-2020/>. 29
- BOAGLIO, Fernando. **Spring Boot: Acelere o desenvolvimento de microsserviços.** [S.l.]: Casa do Código, 2017. 73
- BONNEY, Meg. **Watch Terry Crews play DD on Nerdist's 'CelebriDD' series!** 2018. <https://www.purefandom.com/2018/08/02/watch-terry-crews-play-dd-nerdist-celebriidd-series/>. 22
- BOOCH, Grady. **UML: guia do usuário.** [S.l.]: Elsevier Brasil, 2006. 52
- BUTLER, Nathan. **Star Wars: Force and Destiny: Core Rulebook Overview.** 2015. <https://www.youtube.com/watch?v=MONuCQIouNw>. 25
- COCKBURN, Alistair. **Escrevendo Casos de Usos Eficazes: Um guia prático para desenvolvedores de software.** [S.l.]: Bookman Editora, 2005. 51
- CREPALDI, Roselene. Jogos, brinquedos e brincadeiras. **Curitiba: IESDE Brasil SA,** 2010. 16
- CUSTÓDIO, José Antônio Loures; SOUZA, AP DE. O ethos religioso na antiguidade: A origem ritualística dos jogos de tabuleiro. **Revista Científica/FAP**, v. 20, n. 1, 2019. 16

DEVMEDIA. **Introdução ao Maven.** 2012. Disponível em: <https://www.devmedia.com.br/introducao-ao-maven/25128/>. 73

DIAS, Ana Luiza; FORTES, Renata PM; MASIERO, Paulo César; GOULARTE, Rудиней. Uma revisão sistemática sobre a inserção de acessibilidade nas fases de desenvolvimento da engenharia de software em sistemas web. In **IHC**, p. 39–48, 2010. 44

ELMASRI, Ramez; NAVATHE, Shamkant B. **Fundamentals of Database Systems.** 7th. ed. [S.l.]: Pearson, 2016. 64, 71

FREITAS FELIPE DELLA CORTE, Guilherme Dei Svaldi J.M. Trevisan Leonel Caldela Marcelo Cassaro Rafael Dei Svaldi e Rogerio Saladino Álvaro. **Tormenta 20.** [S.l.]: Jambô, 2020. 31, 32

GONZAGA, Matheus. **Como narrar em um evento de RPG.** 2020. Disponível em: <https://blog.jamboeditora.com.br/como-narrar-em-evento-de-rpg/>. 37

MACRAE, Callum. **Vue.js - Up and Running.** [S.l.]: O'Reilly Media, 2018. 80

MAGO, Conselho do. **TUDO QUE VOCÊ PRECISA SABER SOBRE DADOS DE RPG.** 2020. Disponível em: <https://conselhodomago.com.br/dados-rpg/>. 30

MARQUES, Claudio Faria. Cyberpunk, feitiçaria e paranóia. 2003. 25

MEARLS, Jeremy Crawford Mike. **D&D Quinta edição: Regras Básicas para Jogadores Versão 0.3.3.** [S.l.]: Wizards of the Coast, 2014. 23, 24, 25, 26, 27, 31, 32, 33, 34, 36, 37

MONTANHEIRO, Lucas Souza; CARVALHO, Ana Maria Martins; RODRIGUES, Jackson Alves. Utilização de json web token na autenticação de usuários em apis rest. **XIII Encontro Anual de Computação. Anais do XIII Encontro Anual de Computação EnAComp**, p. 186–193, 2017. 74

MTELLES. **Numenera RPG (resenha).** 2015. <https://www.rederpg.com.br/2015/09/18/numenera-rpg-resenha/>. 18

\_\_\_\_\_. **A Importância da Tradução de Obras de RPG.** 2019. Disponível em: <https://www.rederpg.com.br/2019/01/24/importancia-da-traducao-de-obras-de-rpg/>. 28

OLIVEIRA, Rafael Camargo de. Tipos de narrador. 2018. 23

PEINADO, JURANDIR; AGUIAR, G. Compreendendo o kanban: um ensino interativo ilustrado. **Revista Da Vinci. Curitiba-PR**, v. 4, n. 1, p. 133–146, 2007. 62

PEREIRA, Farley Eduardo Lamines. No limite da ficção: comparações entre literatura e rpg-role playing games. Universidade Federal de Minas Gerais, 2007. 22

PETERSEN, Sandy; WILLIS, Lynn; MASON, Mike. **Call of Cthulhu Rpg Keeper Rulebook: Horror Roleplaying in the Worlds of.** [S.l.: s.n.], 2015. 25

PRESSMAN, Roger S. **Software Engineering: A Practitioner's Approach.** [S.l.: s.n.], 2015. 48, 72, 76

PRIETTO, Thiago. Literatura e os jogos de rpg: Trajetória de apropriações e intertextos. **Translatio**, n. 6, p. 8, 2013. 16, 17

RANGEL, Taiguara. **D&D Nova versão de Dungeons & Dragons é anunciada para 2024.** 2021. <https://mesaderpg.com.br/dungeons-dragons/nova-versao-de-dungeons-dragons-2024/>. 17

RAPHAEL, Pablo. **Horror cósmico: Conheça os RPGs inspirados nos contos de H.P. Lovecraft.** 2020. <https://www.theenemy.com.br/pc/horror-cosmico-hp-lovecraft-chamado-de-cthulhu-rastro-de-cthulhu-arkham-horror#:~:text=A%20primeira%20ediÃ§Ã£o%20de%20Call,RPGs%20mais%20populares%20do%20mundo>. 18

RPGPLANET. **A Mina Perdida de Phandelver (Parte 2).** 2020. <https://www.youtube.com/watch?v=z7FULZE0cGs&list=PL42CWo39PLP8VJDpJeVSz5QL712XtOHGN&index=2/>. 33

SLAVICSEK ANDY COLLINS, JD Wiker Bill. **Star Wars RPG Edição Saga - Livro Básico Revisado.** [S.l.]: Wizards of the Coast, 2008. 23, 31, 33, 35, 36, 37

SOARES, Michel dos Santos. Comparação entre metodologias ágeis e tradicionais para o desenvolvimento de software. **INFOCOMP Journal of Computer Science**, v. 3, n. 2, p. 8–13, 2004. 62

SOMMERVILLE, Ian. **Software Engineering.** 9. ed. [S.l.]: Addison-Wesley, 2010. 44, 48, 72, 76

TANENBAUM, Andrew S.; STEEN, Maarten Van. **Distributed Systems: Principles and Paradigms.** [S.l.]: Pearson, 2009. 71, 80

# **Apêndice A – Simulação de uma partida do “Mysthen RPG”**

Segue a descrição de uma partida de RPG simulada para este trabalho, ilustrando a aplicação das regras do “Mysthen RPG”, conforme descrito no capítulo 3. Será demonstrado o funcionamento do jogo que deve ser viabilizado pela aplicação, facilitando a compreensão a respeito da proposta explorada.

Esta sessão também cumpre o papel de evidenciar que o “Mysthen RPG”, embora seja um jogo autoral, se trata, de fato, de um autêntico RPG de mesa tradicional, que opera pelos conceitos descritos e referenciados no capítulo 2. Inclusive, a subdivisão dos assuntos abordados objetiva traçar um paralelo explícito com as sessões 2.3 e 2.4.

Também é importante ponderar que esta simulação considera a totalidade das regras do “Mysthen RPG”, demonstrando que o jogo se trata de um RPG completo e funcional. Adicionalmente, dando essa visão geral sobre a totalidade das regras vislumbra-se a alta quantidade das mesmas, evidenciando e justificado porque apenas um recorte dessas regras foi selecionado para a implementação.

Entretanto, o fato de abordar a totalidade das regras faz com que esta simulação comente sobre mecânicas que não foram descritas no capítulo 3, e consequentemente, não foram inclusas no software desenvolvido. Como mencionado inúmeras vezes no trabalho, a totalidade das regras pode ser consultada na página [www.notion.so/rpgleethak/Mysthen-RPG](http://www.notion.so/rpgleethak/Mysthen-RPG) e almeja-se que todo este conteúdo seja adaptado para a aplicação em trabalhos futuros.

## **A.1 Obtenção do jogo e participantes na simulação**

Como mencionado, o jogo abordado na simulação será o mesmo adaptado na aplicação proposta, o “Mysthen RPG”, que ocorre no cenário fictício “Leethak”, e regras pelo sistema “SIR”. Quanto aos participantes, considera-se um mestre e três jogadores, referenciados como “mestre”, “jogador 01”, “jogador 02” e “jogador 03”.

## A.2 Planejamento da aventura simulada

Como comentado na sessão 2.3.4, o planejamento consiste na elaboração de um plano de fundo para a história dentro do cenário proposto pelo jogo. O cenário “Leethak” é de fantasia medieval, portanto, na simulação, o mestre elabora um plano de fundo com os seguintes elementos:

- 1.Existe um livro lendário contendo conhecimentos mágicos proibidos.
- 2.O livro está escondido nas ruínas de um templo perdido em um pântano.
- 3.Um mago poderoso busca pelo livro
- 4.O mago descobre a região onde o livro pode ser encontrado.
- 5.O mago convence o rei da região a financiar sua busca pelo livro.
- 6.O rei possui uma doença incurável
- 7.O rei é convencido de que o mago poderá curá-lo usando o poder do livro.
- 8.O mago, por sua vez, pretende tomar o poder para si e dominar o reino.
- 9.O mago possui uma aprendiz que é cúmplice em sua conspiração.
- 10.Juntos eles investigam e descobrem o paradeiro do livro.
- 11.Em suas pesquisas, a aprendiz descobre que o livro possui uma maldição.
- 12.A aprendiz omite a informação, deixando que o mago ative a maldição.
- 13.Ao tocar o livro, o mago é paralisado pela maldição.
- 14.Como sua última barreira de proteção, o livro desperta criaturas guardiãs.
- 15.O mago permanece vivo, mas paralisado dentro do templo.
- 16.O rei deseja reaver o livro e o mago.

A partir desse plano de fundo o mestre pode criar algumas problemáticas com os quais os personagens dos jogadores poderão interagir. Por exemplo, o rei precisa reaver o livro, mas não pode envolver seus soldados ou os cavaleiros sagrados da igreja nessa questão, uma vez que a natureza vil do livro deve ser mantida em sigilo. Portanto, ele decide anunciar um torneio, e aos campeões ele revelaria seus planos, os contratando para investigar e resolver a situação.

Outro desdobramento, seria dizer que o despertar das criaturas guardiãs do livro afugenta alguns seres que já habitavam os pântanos, onde o templo é localizado. Dessa forma, esses seres acabam se aproximando de vilarejos, cidades e estradas, ocorrendo ataques que causam uma crise de segurança no reino.

Usando sua criatividade, o mestre pode trabalhar novos elementos que integram essa situação, como intervenções da aprendiz do mago, a participação da igreja e dos cavaleiros sagrados, um possível reino vizinho que poderia se aproveitar dessa instabilidade, etc. Dessa forma, o mestre cria locais, adversários, aliados, criaturas, cenas, situações, entre outros elementos narrativos.

O importante é criar todo um panorama de problemas em aberto antes do jogo de fato ocorrer, dispondo os elementos narrativos com os quais os personagens dos jogadores poderão interagir ou não. Quando o jogo de fato se iniciar, como já comentado, tais elementos serão apresentados gradativamente, a depender do curso da trama, que embora conduzido pelo mestre, depende das ações e objetivos dos próprios personagens, mediante as escolhas dos jogadores.

Essa etapa é evidentemente criativa, e praticamente não se envolve com regras, apenas com o cenário. Portanto, cabe ao material do jogo apresentar opções de criaturas, civilizações, reinos, locais, entre outros elementos que o mestre pode aplicar no plano de fundo construído no planejamento da aventura. Este processo pode levar dias ou meses de preparação, a depender do nível de detalhe do plano de fundo que o mestre está elaborando.

### A.3 Criação de personagens

Os jogadores criam seus personagens seguindo as etapas da criação de personagem, descritas na sessão 3.3. Como já demonstrado, a cada etapa o jogo dispõe de opções para seleção, cabendo aos jogadores anotar as informações referentes a cada escolha. Na última etapa do processo os pontos são manualmente acumulados e calculados, gerando os valores finais que representam o personagem.

É comum que o mestre compartilhe informações sobre o plano de fundo da aventura com os jogadores antes que eles criem seus personagens, ou mesmo durante o processo. Isso permite que os personagens sejam melhor elaborados taticamente, uma vez que os jogadores podem priorizar habilidades e capacidades que eles sabem que serão úteis durante a aventura, o que melhora o engajamento dos envolvidos.

Isso também propicia que os jogadores trabalhem os antecedentes e personalidade dos personagens para que se encaixem melhor na trama, criando relações prévias com outros personagens, definindo relações com localidades, elaborando objetivos secundários, etc. Isso fomenta o interesse e aptidão dos jogadores em explorar os elementos narrativos do jogo, enriquecendo a experiência.

A exemplo disso, na simulação, o mestre compartilhou previamente que os personagens se conheceram no caminho para uma grande cidade, onde o jogo deve se iniciar, e cada personagem deve possuir um motivo para viajar até lá. Também foram citados o envolvimento de estudantes de magia, artefatos proibidos e um cenário urbano explorável. Diante disso, os jogadores criaram seus personagens, como consta nas subseções a seguir.

### A.3.1 Jogador 01 - Personagem Lammar Hyen

Seu personagem é um mago que busca aprimorar seu domínio arcano. Ele viajará para a cidade visando atender ao convite de um antigo professor de magia, para participar de uma pesquisa secreta. As escolhas do jogador 01 para a construção de seu personagem estão listadas abaixo:

- **Espécie e Raça:** Humano Ehleni
- **Qualidade:** Sagaz
- **Defeito:** Indisposto
- **Treinamentos:** Conjurador e Estudioso
- **Antecedentes:** Nobreza e Arcanista
- **Personalidade:** Orgulhoso e Discreto

Na figura A.1 é demonstrado o resultado da finalização da criação do personagem do jogador 01, exibindo a soma e cálculo dos pontos do mesmo, conforme as operações descritas ao longo da sessão ??.

capacidad	valor	bonus	INTEGRIDADE	FÍSICA	MENTAL
Estrutura					
FORÇA	1	1			
MOBILIDADE	0	0			
RESISTÊNCIA	-1	-1			
Coordenação					
AGILIDADE	0	0			
SUTILEZA	0	0			
DESENVOLTURA	0	0			
Saúde					
VIGOR	0	2			
TENACIDADE	1	3			
IMUNIDADE	0	2			
Inteleco					
ATENÇÃO	3	5			
OBSTINAÇÃO	2	4			
RAZÃO	3	5			
Social					
CARISMA	0	2			
RETÓRICA	1	3			
IMPOSIÇÃO	0	2			
			Conhecimentos	valor	bonus
			Nobreza	3	8
			Magia Eletricidade	3	8
			Magia Fogo	3	8
			Arma	ATK	Propriedade
			Adaga	-4	traíçoeira aparadora

Figura A.1: Pontos do personagem Lammar Hyen

Após a criação, o jogador obteve para o personagem uma armadura da categoria vestimenta reforçada, com 1 ponto de proteção, além de uma adaga, para a qual não cumpre os requisitos de força e agilidade, resultando em -4 de bônus de ataque. Quanto a magias, sem entrar em detalhes, o tipo e nível de conhecimento arcano do personagem define quais magias ele tem acesso, e a base de poder com cada magia depende dos requisitos da mesma.

### A.3.2 Jogador 02 - Personagem Talyria'Yamari

Sua personagem é uma ladra que cometeu alguns erros no passado sendo expulsa de sua guilda. Portanto, ela busca se instalar na cidade para recomeçar sua vida, e se possível conseguir uma vaga em uma nova guilda de ladrões. As escolhas do jogador 02 para a construção de sua personagem estão listadas abaixo:

- Espécie e Raça:** Igneray Azerak
- Qualidade:** Sorrateira
- Defeito:** Dispersa
- Treinamentos:** Evasivo e Furtivo
- Antecedentes:** Explorador, criminoso e perseguidor
- Personalidade:** Hiperativa e Inquieta

Na figura A.2 é demonstrado o resultado da soma e calculo dos pontos de capacidade da personagem, conforme as operações descritas ao longo da sessão ??.

capacidade	valor	bonus	INTEGRIDADE FÍSICA MENTAL		
Estrutura			2	2	0
FORÇA	-1	1			
MOBILIDADE	1	3			
RESISTÊNCIA	0	2			
Coordenação					
AGILIDADE	2	4			
SUTILEZA	4	6			
DESENVOLTURA	4	6			
Saúde					
VIGOR	1	3			
TENACIDADE	0	2			
IMUNIDADE	0	2			
Inteleco					
ATENÇÃO	2	2			
OBSTINAÇÃO	-1	-1			
RAZÃO	1	1			
Social					
CARISMA	0	0			
RETÓRICA	0	0			
IMPOSIÇÃO	0	0			
Conhecimentos			valor	bonus	
Comercial			3	9	
Militar			2	8	
Criminalidade			1	7	
Arma		ATK	Propriedade		
Espada Curta		1	aparadora		

Figura A.2: Pontos da personagem Talyria'Yamari

Quanto aos itens o jogador opta por fazer com que seu personagem traje uma armadura da categoria “armadura leve”, a qual possui 3 pontos de proteção. Ela também possui uma espada curta, cujo calculo dos requisitos resultam em base ofensiva 1. Adicionalmente, o jogador 02 também teria selecionado ferramentas de ladrão, entre outros itens, para a personagem.

### A.3.3 Jogador 03 - Personagem Malvin Malori

Seu personagem é um guerreiro contrabandista que atua no mercado negro de componentes alquímicos, ele está indo para a cidade para entregar uma encomenda, um item alquímico perigoso e caro. As escolhas do jogador 03 para a construção de seu personagem estão listadas abaixo:

- Espécie e Raça:** Humano Azhad
- Qualidade:** Persuasivo
- Defeito:** Instintivo
- Treinamentos:** Ofensivo e Acrobático
- Antecedentes:** Militar, criminoso e comercial
- Personalidade:** Desconfiado e Impulsivo

A raça “Azhad”, diferente das demais, não consta como uma das opções implementadas na aplicação, servindo como lembrete para o fato de que existem muitas opções de escolha além das comentadas na sessão 5.4. Na figura A.3 é demonstrada a relação de pontos calculados, tal como os outros personagens.

capacidade	valor	bonus	INTEGRIDADE FÍSICA MENTAL		
Estrutura			<b>3</b>	<b>2</b>	<b>1</b>
<b>FORÇA</b>	<b>0</b>	<b>2</b>			
<b>MOBILIDADE</b>	<b>2</b>	<b>4</b>			
<b>RESISTÊNCIA</b>	<b>2</b>	<b>4</b>			
Coordenação			<b>Suporte de Peso</b>	<b>22</b>	
<b>AGILIDADE</b>	<b>1</b>	<b>3</b>	<b>Iniciativa</b>	<b>4</b>	
<b>SUTILEZA</b>	<b>1</b>	<b>3</b>			
<b>DESENVOLTURA</b>	<b>2</b>	<b>4</b>	<b>GUARDA</b>	<b>9</b>	
Saúde			<b>Proteção</b>	<b>4</b>	
<b>VIGOR</b>	<b>1</b>	<b>4</b>	<b>Pós Guarda</b>	<b>13</b>	
<b>TENACIDADE</b>	<b>1</b>	<b>4</b>			
<b>IMUNIDADE</b>	<b>2</b>	<b>5</b>	<b>BASE OFENSIVA</b>	<b>3</b>	
Inteleco			<b>BASE PODER</b>	<b>4</b>	
<b>ATENÇÃO</b>	<b>0</b>	<b>1</b>			
<b>OBSTINAÇÃO</b>	<b>3</b>	<b>4</b>	Conhecimentos	valor	bonus
<b>RAZÃO</b>	<b>-2</b>	<b>-1</b>	Comercial	<b>3</b>	<b>7</b>
Social			Militar	<b>2</b>	<b>6</b>
<b>CARISMA</b>	<b>0</b>	<b>1</b>	Criminalidade	<b>1</b>	<b>5</b>
<b>RETÓRICA</b>	<b>2</b>	<b>3</b>			
<b>IMPOSIÇÃO</b>	<b>0</b>	<b>1</b>	Arma	ATK	Propriedade
			Espada Curva	<b>2</b>	traiçoeira aparadora

**Figura A.3: Pontos do personagem Malvin Malori**

O jogador opta por fazer com que seu personagem traje uma armadura da categoria “armadura básica”, a qual possui 4 pontos de proteção. Ele também possui um par de espadas curvas, com base ofensiva 2 para ambas. O jogador também escolhe alguns itens úteis para o personagem, como um kit de sobrevivência com barraca, corda e rações, além do tal componente alquímico que o personagem está transportando.

Comumente, “criação de personagens” e “planejamento da aventura” são atividades realizadas em paralelo, sendo proveitosa a troca de informações entre o mestre e jogadores. Anteriormente, foi pontuado como informações sobre a trama podem afetar a criação dos personagens, mas da mesma forma, conhecer os antecedentes dos personagens pode afetar o planejamento da trama pelo mestre.

Dessa forma, o mestre pode trabalhar relações narrativos entre os personagens e a trama, como definir que a citada aprendiz do mago foi colega de classe de Lammar, ou que o mago desaparecido quem teria encomendado o item alquímico que Malvin está transportando, ou ainda, trabalhar a participação de uma guilda de ladrões nesse enredo, garantindo um envolvimento para a personagem Talyria.

Apesar do foco em apresentar o processo de criação dos personagens dos jogadores, é válido pontuar que o mestre pode utilizar o mesmo processo para criar os seus personagens coadjuvantes e antagonistas. Porém, o livro de regras contem uma lista de personagens prontos, incluindo pessoas, animais, monstros e todo tipo de criatura que compõe o cenário, para que o mestre apenas escolha quais deseja incluir.

Todo esse processo de criação do plano de fundo, definição de enredos e desafios, além da elaboração de personagens pode levar horas ou semanas, a depender de diversos fatores, como a complexidade do sistema de regras, o nível de detalhamento da história, a disponibilidade dos participantes, o envolvimento dos jogadores, etc.

## A.4 Preparativos da partida

As tarefas que antecedem o jogo também incluem a preparação física do material, como fichas, dados, mapas, ilustrações, etc. Em alguns casos são utilizadas miniaturas, o mesmo maquetes. Tudo depende da dedicação e do grau de detalhamento desejado pelos envolvidos. Além dos materiais base, comentados na sessão 2.3.1, os jogadores definem onde e quando a partida deve ocorrer.

Na simulação, mestre e jogadores se reúnem no dia e horário marcado, na casa do mestre. Sobre a mesa estão dispostos os dados multifacetados, fichas de personagem impressas e preenchidas, mapas desenhados pelo mestre e um notebook para consulta de regras na página do jogo na internet. Como qualquer objeto pode ser usado como referência de posicionamento, um tabuleiro de Xadrez foi improvisado como painel quadriculado e as peças serão usadas no lugar de miniaturas.

Finalizados os preparativos, a partida de jogo pode ser finalmente iniciada. Como descrito na sessão 2.4, a partida começa com uma introdução que contextualiza a aventura, em seguida, a cada cena descrita os jogadores podem realizar interações, cujas consequências resultam em novas cenas.

## A.5 Introdução da trama

No início da primeira partida o mestre contextualiza e introduz a trama através de uma narração de apresentação — Vocês se conheceram em um navio cargueiro, e foram colegas de quarto por aproximadamente um mês, nesse período vocês se aproximaram e descobriram que possuem um objetivo em comum, chegar até a cidade de “Fendall Myr”, localizada no reino de “Eremyr”, região leste de “Jazin Kindel”.

Avançando na narrativa, o mestre introduz os primeiros elementos da aventura, ele diz — Após o desembarque, vocês ouviram alguns boatos a respeito da crescente periculosidade da região, motivo pelo qual decidiram continuar viajando juntos, uma vez que seriam oito dias de viagem do porto até a capital. Nesse meio tempo vocês se conhecem ainda melhor, portanto apresentem seus personagens e digam o que eles sabem uns dos outros.

Dada a oportunidade, os jogadores começam a participar da narrativa. O jogador 01 diz — Meu nome é “Lammár”, da casa “Hyen”, sou um mago humano natural do grande império de “Akhamep”. Sou alto e magro, de pele negra e barba longa, trajando túnicas elegantes e turbante na cabeça. Você pode notar que falo e me comporto como um erudito, e estou sempre lendo ou observando as estrelas. Porém, não sabem nada sobre meus objetivos, sou muito reservado quanto aos meus assuntos.

Na sequência, diria o jogador 02 — Eu sou “Talyria’Yamari”, uma jovem da raça “Ygneray”, etnia “Iralik”, tenho os cabelos ondulados e curtos, sendo ruivos em tonalidade que se aproxima do rosa, como é típico entre os membros do meu clã. Tenho as orelhas pontudas e largas, meus chifres são curvos e enfeitados com anéis. Também tenho algumas tatuagens nas mãos e na cauda. Você notam que sou bastante inquieta e falante, a essa altura já sabem que eu fui expulsa da minha guilda de ladrões, e que estou buscando recomeçar na cidade de “Fendall”.

E o jogador 03 — Meu personagem é “Malvin Malori”, um humano das profundezas, guerreiro “Azhad”. Naturalmente evito a luz do sol, sou mais ativo à noite e costumo cobrir o meu corpo inteiro com trajes escuros. Tenho 1,75m de altura, corpo esguio, e como qualquer “Azhad” tenho a pele e cabelos tão brancos quanto um fantasma. O que vocês sabem sobre mim é que tenho negócios na cidade, digo que sou um mercador, mas não entrei em detalhes.

As descrições físicas dos personagens viabilizam que todos os participantes possam visualizar, em sua imaginação, os protagonistas da história. Mestre e jogadores podem seguir debatendo o que os personagens sabem uns dos outros, que tipo de laço foi forjado na viagem, do que se tratam os referidos perigos citados pelo mestre, entre outros elementos que compõe o contexto inicial da trama.

## A.6 Dinâmica do jogo

Como mencionado na sessão 2.4, a narração do mestre deve passar por uma transição, saindo do momento de contextualização para entrar na primeira cena, ele narra — Vocês seguiram pelas estradas direto para o vilarejo de “Jahib”, parando apenas para acampar. Vocês chegaram por volta do sétimo dia de viagem, e se hospedaram na pousada “Oásis da Aurora”. Vocês estão a menos de um dia de Fendall, e pretendem seguir viagem amanhã de manhã. Agora, vocês estão em uma taberna acoplada à pousada, sentados á mesa e prontos para pedir o jantar.

### A.6.1 Cena

O mestre compõe a cena descrevendo e detalhando o ambiente em que os personagens estão, imergindo os jogadores na narrativa — O estabelecimento é bastante amplo, com várias mesas de madeira, circulares como escudos. Suas narinas são invadidas pelo cheiro dos temperos e da carne de javali sendo assada. Entretanto, o local está recebendo muitos viajantes, provocando uma algazarra de ruídos.

Outro aspecto de uma cena são as oportunidades de interação — Ao fundo dessa estalagem existe um balcão de madeira, de onde uma senhora gorda de pele escura grita comandos às garçonetes. Alguns homens, dos mais barulhentos por sinal, estão aglutinados em torno de uma mesa no canto do salão, aparentemente entretidos com algum jogo de apostas. Vocês não sabem o motivo de toda aquela movimentação, estão descansados, porém, começando a sentir fome, o que vão fazer?

### A.6.2 Interação

Cabe aos jogadores descrever as atitudes e intenções de seus personagens dentro da cena, fazendo com que realizem qualquer ação condizente a suas capacidades e personalidade. O jogador 01 diz — Eu quero puxar conversa com essa senhora atrás do balcão, pedir alguma coisa para beber e aproveitar para tentar descobrir se toda essa movimentação é atípica ou não. Então eu me levanto, e sigo em direção a ela.

### A.6.3 Reação

Como resposta, o mestre deve avaliar as escolhas e dar continuidade à cena, utilizando seu planejamento e capacidade de improvisação para narrar e detalhar os acontecimentos. Ele diz — Certo, você se levanta e segue em direção ao balcão, desviando das pessoas e das mesas em seu caminho. A senhora está guardando uma garrafa na estante, enquanto berra com um homem bêbado que dorme recurvado sobre o balcão ao lado de um copo quase vazio, ela não parece notar a sua aproximação...

Dessa forma, o mestre conclui a terceira etapa da dinâmica, a reação do cenário dentro da narrativa. Como já pontuado, a terceira etapa inclui a narração da continuidade da cena, remetendo a primeira etapa e fechando o ciclo da dinâmica.

Ou seja, a partir daqui o jogador pode decidir como abordar a senhora, seja oferecendo ouro para chamar-lhe a atenção, iniciando uma conversa formal, escolhendo falar com uma das garçonetes que pareça menos ocupada, ou mesmo sondando um pouco mais a situação, postergando a abordagem. De qualquer forma, o jogador deve avaliar a situação e procurar uma nova interação, avançando na trama.

Outro fator notável é a improvisação, uma vez que o mestre havia planejado elementos-chave como, onde a aventura se inicia, a senhora atrás do balcão com informações úteis e a aglomeração atraída pelo torneio na cidade, entretanto, os demais elementos narrados são improvisados, apenas para reforçar a imersão e coerência das cenas, como o formato das mesas, o grupo de apostadores, o bêbado sobre o balcão, etc.

## A.7 Atuação

Como mencionado nos últimos parágrafos da sessão 2.3.3, a atuação é um aspecto recorrente de jogos da modalidade. Por exemplo, o jogador 01, poderia seguir a abordagem dizendo — Eu me aproximei e fui em direção ao homem sobre o balcão, o agarrei pela gola e o puxei com força para que se levantasse. Peguei o copo e entreguei na mão dele, depois o empurrei gentilmente para que saísse dali.

Enquanto fala, o jogador usa suas expressões faciais para demonstrar a face de seu personagem na cena. Adicionalmente, a ação de “puxar pela gola”, “entregar o copo” e “empurrar gentilmente” foram representadas, não apenas pela descrição falada, mas com gestos, para ilustrar a linguagem corporal do personagem.

O mestre narra em resposta — O homem embriagado sentiu o puxão e levantou irritado, mas ao receber o copo na mão ele abriu um sorriso e seguiu cambaleante após o empurrão, sem oferecer resistência ou mesmo olhar para você. A senhora lhe devolve um olhar rápido e um sorriso discreto de aprovação.

Ao que o jogador 01 prossegue — Eu me sento e cruzo os braços sobre o balcão, falei o seguinte: “Vejo que a casa está cheia, devo estar diante de uma dama tão rica quanto o próprio rei”. Nesse momento o jogador atua, mudando seu tom de voz e demonstrando que seu personagem fala em tom amigável.

Ao mestre cabe demonstrar a reação do cenário, narrando — A senhora se aproximou e foi falando enquanto lhe servia um vinho não solicitado... — O mestre imita uma voz feminina em tom grave, e aproveita a oportunidade para revelar uma informação — “Ora, não seja tolo meu caro, essa gente toda só está aqui por causa do torneio de amanhã, se não eu já teria me mudado para a capital”.

## A.8 Aplicação de regras

Em meio a tantas formas de interação, os jogadores podem se colocar em situações que exigem a aplicação de regras. Adiante, a simulação apresenta algumas situações que demonstram o uso de algumas das regras do jogo “Mysthen RPG”.

### A.8.1 Solicitação e realização de teste

O jogador 02 que joga com a ladra Talyria’Yamari, comenta — Eu me virei para o Malvin e disse: “Veja se consegue pedir algo para bebermos, vou dar uma volta, estarei aqui em um minuto”. Então eu me levanto e começo a caminhar pela estalagem, quero observar as pessoas, tentar ouvir algum boato ou algo assim.

Em resposta, o mestre diz — Não é difícil obter informações, mas como o local está muito cheio de pessoas falando, você tem chances de se distrair, ou dar o azar de não coletar nada importante. Portanto, faça um teste de Atenção.

Como explicado na sessão ??, os testes de capacidades servem para determinar se o personagem foi bem-sucedido ao realizar uma ação com chance de falha. Neste caso, o teste serve para determinar quão sortuda e atenta foi a personagem para coletar e filtrar informações úteis. Analisando a situação, o mestre define que a dificuldade do teste será “Médio”, como determinado na figura ??, estabelecendo Nível de Dificuldade (Nd) 14.

Uma vez solicitado o teste, o jogador 02 rola o dado de 20 faces (d20), obtendo o valor 13. Por se tratar de um teste de “Atenção” ele verifica que o valor dessa capacidade para a sua personagem é 2, como consta na figura A.2. Essa capacidade está relacionada a aptidão mental do personagem, que no caso de Talyria’Yamari é 0. Portanto, no teste é calculado  $13 + 2 + 0 = 15$ .

### A.8.2 Conclusão e narração do teste

Por obter o resultado 15 para sua tentativa, a personagem superou a dificuldade estabelecida, Nd14. Portanto, o mestre deve fornecer algumas informações úteis, o que não ocorreria caso ela tivesse falhado no teste. Adicionalmente, o mestre poderia fornecer informações ainda mais úteis caso o sucesso no teste fosse maior.

Interpretando o resultado, o mestre improvisa a reação compondo uma cena — Caminhando você nota muitos homens vestindo túnicas caras e joias, nota também alguns brasões e insígnias de guildas, companhias mercenárias e ordens de cavalaria. Você ouviu dois boatos que lhe chamaram a atenção, um se referia a um torneio que deveria ocorrer em breve na capital, outro sobre um ataque que teria ocorrido a um vilarejo próximo.

### A.8.3 Iniciando um Combate

O combate é um dos momentos onde as regras são mais numerosas e criteriosas, e como mencionado, esse aspecto do jogo não consta na aplicação, e deve ser abordado em trabalhos futuros. Portanto, pode-se considerar que aplicação cobre a criação de personagens e as mecânicas de exploração do jogo.

Porém, a simulação se propõe abordar esse tema, comentando superficialmente as regras envolvidas, as quais são verificáveis na página do jogo na plataforma notion.com. Portanto, considera-se que no decorrer da simulação o jogador 02 decidiu que sua personagem iria furtar um homem na taberna, porém, ela falha no teste. A personagem foi vista e inicia uma confusão, de forma que Malvin e Talyria são expulsos da estalagem.

Assim um combate se inicia do lado de fora, contra dois mercenários, um bêbado alto e magro com uma espada, e o outro sóbrio, calvo e barbado, usando um punhal. Vale comentar que o mestre não planejou este combate, logo ele precisa improvisar as características dos adversários, tal como pontos de capacidade, armas usadas, etc.

### A.8.4 Ordem de Combate

A ordem de combate é definida por testes de “Iniciativa”. Considerando que cada um dos personagens realizou o teste, o mestre analisa os resultados obtidos e estabelece a seguinte ordem: 1º Mercenário calvo, 2º Talyria, 3º Malvin, 4º Mercenário bêbado.

### A.8.5 Turno de Combate

Uma vez estabelecida a ordem de combate, os turnos são iniciados. O controle de tempo é mais rigoroso durante o combate, e nessas condições a quantidade de ações possíveis é limitada pelas regras, obrigando que os jogadores escolham cuidadosamente as interações e testes que vão fazer.

### A.8.6 Ataque

O primeiro ataque é do mercenário calvo, criado de improviso, com “Integridade” 2, “Base de Ataque” 1 usando o punhal, “Guarda” 9 e “Pós-Guarda” 13. O mestre narra — Vocês foram chutados para fora da estalagem e as pessoas se aglomeram eufóricas para assistir essa briga. O mercenário com o punhal vai em direção a Talyria desferindo estocadas e berrando como um louco.

O mestre rola os dados e calcula os valores pertinentes, obtendo o valor final 13 para o ataque do personagem em questão. Conforme consta na figura A.2, a personagem Talyria, alvo do ataque, possui Guarda 11 e Pós-Guarda 14. Logo, de acordo com as regras de guarda, ela foi atingida entre as guardas, sofrendo um redutor de postura.

### A.8.7 Postura

O jogador 02 adquiriu um redutor da categoria postura, acumulando 1 ponto de “Adversidade”. De acordo com as regras da sessão ??, a personagem possui -1 em todos os testes, prejudicando a efetividade de suas ações. Além disso, a adversidade também reduz 1 ponto da guarda do personagem, como se abrisse uma pequena brecha temporária em suas defesas.

Geralmente, redutores como “Postura” são revertidos narrativamente, porém, existem casos em que algumas regras se aplicam nesse sentido, sendo este um desses casos. Por norma, o redutor se aplica até a próxima ação do personagem afetado, ou seja, os efeitos dessa perda de postura permanecem ativos durante todo o turno atual, se esgotando assim que chegar o momento de ação da personagem no turno seguinte.

Como mencionado na sessão A.8.5 deste apêndice, o número de ações que o personagem pode realizar no turno é limitada. Portanto, cabe ao jogador 02 decidir entre tentar reverter a situação ainda neste turno (gastando sua ação para tentar se equilibrar ou se reposicionar) ou aceitar a condição (preservando sua ação para outra finalidade).

### A.8.8 Contra ataque

Analisando a situação, o jogador 02 opta por atacar, seu teste de ataque resulta no valor 9, mas a perda de postura reduz o resultado em 1 ponto. Como citado anteriormente, o mercenário calvo possui Guarda 9 e Pós-Guarda 13, logo, o ataque foi inferior a “Guarda” do inimigo, e consiste em erro, ou seja, não causa ferimento ou perda de postura.

Após interpretar os valores, o mestre cria uma cena que represente os resultados, narrando — Ele avançou agressivamente e desferiu várias facadas na altura do seu abdome, algumas atingiram sua armadura, mas não perfuraram o tecido. A intensidade do ataque te obrigou a recuar, você arriscou alguns golpes, mas seu posicionamento é totalmente desfavorável, você não teve espaço para atingi-lo efetivamente.

### A.8.9 Deslocamento em combate

O jogador 01, que está de fora do combate, decide ir ao encontro de seus aliados, mas para isso, ele precisa sair do balcão e cruzar toda a taberna até a saída. Como em nenhum momento o mestre definiu essa distância, ele improvisa a medida de 12 metros. O personagem possui deslocamento de 7 metros por turno e precisa passar por várias pessoas que atrapalham seu caminho, logo, pelas regras de deslocamento, fica estabelecido que ele só chegará ao local do combate no início do terceiro turno.

### A.8.10 Ferimento

O próximo na ordem de combate é o jogador 03, que decide atacar o mercenário que avançou sobre sua aliada. Ele diz — Meu impulso é proteger a Talyria, então eu empunho minhas duas espadas curvas e avanço contra o mercenário. O jogador obtém 17 como valor final de seu ataque. Como este inimigo possui Guarda 9 e Pós-Guarda 13, o ataque de Malvum o atinge acima da Pós-Guarda, provocando tanto a perda de postura, quanto um ferimento.

O mestre narra — Você viu avançando sobre a Talyria, e sua reação imediata foi agir contra ele, desferindo golpes cortantes com ambas as espadas. Ele esquiva de alguns golpes, mas não de todos, sofrendo alguns cortes rasos no peito e nos braços.

O ferimento acumula 1 ponto de “Adversidade” para o personagem atingido. Porém, diferente da perda de postura, o ferimento não é anulado de um turno para o outro. Ao invés disso, é preciso tratar o ferimento narrativamente, com medicamentos, bandagens e outras medidas, algo que não é possível durante o combate.

### A.8.11 Fim do primeiro turno

O próximo a agir é o outro mercenário, o homem magro e bêbado que empunha uma espada. Como o homem está embriagado, o mestre determina bônus baixos para ele, sendo “Integridade” 1, “Base de Ataque” -3 usando a espada e “Guarda” 6, além disso, o homem não usa nenhum tipo de armadura, de forma que sua “Pós-Guarda” também é 6.

Na cena, considera-se que o homem ataca Malvin, mas seu ataque é inferior a guarda de seu alvo, resultando em um erro. O que é relevante a ser pontuado neste momento, é que esse personagem é o último a agir pela ordem de combate estabelecida, portanto, sua ação demarca o fim do primeiro turno.

### A.8.12 Troca de Turnos

Ao fim de um turno é necessário definir a nova ordem de combate, obrigando os jogadores a readaptar suas estratégias de acordo com a nova ordenação de ações. Uma vez realizadas as rolagens de iniciativa, o mestre pode definir a nova ordem de combate, sendo: 1 Malvin, 2 Talyria, 3 mercenário calvo, 4 mercenário bêbado.

Como os turnos são recortes de tempo, também é necessário verificar fatores temporais sempre que ocorre a troca de turnos. No caso da simulação, a condição de perda de postura da personagem Talyria, do jogador 02, dura até o momento de ação do personagem no turno seguinte ao que o redutor foi sofrido, portanto, os efeitos negativos da condição de perda de postura devem ser revertidos assim que o jogador 02 puder agir no turno que se inicia.

Outro caso é a chegada do personagem Lammar, do jogador 01, ao combate. Está definido que ele levará o tempo de 2 turnos completos para chegar até o local onde a luta acontece. Portanto, a troca de turnos estabelece a conclusão do primeiro turno de espera, evidenciando que o mesmo ainda não teve tempo suficiente para chegar ao local do combate.

### A.8.13 Morte

No novo turno, o personagem do jogador 03 será o primeiro a agir. Ele decide avançar novamente sobre o homem calvo. Este inimigo possui originalmente Guarda 9 e Pós-Guarda 13, porém, a perda de postura e o ferimento sofrido no turno anterior reduzem suas defesas em 2 pontos, resultando em Guarda 7 e Pós-Guarda 10.

No ataque, Malvin obtém o valor final 13, superando a Pós-Guarda atual deste inimigo, que será novamente afetado com perda de 1 ponto de postura e 1 de ferimento, acumulando 4 pontos de adversidade. Quando o acúmulo de ferimentos alcança o valor da “Integridade” do personagem, o mesmo está sob risco de vida, e na concepção do adversário em questão, o mestre definiu 2 pontos de integridade para ele.

O mestre conclui — Você continua avançando implacável enquanto ele tenta recuar. Entre os vários golpes desferidos com ambas as lâminas, você atinge um corte rápido no pescoço do inimigo, que sente o golpe e se apavora, hesitando por um segundo. Foi tempo suficiente para que você o derrotasse, estocando seu peito com a sua outra lâmina. O outro homem, horrorizado, desiste do combate e foge.

Como os personagens do mestre costumam ser usados como desafios a serem enfrentados e superados, principalmente os inimigos eventuais e não muito importantes, é comum que eles morram assim que o acúmulo de ferimentos alcance sua integridade, como ocorreu no trecho citado.

Porém, os personagens dos jogadores, ou mesmo alguns personagens importantes do mestre, podem entrar em colapso ao invés de morrer, ativando uma mecânica na qual o personagem está sob risco de morte, mas pode se salvar, uma condição na qual o sistema dispõe regras para postergar a perda do personagem, dando chances para que ele se salve da morte.

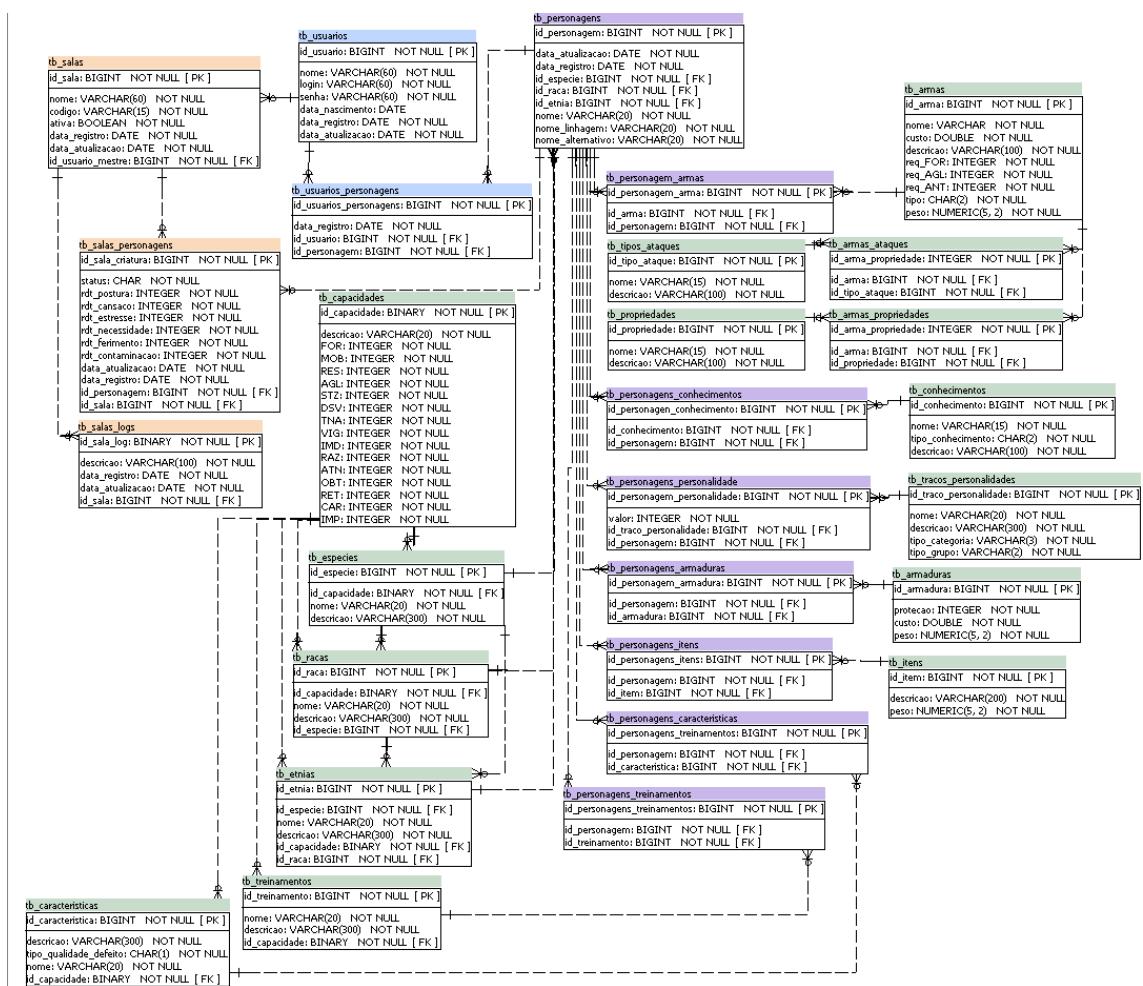
### A.8.14 Conclusão da simulação

Fica demonstrado o funcionamento do jogo “Mysthen RPG” em uma sessão de jogo hipotética. Mesmo resumindo vários tópicos e regras, foi possível ilustrar como as regras se aplicam e integram a narrativa, permeando as decisões dos jogadores, que interpretam personagens e constroem uma história em conjunto.

Em uma situação real, essa sessão de jogo se estenderia por horas a fio, uma vez que a confusão na taberna traria vários desdobramentos narrativos, a partir dos quais surgiriam novas situações e escolhas a serem feitas. Gradativamente, os eventos da trama principal e os objetivos dos personagens poderiam ser abordados e desenvolvidos, ao passo que os jogadores acabam criando situações e gerando problemáticas inesperadas.

Pensando na aplicação desenvolvida neste trabalho, fica demonstrado como funciona o jogo adaptado, e como existem vários pontos passíveis de automatização, simplificação e otimização. O RPG é um jogo casual que demanda preparação e comprometimento, sendo bem-vinda uma aplicação que gerencie processos e reduza a complexidade, permitindo que os jogadores se concentrem no lado narrativo e estratégico da atividade.

# Apêndice B – Diagrama Entidade-Relacionamento



**Figura B.1:** Diagrama Entidade-Relacionamento referente ao banco de dados modelado para a aplicação