

Aplicação Web para Gerenciamento de Fichas de RPG

Guilherme Guglielmin Munhoz de Paula¹
Lidiane Visintin¹

¹Instituto Federal de Santa Catarina - Câmpus Lages (IFSC)
R. Heitor Villa Lobos, 225 - Sao Francisco, 88506-400- Lages – SC – Brasil

guilherme.gml3@aluno.ifsc.edu.br

lidiane.visintin@ifsc.edu.br

Abstract. *The world of digital games has evolved, creating space for applications focused on tabletop role-playing games, whose current applications do not meet the needs of all players. The objective of this work is to develop a Web application for character sheet management, offering a flexible solution that unifies different systems into a single platform. The methodology consists of applied research with a qualitative approach, oriented toward the development and validation of a Minimum Viable Product. Validation in a closed beta test with players indicated good usability and acceptance, signaling adoption potential.*

Resumo. *O mundo dos jogos digitais tem evoluído, abrindo espaço para aplicações voltadas aos jogos de RPG de mesa, cujos aplicativos atuais não suprem as necessidades de todos os jogadores. O objetivo deste trabalho é desenvolver uma aplicação Web para gerenciamento de fichas de personagem, oferecendo uma solução flexível que unifique diferentes sistemas em uma única plataforma. A metodologia consiste em pesquisa aplicada, de abordagem qualitativa, orientada ao desenvolvimento e validação de um Mínimo Produto Viável. A validação, em teste beta fechado com jogadores, indicou boa usabilidade e aceitação, sinalizando potencial de adoção.*

1. Introdução

O desenvolvimento de jogos digitais consolidou-se como um campo relevante de investigação acadêmica e tecnológica, abrangendo diferentes aspectos que vão desde a concepção de jogos digitais até a criação de ferramentas que apoiam o processo de produção.

Desta forma, o presente trabalho está inserido no campo de pesquisa relacionado ao desenvolvimento de jogos digitais, com ênfase nas ferramentas que os apoiam. Muitos jogos contemporâneos incorporam elementos originados dos jogos de RPG (Role-Playing Game, ou jogo de interpretação de papéis), como a criação de personagens, progressão por atributos e construção narrativa baseada em escolhas. Títulos como Baldur's Gate 3, baseado diretamente nas regras do sistema *Dungeons and Dragons (D&D)*, evidenciam essa influência ao transportar a experiência dos RPGs de mesa para o ambiente digital (Larian Studios, 2025). Esse intercâmbio entre mídias contribuiu para o fortalecimento da cultura de RPG como um todo, influenciando tanto jogos eletrônicos quanto o ressurgimento do interesse pelos sistemas tradicionais de mesa.

Nesse contexto, os jogos de RPG de mesa continuam a crescer em popularidade, impulsionados por comunidades ativas em plataformas como Twitch, YouTube e redes sociais. Transmissões ao vivo, como as realizadas pelo grupo Critical Role, transformaram as sessões de RPG em entretenimento de massa, atraindo milhões de espectadores e ampliando o alcance do hobby. Além das plataformas digitais, eventos presenciais dedicados ao tema, como a Gen Con, tornaram-se pontos de encontro essenciais para fãs e jogadores, promovendo a interação e o crescimento da comunidade (University of Wisconsin-Parkside, 2024). Essas experiências colaborativas e imersivas reúnem grupos de jogadores em sessões contínuas que simulam mundos fictícios complexos. A dinâmica baseia-se em sistemas de regras que determinam como os personagens interagem com o ambiente, geralmente por meio de fichas que registram atributos, habilidades e outras informações relevantes.

Com o surgimento de diversos sistemas, gerenciar personagens utilizando fichas de papel torna-se uma tarefa trabalhosa e suscetível a erros. Embora existam ferramentas como o *Roll20* e o *D&D Beyond*, essas soluções não suprem integralmente as necessidades da comunidade. O primeiro é útil para a condução tática de sessões, mas apresenta uma curva de aprendizado elevada para novos usuários (Roll20, 2024b). Já o *D&D Beyond* limita-se ao sistema D&D e possui uma interface considerada confusa por parte da comunidade (Reddit, 2024).

Com base nessas limitações, a comunidade de RPG busca soluções alternativas que sejam mais acessíveis e versáteis. Projetos como o *Foundry VTT* têm ganhado popularidade por serem altamente customizáveis, embora exijam conhecimentos técnicos avançados e a hospedagem própria por parte do usuário (Foundry Gaming, LLC, 2024a). Nesse contexto, aplicativos de gerenciamento de fichas, que permitam integração com diferentes sistemas e sejam intuitivos mesmo para usuários iniciantes, representam uma demanda crescente dos jogadores (RPGamer, 2020).

Diante desse contexto, estabelece-se o seguinte problema de pesquisa: *De que forma pode ser desenvolvido um Mínimo Produto Viável (MVP) de uma aplicação Web para o gerenciamento de fichas de RPG de múltiplos sistemas?* Observa-se que as soluções existentes são, em geral, limitadas a sistemas específicos, apresentam elevada complexidade de uso ou demandam conhecimentos técnicos para customização, o que compromete sua adoção por jogadores e mestres com diferentes perfis.

Este projeto propõe o desenvolvimento de uma aplicação Web capaz de gerenciar fichas de RPG de múltiplos sistemas, permitindo cadastro, edição e visualização dos dados. O Mínimo Produto Viável (MVP) incluirá suporte a pelo menos dois sistemas distintos, interface responsiva e funcionalidades como rolagem de dados integrada.

A solução foi construída utilizando C# com .NET no *back-end* e Angular no *front-end*, adotando o padrão de arquitetura limpa (*Clean Architecture*) para garantir separação de responsabilidades e facilidade de manutenção. A estrutura das fichas é dinâmica, baseada em JSON, e os dados armazenados em um banco NoSQL, adequado para modelagens flexíveis e não padronizadas.

Este trabalho pode ser classificado como uma pesquisa aplicada, com abordagem qualitativa e foco no desenvolvimento de um produto voltado à resolução de um problema real enfrentado por jogadores e mestres de RPG. Além desta seção introdutória, este docu-

mento é composto pelas seguintes seções: a seção 2 apresenta os assuntos relacionados a fundamentação do trabalho; a seção 3 apresenta o planejamento e a documentação acerca do MVP desenvolvido, a seção 4 registra o desenvolvimento da aplicação; e a seção 6 apresenta as considerações finais.

2. Referencial Teórico

Esta seção apresenta o referencial teórico, dividido em 4 partes distintas. Na subseção 2.1, são definidos os principais termos utilizados ao longo do trabalho, a fim de garantir clareza e padronização conceitual. Por conseguinte, nas subseções 2.2 e 2.3 são apresentados os conceitos de RPG de mesa e apresentados os sistemas de *Dungeons & Dragons* e *Old Quest*, respectivamente. Por fim, na subseção 2.4, são analisados trabalhos relacionados, como *Roll20*, *FoundryVTT*, *D&D Beyond* e *Lorde dos Dados*, detalhando suas funcionalidades, pontos fortes e limitações, com uma subsequente análise.

2.1. Dicionário de Termos

Esta subseção apresenta os principais termos relacionados a jogos de RPG de mesa, com o objetivo de padronizar o entendimento e garantir clareza ao longo deste trabalho.

- **RPG:** Sigla para *Role-Playing Game*, é um tipo de jogo em que os participantes assumem personagens fictícios e constroem narrativas colaborativas, mediadas por regras e conduzidas por um narrador.
- **Mesa:** Grupo formado por jogadores e pelo narrador (ou Mestre) que se reúne para jogar. Também pode se referir ao conjunto da sessão ou campanha em andamento.
- **Campanha:** Conjunto de sessões interligadas que compõem uma narrativa contínua, geralmente de longa duração.
- **Sessão:** Encontro individual de jogo em que parte da história da campanha é desenvolvida.
- **Personagem:** Identidade fictícia assumida por um jogador. Possui atributos, habilidades e histórico próprios, além de evoluir ao longo da campanha.
- **Jogador** (ou *Player*): Participante da mesa que interpreta um ou mais personagens dentro da narrativa.
- **Mestre** (ou *Mestre de Jogo, Narrador, Storyteller, Game Master (GM)* ou *Dungeon Master (DM)*): Participante responsável por conduzir a narrativa, arbitrar as regras e interpretar todos os personagens e elementos fora do controle dos jogadores.
- **NPC:** Sigla para *Non-Player Character*, representa qualquer personagem que não é controlado por jogadores. No RPG de mesa, é interpretado exclusivamente pelo Mestre e cumpre funções narrativas, sociais ou de desafio no jogo.
- **Atributos:** Valores numéricos que representam capacidades físicas, mentais ou sociais de um personagem, como Força, Destreza ou Inteligência.
- **Habilidades:** Ações especiais, conhecimentos ou poderes que um personagem pode utilizar, conforme definido pelas regras do sistema.
- **Dados:** Objetos utilizados para gerar resultados aleatórios. Comumente apelidados com a letra 'd' seguido da quantidade de faces. Os mais comuns são d4, d6, d8, d10, d12 e d20.
- **Testes:** Rolagens feitas com dados para verificar o sucesso ou fracasso de ações, com base nos atributos e modificadores do personagem.

- **Vantagem:** Situação em que o personagem tem maiores chances de sucesso, geralmente representada por rolar dois dados e escolher o melhor resultado.
- **Desvantagem:** Situação oposta à vantagem, em que se rolam dois dados e considera-se o pior resultado.
- **Livro de Regras:** Documento que contém o conjunto de normas, mecânicas e ambientações de um sistema de RPG.
- **Sistema de Jogo:** Conjunto de regras que define como o RPG será jogado. Inclui criação de personagens, mecânicas de combate, evolução e testes.
- **Ficha de Personagem** (ou *Ficha*, *Ficha de Jogador*): Documento que reúne todas as informações do personagem, como atributos, habilidades, equipamentos, histórico e progresso.
- **Nível:** Representa a progressão do personagem. Ao subir de nível, o personagem pode adquirir novas habilidades, melhorar atributos ou desbloquear capacidades especiais.
- **Classe** (ou *Profissão*): Representa a ocupação ou estilo de jogo do personagem, como Guerreiro, Mago ou Ladino.
- **Raça** (ou *Espécie*): Define a origem biológica ou ancestralidade do personagem, como Humano, Elfo ou Anão, geralmente com bônus e habilidades específicas.
- **Inventário:** Conjunto de itens que o personagem carrega, como armas, poções, pergaminhos e equipamentos diversos.
- **Ambientação** (ou *Cenário*): Mundo fictício em que a campanha ocorre, incluindo geografia, política, cultura, eventos históricos e elementos fantásticos.

2.2. O que é RPG

Mais do que um simples jogo, o RPG é uma experiência narrativa coletiva, na qual os participantes criam histórias em mundos fictícios por meio da interpretação de personagens e da tomada de decisão em tempo real. A prática do RPG envolve a construção de personagens com identidades e histórias próprias. Esses personagens estão atrelados ao mundo apresentado pela entidade conhecida como Mestre, o jogador responsável por descrever os cenários, interpretar NPCs e narrar as consequências das ações dos demais jogadores.

As ações realizadas pelos personagens não são arbitrárias. Elas passam por um sistema mecânico que introduz elementos de risco, probabilidade e consequência. Rolagens de dados, modificadores estatísticos e condições impostas por atributos ou habilidades são componentes centrais desse processo. Dessa forma, o RPG mantém uma tensão constante entre liberdade criativa e limitação mecânica, o que garante tanto a imprevisibilidade da narrativa quanto sua consistência interna.

Ao longo das décadas, o RPG ultrapassou as fronteiras do entretenimento e passou a ser utilizado como ferramenta em contextos pedagógicos, terapêuticos e de desenvolvimento de habilidades sociais e emocionais. Henning (2021) descreve o uso do RPG como uma ferramenta lúdica e motivacional na terapia de grupo, especialmente com adolescentes, permitindo que habilidades sociais sejam trabalhadas sem que o caráter terapêutico seja explicitamente percebido, o que facilita a adesão ao processo psicoterápico.

2.3. Sistemas de Referência

Para a fundamentação teórica e conceitual deste trabalho, foram selecionados dois sistemas de RPG com propostas distintas, de modo a representar diferentes abordagens dentro

do universo dos jogos de interpretação de papéis. O primeiro é o *Dungeons & Dragons*, um sistema amplamente reconhecido e consolidado internacionalmente, cuja estrutura complexa e rica influenciou profundamente o gênero. O segundo é o *Old Quest*, um sistema brasileiro contemporâneo que propõe uma experiência mais simplificada e voltada à narrativa. A seguir, discorrem-se as principais características de cada sistema, considerando suas origens, mecânicas e impactos culturais.

2.3.1. Dungeons & Dragons

O *Dungeons & Dragons* (D&D), publicado pela *Wizards of the Coast* (Wizards of the Coast, 2025), é um sistema de RPG de mesa que combina narrativa colaborativa com um conjunto estruturado de regras. Criado na década de 1970 por Gary Gygax e Dave Arneson, o sistema é considerado o precursor dos RPGs de mesa modernos e influenciou diretamente a consolidação do gênero (Silva, 2014).

O funcionamento do D&D baseia-se na interpretação de personagens fictícios que evoluem ao longo de campanhas por meio de níveis, classes e atributos. As ações dos personagens são resolvidas por meio de rolagens de dados poliédricos, principalmente o dado de vinte faces (d20), associadas a modificadores e regras específicas. O sistema prioriza um equilíbrio entre liberdade narrativa e controle mecânico, oferecendo suporte detalhado para combate, exploração e interação social.

A ficha de personagem no D&D apresenta uma estrutura extensa e padronizada, contendo atributos, perícias, habilidades, magias, equipamentos e pontos de vida. Essa complexidade torna o sistema robusto e flexível, porém exige maior organização por parte dos jogadores. Essa característica evidencia a necessidade de ferramentas digitais que auxiliem no gerenciamento e na atualização dinâmica dessas informações durante o jogo.

Entre os diferentes sistemas desenvolvidos para sustentar a experiência do RPG de mesa, destaca-se o *Old Quest*, que será detalhado a seguir.

2.3.2. Old Quest

O *Old Quest* é um sistema brasileiro de RPG de mesa criado por Bruno Moraes Alves (Alves, 2022) e lançado em 2022. Sua proposta central é a simplificação das regras, com foco na narrativa e na acessibilidade, tornando-o adequado para jogadores iniciantes ou para grupos que buscam partidas rápidas e menos dependentes de mecânicas complexas.

O sistema utiliza exclusivamente um dado de seis faces e apresenta regras condensadas, reduzindo significativamente a quantidade de cálculos e exceções. Essa abordagem minimiza a dependência de consultas constantes ao livro de regras e prioriza a fluidez da sessão e a improvisação narrativa.

A ficha de personagem do *Old Quest* reflete essa simplicidade, reunindo apenas os atributos essenciais e informações básicas do personagem. A estrutura enxuta facilita o preenchimento manual e reduz a necessidade de automatizações complexas, contrastando diretamente com sistemas mais detalhados como o D&D. Essa diferença evidencia a diversidade estrutural entre sistemas de RPG e reforça a necessidade de soluções flexíveis

capazes de acomodar modelos distintos de fichas.

Neste sentido, pode-se observar que a área de jogos digitais, em especial a área de jogos de RPG apresentam diferentes propostas. Desta forma, realizou-se uma busca na literatura para compreender os principais trabalhos relacionados, sendo estes apresentados na sequência.

2.4. Trabalhos Relacionados

Diversas ferramentas digitais foram desenvolvidas com o objetivo de facilitar a experiência do RPG de mesa em ambientes online. Algumas delas se destacam por sua ampla adoção pela comunidade, enquanto outras são propostas acadêmicas ou soluções voltadas a nichos específicos. Nesta seção, analisam-se quatro ferramentas amplamente utilizadas: *Roll20*, *Foundry Virtual Tabletop (FoundryVTT)*, *D&D Beyond* e *Lorde dos Dados*, com foco em como lidam com a criação e o gerenciamento de fichas de personagem. A análise considera a facilidade de uso, a flexibilidade do sistema de fichas e a compatibilidade com múltiplos sistemas de RPG.

2.4.1. Roll20

O *Roll20* é uma das plataformas mais populares para jogos de *RPG online*, oferecendo uma interface baseada em navegador que permite a criação de mapas, rolagens automáticas de dados, uso de *tokens* e integração com vídeo e áudio. Lançado em 2012, o sistema se popularizou por sua acessibilidade, permitindo que jogadores se conectem rapidamente e conduzam partidas à distância. A plataforma oferece suporte nativo a diversos sistemas populares, como *Dungeons & Dragons*, *Pathfinder* e *Call of Cthulhu*, por meio de folhas de personagem pré-configuradas (Roll20, 2024a).

No entanto, o sistema de fichas do *Roll20* é relativamente fechado para usuários comuns. A personalização de fichas exige o uso de HTML, CSS e JavaScript, conhecimento técnico que nem todos os mestres ou jogadores possuem. Além disso, muitas funcionalidades avançadas estão disponíveis apenas para usuários pagantes, o que limita o acesso a ferramentas mais robustas. Essa combinação de fatores torna a plataforma eficiente para sistemas tradicionais, mas pouco prática para grupos que desejam utilizar sistemas próprios ou pouco convencionais, restringindo sua flexibilidade.

2.4.2. Foundry Virtual Tabletop (FoundryVTT)

O *FoundryVTT* é uma aplicação independente que oferece uma das experiências mais completas e personalizáveis entre as mesas virtuais digitais. Diferente do *Roll20*, ele precisa ser instalado localmente ou em um servidor, o que dá maior liberdade de configuração, mas também exige maior conhecimento técnico por parte do usuário. O sistema é voltado para narradores e jogadores que desejam controle total sobre sua mesa virtual, desde a criação de mapas até a automação de regras (Foundry Gaming, LLC, 2024b).

O sistema de fichas no *FoundryVTT* é altamente adaptável, graças à sua arquitetura modular. É possível instalar sistemas prontos, desenvolvidos pela comunidade, ou

criar novos sistemas do zero utilizando JavaScript e JSON. Essa flexibilidade, no entanto, impõe uma barreira de entrada significativa, já que a criação de um sistema próprio ou a customização profunda requer familiaridade com programação. Além disso, o *Foundry* é um software pago, e não possui versão gratuita funcional, o que pode limitar sua adoção por parte de jogadores casuais ou iniciantes.

2.4.3. D&D Beyond

O *D&D Beyond* é a plataforma oficial licenciada pela *Wizards of the Coast* para suporte ao sistema *Dungeons & Dragons* 5ª edição. Lançado em 2017, o site centraliza informações de regras, suplementos, aventuras e gerenciamento de personagens em uma interface amigável e de fácil acesso (Wizards of the Coast, 2024). Sua proposta é simplificar o acesso ao conteúdo oficial, oferecendo uma experiência intuitiva para jogadores de todos os níveis.

O sistema de fichas do *D&D Beyond* é altamente automatizado e interativo, permitindo que o usuário crie personagens por meio de um assistente passo a passo. Durante a criação, o sistema já aplica os bônus e restrições de cada classe, raça e antecedente, além de calcular automaticamente perícias, modificadores e outras estatísticas. No entanto, sua grande limitação é a rigidez: ele é exclusivo para D&D 5e, não oferecendo suporte oficial a outros sistemas. A adaptação de regras ou a inclusão de sistemas personalizados não é possível sem recorrer a soluções externas. Apesar da usabilidade elevada, o *D&D Beyond* é um ambiente fechado, adequado apenas aos jogadores da quinta edição.

2.4.4. Lorde dos Dados

O Lorde dos Dados é um projeto acadêmico desenvolvido por alunos do ensino técnico com o objetivo de criar uma plataforma digital para gerenciamento de fichas e aventuras de RPG (Shibata et al., 2023). O sistema foi projetado para automatizar interações complexas, como cálculos de dano, testes de atributos e controle de pontos de vida, tornando a experiência de jogo mais fluida.

Com foco nos sistemas *Tormenta20* e *Ordem Paranormal*, a aplicação apresenta recursos visuais e organizacionais, como fichas interativas, gerenciamento de salas privadas, rolagem de dados integrada e suporte a elementos gráficos como mapas e *tokens*. A ficha do jogador no sistema é estruturada para conter atributos, perícias, equipamentos e habilidades específicas, além de permitir visualização dinâmica e atualizações em tempo real durante as sessões. Apesar de ser um sistema ainda em desenvolvimento e voltado inicialmente para dois sistemas específicos, o projeto busca oferecer flexibilidade para futuras expansões e apresenta potencial como base para soluções mais amplas.

Assim, o Quadro 1 apresenta uma síntese das principais características dos sistemas descritos previamente, considerando critérios como código aberto, compatibilidade com múltiplos sistemas, facilidade de uso e recursos de edição de fichas. Também foi incluído o projeto proposto neste trabalho como referência para contraste.

A análise comparativa evidencia que as ferramentas existentes atendem a recortes específicos do universo de RPG, mas não contemplam de forma abrangente a diversi-

Sistema	Código aberto	Multi-sistema	Facilidade de uso	Personalizável	Observações
Roll20	Não	Limitado	Média	Sim (limitada)	Personalização requer plano pago e conhecimento técnico.
FoundryVTT	Sim	Parcial	Baixa	Sim (via código)	Altamente flexível, mas com curva de aprendizado elevada.
D&D Beyond	Não	Não	Alta	Não	Exclusivo para D&D 5e; fechado e dependente de conteúdo pago.
Lorde dos Dados	Sim	Não	Média	Sim	Projeto acadêmico com ênfase em automatização e visual.
(Sistema Proposto)	Sim	Sim	Alta	Sim	Flexível, com foco em múltiplos sistemas e interface dinâmica.

Quadro 1: Comparativo entre sistemas e plataformas relacionadas

dade de sistemas e estilos de jogo. Soluções como o *Roll20* e o *D&D Beyond* priorizam sistemas populares ou funcionalidades específicas, enquanto plataformas mais flexíveis, como o *Foundry VTT*, exigem conhecimentos técnicos avançados para personalização e manutenção.

Essa fragmentação decorre, em grande parte, das diferenças estruturais entre os sistemas de RPG, que variam significativamente quanto à complexidade das regras, à quantidade de atributos e à forma de evolução dos personagens. Enquanto alguns sistemas adotam fichas extensas e altamente mecanizadas, outros privilegiam modelos simplificados e narrativos, dificultando a adoção de soluções rígidas ou padronizadas.

Diante desse cenário, observa-se a necessidade de uma solução dinâmica e unificada, capaz de adaptar-se a diferentes modelos de ficha sem impor limitações técnicas ao usuário final. Essa lacuna fundamenta a proposta deste trabalho, que busca oferecer uma aplicação flexível para o gerenciamento de fichas de RPG independentes de sistema.

3. Modelagem

Esta seção descreve as decisões técnicas adotadas no desenvolvimento do sistema, abordando as tecnologias selecionadas, os critérios para sua escolha e a forma como elas contribuem para os objetivos do projeto. Também são apresentados os métodos de modelagem utilizados na definição da arquitetura, estrutura dos dados e a organização das interfaces. O objetivo é garantir que o sistema seja escalável, modular e aderente às boas práticas de engenharia de software.

3.1. Tecnologias

A implementação do sistema foi planejada com foco em modularidade, desempenho e facilidade de manutenção. Para isso, foram selecionadas tecnologias modernas que atendem aos requisitos do projeto tanto no *back-end* quanto no *front-end*. A seguir, são descritas as principais ferramentas e *frameworks* adotados.

3.1.1. Plataforma .NET e Clean Architecture

O *back-end* do sistema foi implementado utilizando a linguagem C# com a plataforma .NET, que oferece alto desempenho, ampla documentação e suporte robusto para aplicações Web. A escolha dessa tecnologia se baseia na familiaridade com o ecossistema

e na sua capacidade de escalar horizontalmente sem comprometer a organização interna do projeto.

A arquitetura adotada foi a *Clean Architecture*, que propõe a separação entre regras de negócio, lógica de aplicação e detalhes de infraestrutura. Essa separação facilita a testabilidade, manutenção e evolução do sistema ao longo do tempo, permitindo que novas funcionalidades sejam adicionadas sem gerar acoplamentos indesejados entre os módulos. A camada de domínio é responsável por conter as regras centrais da aplicação, enquanto a camada de infraestrutura abriga a comunicação com o banco de dados e demais serviços externos.

No contexto deste projeto, essa abordagem se mostra especialmente vantajosa, pois o sistema foi concebido para aceitar múltiplos *templates* de sistemas de RPG definidos em JSON. Esses *templates* definem a estrutura de fichas de personagens, permitindo que o *back-end* armazene e manipule dados sem depender de esquemas rígidos. Com a *Clean Architecture*, é possível manter o controle sobre as regras que validam esses dados, mesmo com a flexibilidade da estrutura de entrada.

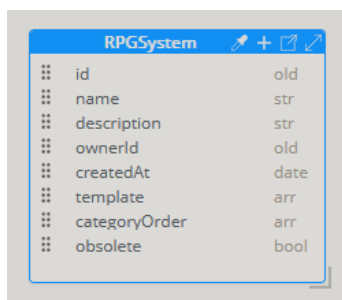
Além disso, a modularidade da arquitetura facilita a criação de *endpoints* genéricos, capazes de receber, validar e persistir diferentes formatos de ficha, mantendo o código reutilizável e adaptável às necessidades de cada sistema criado ou importado pelo usuário.

3.1.2. MongoDB

O *MongoDB* foi utilizado como banco de dados principal do sistema. Por ser uma base NoSQL orientada a documentos, ele permite armazenar estruturas em formato JSON de forma flexível, sem a necessidade de um esquema fixo. Essa característica torna o *MongoDB* especialmente adequado para aplicações que lidam com dados dinâmicos e variados, como fichas personalizadas de personagens.

No contexto do projeto, cada ficha de personagem será salva como um documento, respeitando a estrutura definida pelo template do sistema de RPG correspondente. Como diferentes sistemas possuem atributos e campos distintos, a ausência de um esquema rígido evita a necessidade de alterações frequentes na modelagem do banco de dados, facilitando a escalabilidade e a manutenção.

A coleção **RPGSystem** armazena os templates dos sistemas de RPG, definindo os componentes, categorias e atributos que serão utilizados na criação das fichas, Figura 1.



Field	Type
id	old
name	str
description	str
ownerId	old
createdAt	date
template	arr
categoryOrder	arr
obsolete	bool

Figura 1. Modelagem da coleção de sistemas

Como pode ser visualizado nas Figuras 1 e 2 o banco contará com as seguintes coleções: **RPGSystems**, **Characters**, **Campaigns** e **Users**.

Collection	Field	Type
Characters	id	old
	systemId	old
	userId	old
	createdAt	date
	name	str
	data	doc
RollConfig	enabled	bool
	formula	str
CharacterData	name	str
	value	str
	rollable	doc
	expression	str
	editable	bool
	edited	bool
	sessionEditable	bool
	visible	bool
	category	str
	component	str
	order	num
	options	arr

Figura 2. Modelagem da coleção de personagens

Ainda, as Figuras 3 e 4 apresentam as coleções **Campaign** e **User**. A coleção **Campaign** reúne os dados das campanhas em andamento, incluindo os personagens participantes, o mestre da sessão, os jogadores, além do histórico de mensagens e rolagens de dados por meio do campo de chat integrado. Por fim, a coleção **User** guarda informações básicas como identificador de autenticação, nome, e-mail e referências a sistemas e campanhas, utilizadas para controle de acesso e personalização da experiência do usuário.

Collection	Field	Type
Campaign	id	old
	title	str
	description	str
	systemId	old
	masterId	old
	playerIds	arr
	createdAt	date
	activeSession	bool
	characters	arr
	diceHistory	doc
CampaignCharacter	characterId	old
	playerId	old
	dynamicData	doc
DynamicFields	name	str
	value	str
	edited	str
	expression	str
DiceRoll	rolledBy	old
	expression	str
	result	str
	individualRolls	arr
	timestamp	date

Figura 3. Modelagem da coleção de campanhas

User		
⋮	authId	old
⋮	displayName	str
⋮	createdAt	str
⋮	savedSystemIds	arr
⋮	masterCampaignIds	arr
⋮	playerCampaignIds	arr

Figura 4. Modelagem da coleção de usuários

3.1.3. Angular

O *front-end* do sistema foi desenvolvido com o *framework* Angular, amplamente utilizado para aplicações Web. Sua arquitetura baseada em componentes permite a construção de interfaces modulares e reutilizáveis, além de oferecer suporte nativo a formulários reativos, injeção de dependência, roteamento e integração com APIs REST.

No projeto, o Angular foi essencial para garantir que a interface se adapte dinamicamente ao conteúdo definido nos *templates* JSON dos sistemas de RPG. Cada campo da ficha é renderizado de acordo com os metadados definidos no template, como tipo de dado, permissões de edição, visibilidade e possibilidade de rolagem. Isso exige que a aplicação seja capaz de interpretar e construir formulários completos em tempo de execução, respeitando as variações entre diferentes sistemas de RPG.

Além disso, o uso de Angular facilitará a manutenção e expansão da aplicação, permitindo que novos componentes visuais sejam integrados à medida que novas categorias de campo forem adicionadas aos *templates* no futuro.

3.1.4. Controle de Versão e Gerenciamento de Tarefas

O projeto utilizará o sistema de controle de versão Git, amplamente adotado no desenvolvimento de software por sua confiabilidade e capacidade de rastrear mudanças de forma detalhada. O repositório está hospedado no GitHub, que oferece uma interface amigável para o acompanhamento do histórico de *commits*, controle de *branches* e colaboração.

Para o gerenciamento das tarefas e organização do fluxo de desenvolvimento, foi utilizado o *GitHub Projects*. A ferramenta permite a criação de quadros com colunas e cartões, facilitando a visualização do progresso das funcionalidades, mesmo em projetos individuais. Cada etapa foi registrada como uma tarefa, organizada por prioridade e complexidade, promovendo uma abordagem incremental no desenvolvimento do sistema.

Como o desenvolvimento foi realizado por apenas um autor/desenvolvedor, não foi adotada uma metodologia ágil formal como *Scrum*. Ainda assim, a organização incremental das tarefas no *GitHub Projects* permitirá entregas frequentes, controle de progresso e facilidade de revisão, mesmo sem ciclos estruturados.

3.2. Requisitos Funcionais e Não Funcionais

Nesta subseção, são definidos os requisitos funcionais e não funcionais do sistema proposto, organizados em categorias que representam o comportamento esperado da aplicação e os critérios de qualidade que ela deve atender. Os requisitos foram definidos com base na estrutura planejada do sistema, considerando aspectos como usabilidade, segurança, escalabilidade e flexibilidade de uso.

3.2.1. Requisitos Funcionais

- **RF01** – O sistema deve permitir que o usuário crie uma conta, autentique-se, recupere a senha e realize a exclusão lógica da conta.
- **RF02** – O sistema deve permitir a criação, edição e exclusão de personagens com base em fichas dinâmicas definidas por *templates* JSON.
- **RF03** – O sistema deve permitir que o usuário publique novos sistemas de RPG em um repositório público.
- **RF04** - Caso um sistema publicado seja excluído por seu criador, uma cópia será automaticamente gerada para cada usuário que o tenha salvo previamente. Essa cópia será marcada como obsoleta e poderá ser mantida pelos usuários de forma independente.
- **RF05** – O sistema deve validar se o template de sistema está em conformidade com a estrutura esperada.
- **RF06** – O sistema deve permitir a criação de campanhas, assim como a edição e exclusão delas.
- **RF07** – O sistema deve permitir o vínculo de personagens a campanhas, respeitando o sistema utilizado.
- **RF08** – O sistema deve permitir que os jogadores visualizem fichas de personagens dentro da campanha.
- **RF09** – O sistema deve permitir a criação e encerramento de sessões, armazenando o estado dinâmico das fichas.
- **RF10** – O sistema deve permitir que jogadores editem campos dinâmicos das fichas durante sessões ativas.
- **RF11** – O sistema deve incluir um histórico de rolagens de dados vinculado às sessões.

3.2.2. Requisitos Não Funcionais

- **RNF01** – O sistema deve renderizar dinamicamente os componentes da interface com base nos *templates* JSON dos sistemas de RPG.
- **RNF02** – O sistema deve ser responsivo e acessível em diferentes dispositivos e tamanhos de tela.
- **RNF03** – O sistema deve apresentar desempenho estável mesmo com grandes volumes de dados dinâmicos.
- **RNF04** – O sistema deve seguir princípios de arquitetura modular, separando claramente a lógica de negócio, apresentação e persistência.
- **RNF05** – O sistema deve utilizar controle de versão com Git e repositório remoto no GitHub.

- **RNF06** – O sistema deve ser de fácil manutenção e permitir evolução incremental de suas funcionalidades.
- **RNF07** – O sistema deve apresentar interface intuitiva, permitindo que novos usuários compreendam suas principais funções sem necessidade de treinamento prévio.

Com os requisitos estabelecidos, a próxima etapa consiste na modelagem dos casos de uso, representando graficamente as interações entre os usuários e os principais serviços oferecidos pelo sistema. O objetivo é consolidar a lógica funcional em uma visão integrada e de fácil interpretação.

3.3. Casos de Uso do Sistema

Para compreender o fluxo geral da aplicação, foram elaborados dois diagramas de casos de uso que representam as principais interações dos usuários com o sistema. O objetivo é oferecer uma visão clara das funcionalidades implementadas e da divisão de responsabilidades entre jogadores, mestres e o sistema em si.

No modelo adotado, o fluxo de criação, seleção e uso de fichas está diretamente conectado às campanhas. O sistema permite que jogadores editem fichas com base em *templates* dinâmicos, enquanto mestres podem visualizar, controlar sessões e registrar estados alterados das fichas durante o jogo. O armazenamento e a lógica de permissões são definidos de acordo com os campos *editable* e *sessionEditable*, permitindo separar a edição normal da edição em tempo de sessão.

Além disso, os dados manipulados na aplicação seguem um modelo padronizado baseado em estruturas JSON. Essa abordagem permite a criação de *templates* genéricos e reutilizáveis, que definem a estrutura lógica de cada ficha, conforme demonstrado na próxima subseção.

A Figura 5 apresenta as interações do usuário com a aplicação em um nível geral.

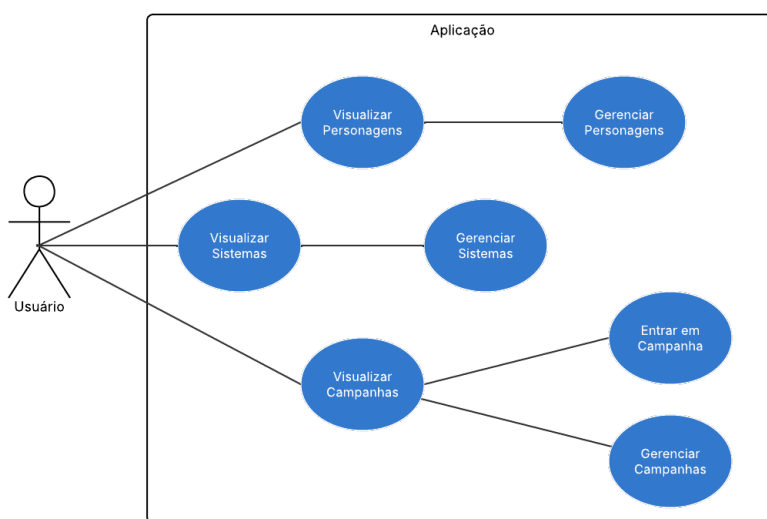


Figura 5. Diagrama de Caso de Uso do Usuário

A Figura 6 detalha as ações específicas realizadas por jogadores e mestres.

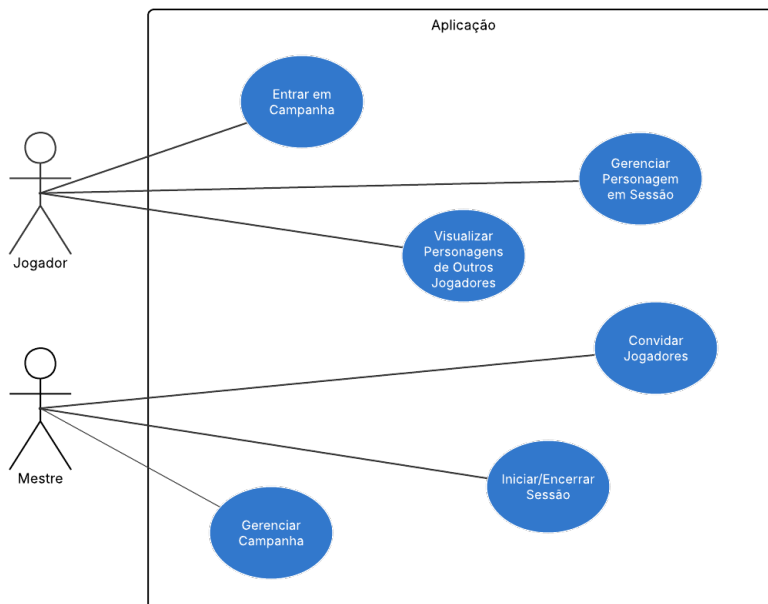


Figura 6. Diagrama de Caso de Uso de Jogadores e Mestres

3.3.1. Estrutura do Template JSON

A estrutura das fichas é modelada com base em um *template* JSON, definido por sistema. Cada *template* contém os campos que devem compor a ficha, com metadados que controlam sua renderização e comportamento na interface. Essa estrutura torna a aplicação adaptável a diferentes sistemas de RPG, independentemente da complexidade ou quantidade de atributos utilizados. Os campos mais relevantes são:

- **name:** Nome do campo visível na interface
- **value:** Conteúdo atribuído ao campo
- **expression:** Expressão que define o valor do campo, utilizada para atualizar automaticamente a propriedade *value*
- **component:** Tipo de componente renderizado
- **editable:** Define se o jogador pode editar esse campo
- **sessionEditable:** Define se o campo pode ser alterado durante uma sessão ativa
- **edited:** Indica se o campo foi alterado manualmente pelo jogador, podendo exigir reset para reaplicar a *expression*
- **rollable:** Indica se o campo pode ser usado em rolagens de dados e qual fórmula deve ser aplicada
- **visible:** Controla a visibilidade do campo
- **category:** Agrupador visual de campos dentro da ficha
- **order:** Define a posição do campo dentro da sua categoria
- **options:** Lista de valores pré-definidos disponíveis para o campo

Esses campos são agrupados em um *array* dentro da ficha do personagem e validados conforme o sistema correspondente, garantindo que todos os dados estejam conforme o sistema selecionado.

3.4. Interface do Sistema

Com o objetivo ter uma organização visual da aplicação e validar a experiência do usuário, foram desenvolvidos protótipos de tela e após estas foram desenvolvidas. As interfaces foram divididas em três seções funcionais principais: personagens, sistemas e campanhas.

- **Área de Personagens:** Exibe uma galeria com os personagens do usuário, como demonstrado na Figura 7, apresentando o nome e o sistema associado. Ao clicar em um personagem, o usuário é direcionado para a tela de edição, onde os campos são renderizados dinamicamente com base no template correspondente.

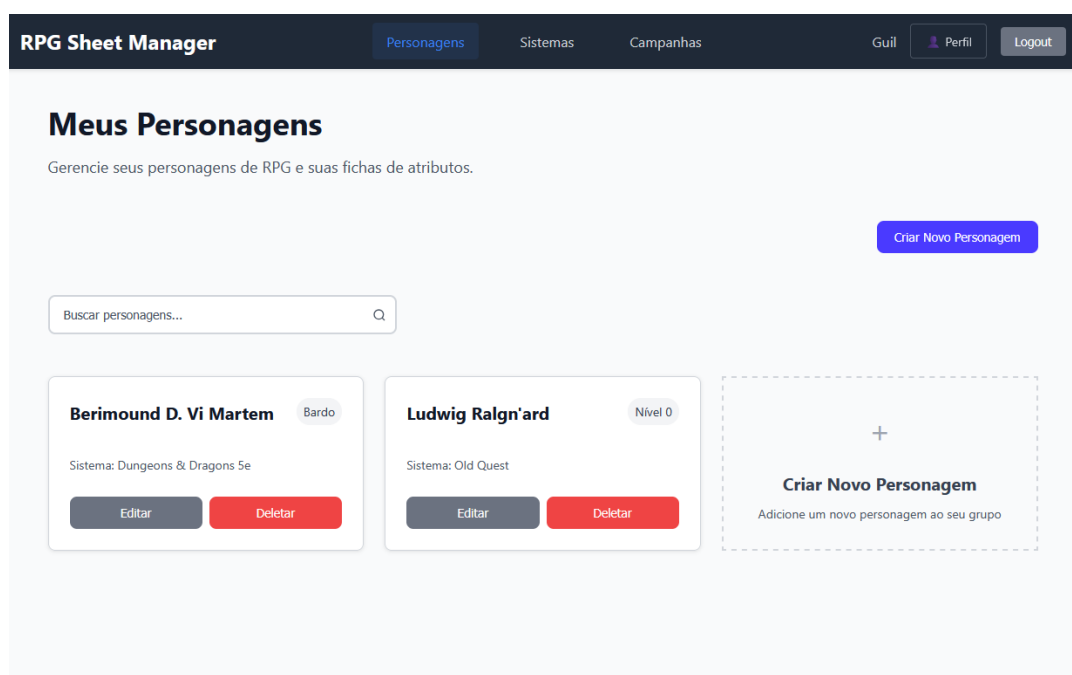


Figura 7. Galeria de personagens

- **Área de Sistemas:** A Figura 8 apresenta os sistemas disponíveis, exibindo a imagem, o nome e a descrição. Nessa interface, é possível realizar a adição de novos sistemas ou salvar sistemas públicos na sua conta.

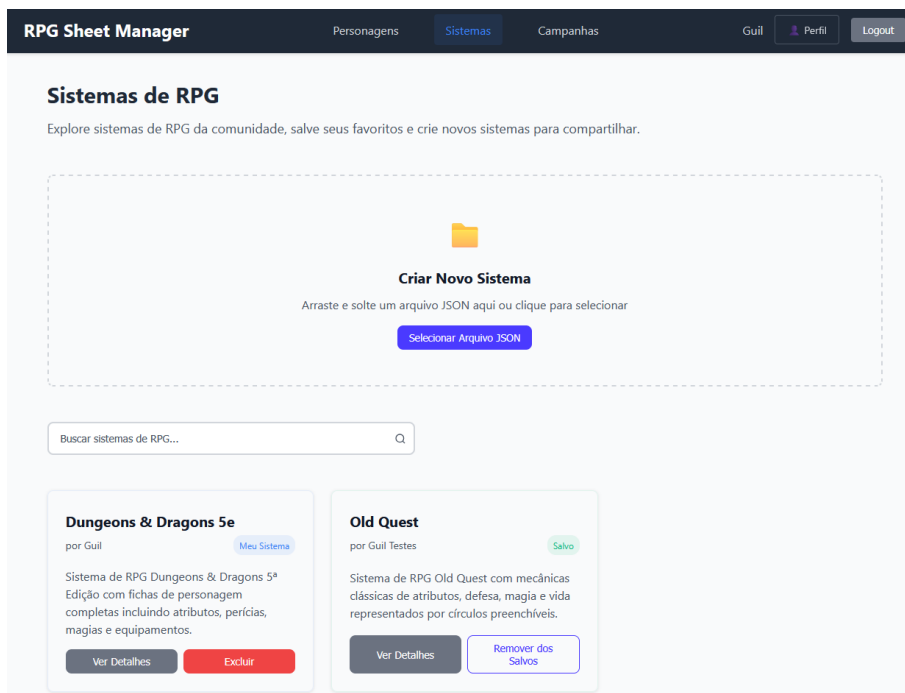


Figura 8. Visualização de sistemas

- **Área de Campanhas:** Por fim, na Figura 9 apresenta-se uma lista das campanhas criadas ou vinculadas ao usuário. Cada item mostra o nome da campanha, os jogadores ativos e o status.

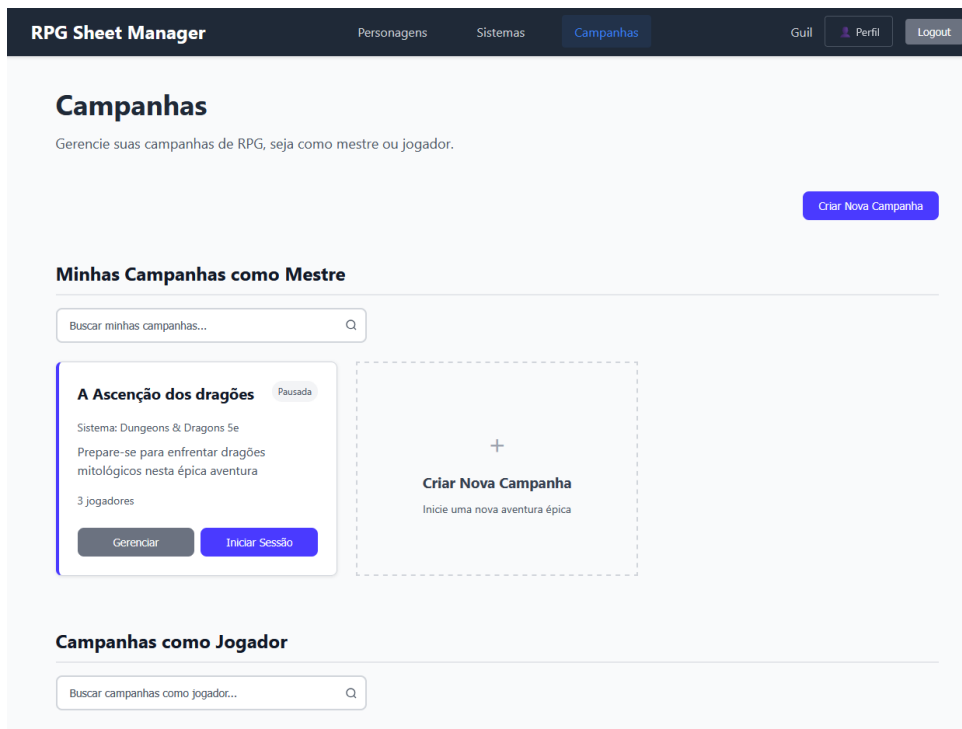


Figura 9. Listagem de campanhas do usuário

Os campos da ficha são interpretados dinamicamente com base nas propriedades definidas no *template*, como tipo de componente, permissões de edição e agrupamento por categoria. Essa abordagem garante flexibilidade e consistência entre sistemas distintos.

4. Desenvolvimento

Essa seção apresenta o processo de desenvolvimento do projeto, fundamentado nas decisões técnicas definidas nas etapas anteriores. O sistema foi concebido como um **MVP**, com foco na demonstração da viabilidade técnica e estrutural da aplicação proposta, sem a execução de testes de desempenho ou validações formais de requisitos. O principal objetivo desta fase foi comprovar a integração entre o *back-end* e o *front-end*, estabelecendo a base operacional mínima do sistema.

O desenvolvimento teve início pela construção da API no *back-end*, responsável por estruturar a comunicação e as funcionalidades centrais da aplicação. Após a conclusão dessa etapa, foi desenvolvido o *front-end*, voltado à interface e à interação do usuário com o sistema.

Para a organização e o acompanhamento das etapas de desenvolvimento, utilizou-se o recurso *GitHub Projects*, que permitiu o controle das atividades por meio da criação de *issues*. Cada *issue* representava uma tarefa específica a ser executada, possibilitando o registro do progresso e a priorização das entregas. Essa abordagem contribuiu para facilitar o planejamento e o acompanhamento da evolução do projeto.

4.1. back-end

O *back-end* foi desenvolvido com base nos princípios da *Clean Architecture*, que estabelece a separação de responsabilidades e a independência entre as camadas do sistema. Essa estrutura visa garantir a facilidade de manutenção, escalabilidade e testabilidade do código. O projeto foi organizado em quatro camadas principais: domínio, aplicação, infraestrutura e entrega, conforme será apresentado nas Figuras 10 a 13.

4.1.1. Camada de Domínio

A camada de domínio constitui o núcleo lógico do sistema, responsável pela definição das entidades, propriedades e regras de negócio. Também contém as interfaces que abstraem os repositórios de dados, assegurando que o domínio permaneça independente da tecnologia de persistência utilizada. Essa organização promove uma comunicação clara entre os componentes e garante a integridade da lógica central do sistema.

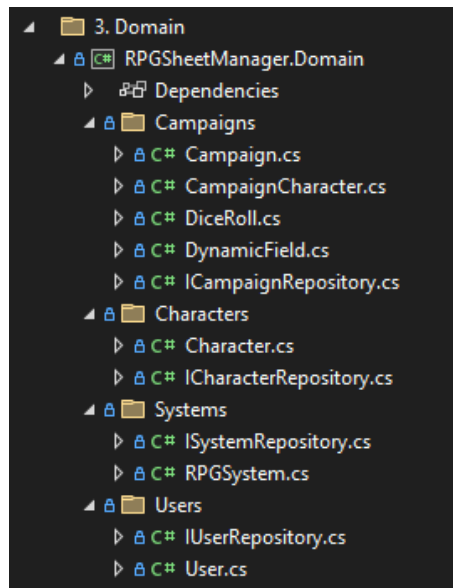


Figura 10. Estrutura da camada de Domínio

4.1.2. Camada de Aplicação

A camada de aplicação coordena a execução das regras de negócio definidas no domínio, orquestrando os casos de uso e integrando-os às funcionalidades disponibilizadas pela infraestrutura. Seu papel é conectar a lógica interna do sistema às operações executadas pelas camadas externas, sem conter detalhes de apresentação ou persistência.

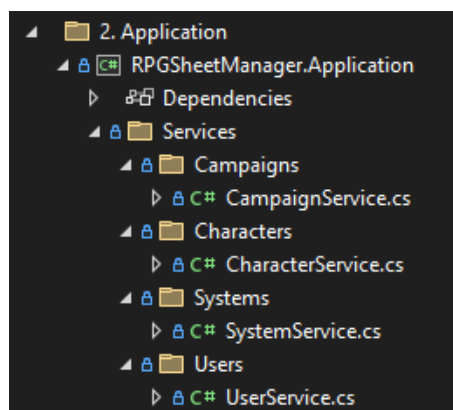


Figura 11. Estrutura da camada de Aplicação

4.1.3. Camada de Infraestrutura

A camada de infraestrutura implementa os mecanismos técnicos de suporte ao sistema, como a persistência de dados e a comunicação com o banco *MongoDB*. Ela fornece as implementações concretas das interfaces do domínio, traduzindo as abstrações em operações executáveis no ambiente físico de execução.

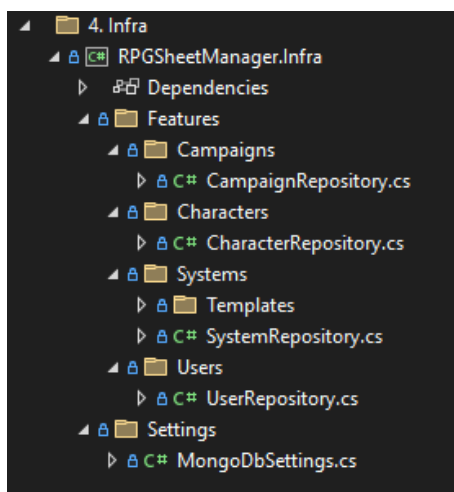


Figura 12. Estrutura da camada de Infraestrutura

4.1.4. Camada de Entrega

A camada de entrega disponibiliza as funcionalidades do sistema por meio de uma *Web API*, servindo como ponto de entrada para as requisições externas. Nela estão definidos os controladores e *endpoints* responsáveis por validar dados, acionar casos de uso e retornar respostas padronizadas em protocolos como HTTP. Essa camada integra o *back-end* ao restante da aplicação, mantendo a separação entre lógica de negócio e exposição de serviços.

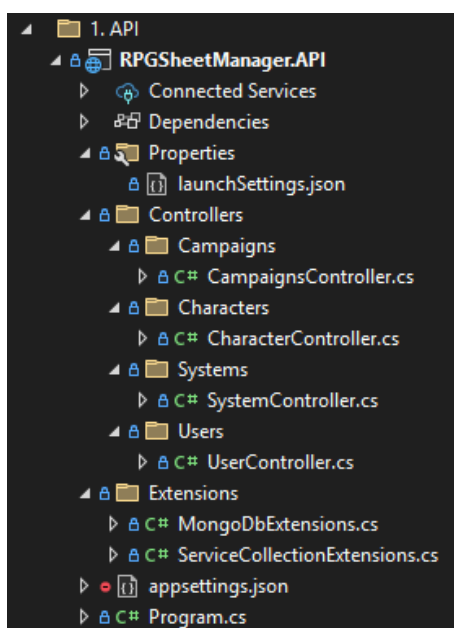


Figura 13. Estrutura da camada de Entrega

4.2. front-end

O desenvolvimento do *front-end* teve início com a criação do projeto utilizando o *framework* Angular. A estrutura inicial concentrou-se na definição das rotas e na organização das páginas principais, implementando o mecanismo de roteamento dinâmico oferecido pelo *framework*. Essa abordagem permite a navegação entre as seções da aplicação sem a necessidade de recarregar toda a página, proporcionando maior fluidez e desempenho durante o uso do sistema.

Foi implementado um componente de barra de navegação fixa, responsável por permitir a transição entre as diferentes áreas da aplicação. Essa barra permanece ativa em todas as telas, garantindo consistência visual e facilitando o acesso às funcionalidades principais, como o gerenciamento de personagens, sistemas e campanhas.

4.2.1. Layout e autenticação

A autenticação dos usuários é realizada por meio da plataforma *Auth0*, que assegura o acesso controlado às rotas protegidas da aplicação. O login é requisito para utilização do sistema, sendo possível autenticar-se por meio de uma conta própria criada na plataforma ou utilizando credenciais do *Google*.

A Figura 14 ilustra a interface de autenticação disponibilizada pelo *Auth0*, na qual o usuário pode realizar o login utilizando suas credenciais ou por meio de provedores externos, como o *Google*. Após a autenticação bem-sucedida, o acesso é liberado para todas as funcionalidades internas da aplicação, permitindo também a visualização do perfil e o logout de forma segura a partir da barra de navegação.

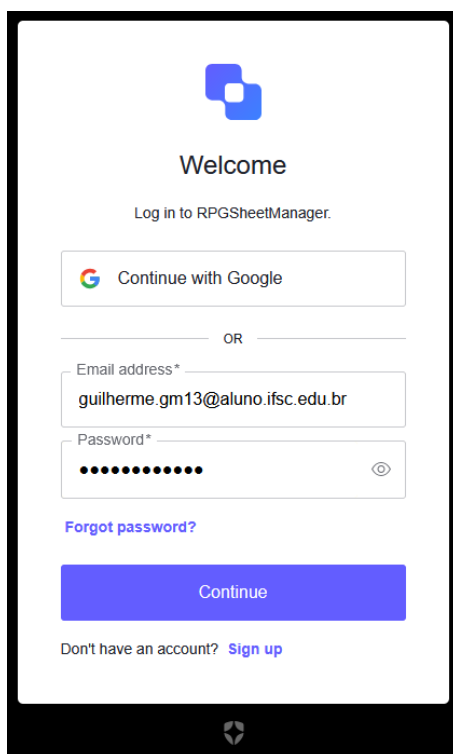


Figura 14. Interface de autenticação do sistema via *Auth0*

4.2.2. Tela de Personagens

A tela de personagens, conforme demonstrado na Seção 3.4, apresenta a listagem dos personagens criados pelo usuário. Cada personagem é exibido em um *card* individual, que contém informações resumidas e dois botões de ação: *Editar* e *Deletar*. Essa estrutura visa facilitar o gerenciamento de múltiplas fichas, permitindo o acesso rápido às funções mais utilizadas.

Ao selecionar a opção de criação de um novo personagem, o usuário é direcionado a um formulário onde deve informar o nome e o sistema de RPG associado. Após a seleção do sistema, os dados referentes aos atributos e características do personagem são carregados automaticamente, conforme as regras e configurações definidas para aquele sistema específico.

Em sistemas com interdependência entre atributos, como no caso do *Dungeons & Dragons 5e*, determinadas propriedades são calculadas dinamicamente a partir de expressões pré-definidas no modelo de dados. Esse mecanismo, representado pela propriedade *expression* no arquivo JSON do sistema, permite que atributos derivados sejam atualizados de forma automática conforme os valores base são alterados. Além disso, atributos que envolvem rolagens de dados apresentam uma funcionalidade adicional que possibilita simular os resultados diretamente na interface, oferecendo uma experiência mais interativa ao jogador.

A imagem mostra a interface de usuário da aplicação 'RPG Sheet Manager' na aba 'Personagens'. O título da página é 'Editar Personagem' com o subtítulo 'Modifique os dados do seu personagem'. O formulário contém os seguintes campos:

- Informações Básicas:**
 - Nome do Personagem *: Berimound D. Vi Martem
 - Sistema de RPG *: Dungeons & Dragons 5e
- Informações Básicas (seção secundária):**
 - Classe: Bardo
 - Level: 12
 - Raça: Humano
 - Antecedente: Artista
 - Alinhamento: Caótico e Bom

Figura 15. Tela de criação e edição de personagens na aplicação

4.2.3. Campos dinâmicos

Para reduzir duplicação e permitir a adaptação do formulário a diferentes sistemas, foi implementado um componente denominado *dynamic-fields*. Esse componente recebe como entrada um esquema em formato JSON que descreve os campos, seus tipos, validações e expressões derivadas. A partir desse esquema, o componente constrói um *FormGroup* reativo e instancia os controles correspondentes, renderizando dinamicamente os campos necessários na interface.

O *dynamic-fields* é utilizado nas telas de Personagens e de Sistemas, garantindo comportamento consistente entre ambas. Essa abordagem centraliza a lógica de construção dos formulários na camada de apresentação, otimiza a manutenção do código e permite incluir novos tipos de campo ou sistemas com regras distintas sem modificações significativas na estrutura do *front-end*.

4.2.4. Tela de Sistemas

A tela de sistemas de RPG tem como objetivo permitir o gerenciamento e a exploração dos sistemas disponíveis na aplicação. Nela, o usuário pode enviar, visualizar, salvar ou remover sistemas, além de importar novos modelos estruturados em formato JSON. A interface foi desenvolvida de modo a simplificar a interação com esses arquivos, centralizando as ações em um componente de seleção de arquivos que valida automaticamente o conteúdo enviado.

O processo de criação de novos sistemas é realizado por meio do campo de *upload* destacado na interface. Esse componente aceita exclusivamente arquivos no formato JSON, cuja estrutura deve seguir o modelo definido pela aplicação. Caso o arquivo não atenda ao padrão esperado, mensagens de erro são exibidas informando a inconsistência do formato. Os sistemas enviados são armazenados e disponibilizados para outros usuários, de modo que cada sistema pode ser reutilizado na criação de personagens e campanhas.

Os sistemas existentes são apresentados em uma listagem de *cards*, contendo informações resumidas, como nome, autor e descrição. A depender da relação do usuário com o sistema, diferentes opções são exibidas. Para sistemas criados pelo próprio usuário, é possível excluí-los, esta ação que marca o sistema como obsoleto e impede novos salvamentos. Já os sistemas criados por outros usuários podem ser salvos ou removidos da lista pessoal, adicionando ou retirando seu identificador da conta do usuário. Essa ação permite o acesso posterior ao sistema para criação de personagens ou campanhas baseadas nele.

O botão *Ver Detalhes* permite inspecionar a estrutura completa do sistema, exibindo os mesmos campos utilizados na criação de personagens. Nessa visualização, os campos permanecem bloqueados, servindo apenas como referência do modelo de atributos e regras definidos para o respectivo sistema.

4.2.5. Componentes de busca

Com o objetivo de padronizar a interação de pesquisa nas diferentes telas da aplicação, foi desenvolvido um componente reutilizável denominado *searchbar*. Esse componente é responsável por filtrar registros localmente ou por meio de consultas à API, conforme a configuração de cada módulo. Sua estrutura leve e genérica permite integração direta com listas de personagens, sistemas e campanhas, garantindo consistência visual e funcional em todo o *front-end*.

Uma variação desse componente, denominada *multi-searchbar*, foi empregada especificamente na criação e edição de campanhas. Essa versão estende as funcionalidades da *searchbar* tradicional, permitindo a seleção múltipla de usuários por meio de uma interface interativa. Dessa forma, o mestre da campanha pode convidar vários jogadores simultaneamente, visualizar os convites pendentes e remover participantes antes da confirmação do envio.

4.2.6. Campanhas

A seção de campanhas tem como objetivo permitir o gerenciamento das mesas de RPG do usuário, tanto na função de mestre quanto na de jogador. Nessa interface, o usuário pode criar, visualizar e administrar campanhas associadas aos sistemas previamente salvos na aplicação.

A tela inicial apresenta a divisão entre *Minhas Campanhas como Mestre* e *Campanhas como Jogador*, organizando as informações de acordo com o papel do usuário em cada grupo. Cada campanha é exibida em um *card* que contém o título, o sistema utilizado, o número de jogadores, o status atual e as opções de ação, como gerenciar ou iniciar uma sessão. Há também um *card* adicional destinado à criação de uma nova campanha.

No processo de criação, o usuário pode definir o nome, o sistema de RPG a ser utilizado, uma descrição opcional e selecionar os jogadores que serão convidados a participar. O campo de seleção de jogadores permite a busca de outros usuários registrados na aplicação, os quais podem aceitar ou recusar o convite posteriormente.

Ao acessar o gerenciamento de uma campanha, são exibidas suas informações gerais, como nome, sistema, mestre e status. Abaixo dessas informações são listados os jogadores ativos, acompanhados de seus nomes, endereços de e-mail e status de participação. O mestre possui ainda a opção de editar os dados da campanha, remover jogadores e encerrar a campanha permanentemente.

A tela de edição mantém a estrutura da criação, apresentando os dados previamente preenchidos e permitindo ajustes no nome, na descrição, no sistema e na lista de jogadores. Os participantes são organizados em duas categorias, sendo elas ativos e convidados. Essa organização possibilita um controle mais preciso da participação e o envio de novos convites.

4.2.7. Serviços e integração com a API

A comunicação entre o *front-end* e o *back-end* é mediada por classes de serviço, responsáveis por encapsular as operações executadas sobre os dados da aplicação. Cada módulo, como Personagens, Sistemas e Campanhas, possui um serviço próprio, implementado a partir do módulo *HttpClient* do *Angular*. Esses serviços interagem com os *endpoints* REST definidos no *back-end*, garantindo a separação entre a camada de apresentação e a lógica de acesso remoto.

Entre os serviços desenvolvidos, destaca-se o *CharacterService*, responsável pelo gerenciamento das fichas de personagem. Essa classe implementa métodos para criação, atualização, exclusão e recuperação de personagens, todos protegidos pelo mecanismo de autenticação descrito anteriormente. As chamadas HTTP são automatizadas pelo interceptador de autenticação, que insere o *token* de acesso nas requisições direcionadas à API. Esse procedimento assegura que apenas usuários autenticados possam realizar operações de escrita ou modificação de dados.

A Figura 16 apresenta um trecho do serviço de personagens, destacando o método de criação de fichas. Nessa operação, o serviço obtém as informações do usuário autenticado por meio do *Auth0*, associa o identificador do usuário ao objeto do personagem e envia o payload para o endpoint `/character`. Após o sucesso da requisição, a lista de personagens é atualizada localmente com o uso do padrão *BehaviorSubject*, o que garante a reatividade da interface e a atualização automática dos componentes que dependem desses dados.

```
public createCharacter(character: Character): Observable<Character> {
  return this.auth.user$.pipe(
    switchMap(user => {
      const characterWithUser = {
        ...character,
        userId: user?.sub
      };

      const characterToCreate = this.prepareCharacterForApi(characterWithUser);

      return this.http.post<Character>(this.apiUrl, characterToCreate).pipe(
        tap(createdCharacter => {
          const currentCharacters = this.charactersSubject.value;
          this.charactersSubject.next([...currentCharacters, createdCharacter]);
        })
      );
    })
  );
}
```

Figura 16. Trecho do serviço de personagens responsável pela criação de fichas

A Figura 17 ilustra a operação correspondente no *back-end*, responsável por persistir o personagem criado na base de dados. O método recebe o objeto enviado pelo *front-end*, insere-o na coleção e retorna o registro criado, completando o fluxo de comunicação entre as duas camadas.

```
public async Task<Character> AddAsync(Character character) {
    await _collection.InsertOneAsync(character);
    return character;
}
```

Figura 17. Método do back-end responsável pela inserção de personagens na base de dados

Esse fluxo exemplifica o padrão adotado em todas as operações do sistema, nas quais o *front-end* é responsável por enviar os dados processados pelos serviços específicos, enquanto o *back-end* realiza a persistência e o retorno das informações por meio de controladores e classes de serviço internas. Essa arquitetura garante a modularidade, o isolamento das responsabilidades e a facilidade de manutenção do sistema.

Além do `CharacterService`, outros serviços, como o `SystemService` e o `CampaignService`, seguem o mesmo princípio de comunicação, realizando operações de criação, atualização e consulta de forma segura e desacoplada. Ambos utilizam operadores reativos da biblioteca `RxJS`, como `switchMap`, `tap` e `catchError`, que permitem o encadeamento assíncrono de operações e o tratamento padronizado de erros, contribuindo para a legibilidade do código e a consistência entre os dados locais e remotos.

Além dos módulos principais, foram implementados serviços auxiliares, como o `UserService`, responsável pela busca e atualização de informações de usuários autenticados, e o `DiceService`, utilizado na simulação de rolagens de dados diretamente no navegador. Esses serviços complementam a camada de integração, ampliando a interatividade do sistema e reforçando sua modularidade.

Em conjunto, os serviços estabelecem um fluxo de comunicação seguro e desacoplado entre o *front-end* e o *back-end*, sustentando a arquitetura modular proposta e favorecendo futuras expansões da aplicação, como a integração de novos tipos de entidades ou funcionalidades colaborativas. A seguir serão apresentadas as dificuldades encontradas durante o desenvolvimento da aplicação.

4.3. Dificuldades e ajustes no desenvolvimento

Durante o processo de implementação, foram enfrentadas diversas dificuldades técnicas e de planejamento que demandaram ajustes nas estratégias inicialmente definidas. Um dos principais desafios esteve relacionado à integração do sistema de autenticação com o serviço `Auth0`. Inicialmente, foram desenvolvidos *endpoints* seguros no *back-end* para controle de autenticação; contudo, após testes e inconsistências observadas, tornou-se necessário reconfigurar o fluxo para que o `Auth0` fosse mantido integralmente no *front-end*. Esse processo exigiu adaptações significativas na comunicação com a API e nas políticas de `CORS`, que frequentemente geravam erros durante as requisições entre domínios.

Outro ponto relevante foi a necessidade de reestruturações no modelo de dados e na definição de entidades. Durante o desenvolvimento, verificou-se que a modelagem inicial não contemplava adequadamente todos os cenários de uso previstos. Nesse contexto, a adoção da *Clean Architecture* mostrou-se essencial, pois possibilitou a implementação das alterações sem comprometer a integridade das demais camadas do sistema, mantendo a consistência do código e a clareza na separação de responsabilidades.

Além disso, a funcionalidade de gerenciamento de campanhas revelou-se mais complexa do que o escopo original do projeto previa. Embora o objetivo principal fosse o gerenciamento de fichas e sistemas de RPG, optou-se por incluir uma versão simplificada do módulo de campanhas. Essa decisão, embora tenha agregado valor conceitual ao sistema, demandou tempo considerável e desviou parte do foco de melhorias que poderiam ter sido aplicadas às funcionalidades principais.

Durante as etapas iniciais, o controle das tarefas foi realizado por meio do recurso *GitHub Projects*, conforme ilustrado na Figura 18. Entretanto, por se tratar de um desenvolvimento individual, essa metodologia foi posteriormente descontinuada, visto que a manutenção do quadro consumia mais tempo do que efetivamente contribuía para o avanço das implementações. A partir desse ponto, o acompanhamento passou a ocorrer diretamente na *branch develop*, com registros locais e *commits* frequentes, priorizando a agilidade no fluxo de trabalho.

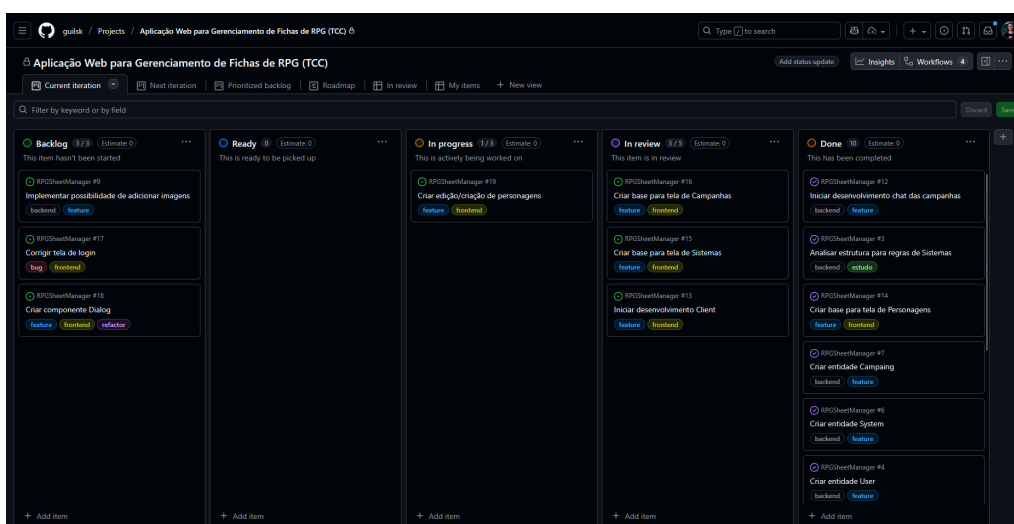


Figura 18. Quadro de tarefas utilizado nas etapas iniciais do projeto no *GitHub*

Concluídas as etapas de implementação e integração entre as camadas do sistema, tornou-se possível disponibilizar o MVP para testes com usuários reais. A seguir, são apresentados os procedimentos adotados na validação da aplicação e as percepções obtidas a partir das avaliações realizadas com jogadores de RPG.

5. Validação

Com a conclusão do desenvolvimento do protótipo mínimo viável (MVP), realizou-se uma etapa de validação com usuários finais, com o objetivo de avaliar a aceitação da aplicação e sua adequação ao problema proposto. A validação teve caráter exploratório e buscou analisar a percepção dos participantes quanto à utilidade, usabilidade e compreensão geral da ferramenta, considerando critérios de qualidade definidos pela norma ISO/IEC 25010, especialmente no que se refere à usabilidade e funcionalidade.

5.1. Objetivo da Validação

O objetivo da validação foi verificar se a aplicação desenvolvida atende às necessidades de jogadores e mestres de RPG de mesa no que diz respeito ao gerenciamento de fichas

de personagens de diferentes sistemas, bem como identificar percepções iniciais sobre facilidade de uso, clareza da proposta e potencial de adoção da ferramenta. Esta etapa não teve como finalidade a validação estatística ou o teste de desempenho do sistema, mas sim a obtenção de *feedback* qualitativo a partir do uso real do MVP.

5.2. Metodologia de Validação

A validação do sistema foi conduzida por meio de uma pesquisa aplicada, de caráter descritivo e abordagem qualitativa. O método adotado foi o teste de uso em ambiente real, seguido da coleta de percepções dos participantes por meio de um formulário eletrônico elaborado no *Google Forms*, disponibilizado juntamente com o acesso ao MVP da aplicação.

A coleta de dados ocorreu entre os dias 3 e 7 de novembro de 2025. O instrumento de pesquisa foi estruturado com base nos critérios de qualidade da norma ISO/IEC 25010 ISO/IEC (2025), contemplando aspectos relacionados à funcionalidade, usabilidade e satisfação geral do usuário. O formulário combinou questões fechadas em escala Likert de cinco pontos, utilizadas como apoio à análise qualitativa, e questões abertas destinadas ao registro de observações, dificuldades percebidas e sugestões de melhoria.

Os participantes foram convidados a utilizar livremente a aplicação antes do preenchimento do formulário, de modo a permitir uma avaliação baseada na experiência prática com o sistema. Ao todo, 13 usuários participaram da validação. As respostas consolidadas e os registros individuais encontram-se disponíveis nos anexos deste trabalho (Anexo B e Anexo A), assegurando a rastreabilidade dos dados coletados.

5.2.1. Perfil dos Participantes

Participaram da validação 13 jogadores de RPG de mesa, com idades entre 18 e 26 anos, sendo a maioria do gênero masculino (76,9%). Em relação à escolaridade, 69,2% dos participantes estavam cursando o ensino superior. O instrumento de coleta não contemplou a identificação da área de formação dos participantes.

Quanto à experiência com RPG de mesa, os participantes relataram entre um e oito anos de prática, atuando principalmente como jogadores, com alguns exercendo também o papel de mestres. Os sistemas citados incluem *Dungeons & Dragons*, *Tormenta20*, *Call of Cthulhu*, *Old Quest* e sistemas autorais. Esse perfil caracteriza um grupo com familiaridade prévia tanto com RPG quanto com ferramentas digitais, o que influenciou a forma como a aplicação foi avaliada.

5.3. Resultados da Validação

De modo geral, os resultados indicaram percepção positiva quanto à usabilidade e à aparência da aplicação Web. As respostas às questões fechadas sugerem que os participantes consideraram o sistema intuitivo e visualmente agradável, reconhecendo sua utilidade para centralizar o gerenciamento de fichas de personagens de diferentes sistemas de RPG.

As respostas abertas reforçam essa percepção, destacando a flexibilidade da proposta e o potencial da ferramenta para acomodar múltiplos sistemas. Contudo, foram identificadas dúvidas recorrentes quanto ao propósito da aplicação, especialmente entre

participantes que mencionaram a ausência de suporte nativo a sistemas específicos, como o *Tormenta20*. Essas observações indicam que parte dos usuários não compreendeu, de forma imediata, que a proposta central da aplicação é permitir a criação e publicação de sistemas personalizados pelos próprios usuários.

Além disso, alguns participantes relataram a falta de instruções iniciais ou de um guia de uso, apontando a necessidade de aprimorar a comunicação das funcionalidades disponíveis e o processo de *onboarding*. Esses aspectos evidenciam oportunidades de melhoria relacionadas à apresentação da proposta e à orientação do usuário.

Em síntese, os resultados da validação qualitativa indicam que o MVP alcançou seus objetivos iniciais, demonstrando aceitação positiva e aderência ao problema proposto, ao mesmo tempo em que os *feedbacks* obtidos fornecem subsídios para aprimoramentos futuros, especialmente no que se refere à clareza da proposta e à inclusão de mecanismos de apoio ao usuário.

6. Considerações Finais

Esta seção apresenta um resumo dos principais resultados alcançados com o desenvolvimento do trabalho, relacionando-os aos objetivos propostos e destacando os aprendizados obtidos durante o processo. Também são indicadas possíveis melhorias e caminhos para continuidade do projeto em versões futuras da aplicação

6.1. Conclusão

O trabalho teve como objetivo desenvolver e validar uma aplicação Web voltada ao gerenciamento de fichas e sistemas de RPG de mesa, oferecendo uma alternativa prática e flexível para jogadores e mestres. A proposta surgiu da necessidade de reunir, em um só lugar, as informações e ferramentas utilizadas nas mesas de jogo, facilitando a organização e reduzindo a dependência do uso de papel e caneta.

Durante o desenvolvimento, foram aplicadas boas práticas de programação e design de interface, buscando criar um ambiente intuitivo para o uso cotidiano da aplicação. A estrutura foi concebida de modo que jogadores e mestres possam utilizar fichas e campanhas sem a necessidade de conhecimentos técnicos. Contudo, a criação de novos sistemas exige a definição de *templates* estruturados em JSON, o que pressupõe familiaridade com hierarquia de dados e com campos específicos, como *expression*, *editable* e *sessionEditable*. Dessa forma, a ferramenta se mostra acessível para uso, mas apresenta maior complexidade quando considerada sua extensão e personalização.

A validação com jogadores de RPG mostrou resultados positivos. Os participantes avaliaram bem a usabilidade e o visual da aplicação, além de reconhecerem seu potencial como centralizador de informações de jogo. Também surgiram críticas construtivas, especialmente sobre a falta de instruções mais detalhadas e a necessidade de deixar mais claro que o sistema pode ser alimentado pelos próprios usuários. Esses pontos serão considerados em versões futuras do projeto.

Com base nos resultados obtidos, conclui-se que os objetivos foram atingidos no que se refere à viabilidade técnica, à utilidade da aplicação e à aceitação inicial por parte do público-alvo. A proposta de não exigir conhecimentos técnicos mostrou-se válida para o usuário final no uso da ferramenta, embora a criação e extensão de novos sistemas ainda

demandem domínio técnico específico. Assim, o trabalho atingiu seu propósito principal ao demonstrar a viabilidade de uma solução flexível e multi-sistema, ao mesmo tempo em que evidencia limitações que orientam aprimoramentos futuros.

6.2. Trabalhos Futuros

Durante o desenvolvimento e a etapa de validação, foram identificadas diversas melhorias e extensões que podem orientar trabalhos futuros e evoluções da aplicação. Entre as principais possibilidades, destacam-se:

1. Criação de um guia ou tutorial interativo dentro da aplicação, auxiliando o usuário a compreender suas principais funcionalidades e o processo de criação de sistemas e fichas.
2. Aprimoramento da comunicação visual e textual da proposta do projeto, deixando mais evidente que os próprios usuários podem criar e compartilhar seus sistemas de RPG.
3. Adição de sistemas pré-configurados adicionais, como *Ordem Paranormal* e *Call of Cthulhu*, servindo como modelos de referência para novos usuários.
4. Implementação de testes de desempenho e acessibilidade, com o objetivo de otimizar o carregamento das páginas e garantir o uso confortável em diferentes dispositivos e condições de conexão.
5. Aprimoramento do suporte a dispositivos móveis, com foco em responsividade e adaptação da interface para telas menores.
6. Desenvolvimento de um sistema de comunicação por texto integrado às campanhas, permitindo que os jogadores interajam dentro da própria plataforma.
7. Criação de um histórico de rolagens de dados vinculados a cada campanha, para consulta e registro das interações ocorridas durante as sessões de jogo.
8. Implementação de uma interface gráfica dedicada à criação de sistemas de RPG diretamente na aplicação, permitindo que o usuário defina atributos, regras, expressões e permissões por meio de formulários visuais, sem a necessidade de manipulação explícita de arquivos JSON. Essa funcionalidade visa reduzir a dependência de conhecimentos técnicos e ampliar o acesso à criação e extensão de sistemas personalizados.
9. Definição de uma identidade visual e desenvolvimento de um *branding* próprio, fortalecendo o reconhecimento da aplicação e alinhando sua estética à proposta do projeto.
10. Realização de uma nova rodada de validação com um público maior e mais diverso, visando obter métricas mais sólidas sobre desempenho, usabilidade e aceitação.

Essas melhorias visam tornar a aplicação mais completa, acessível e representativa da comunidade de RPG, além de preparar o projeto para futuras iterações e possíveis implementações em escala maior.

Por fim, este trabalho representa uma contribuição prática para a integração de tecnologia e narrativa nos jogos de RPG de mesa, servindo como base para desenvolvimentos futuros na área.

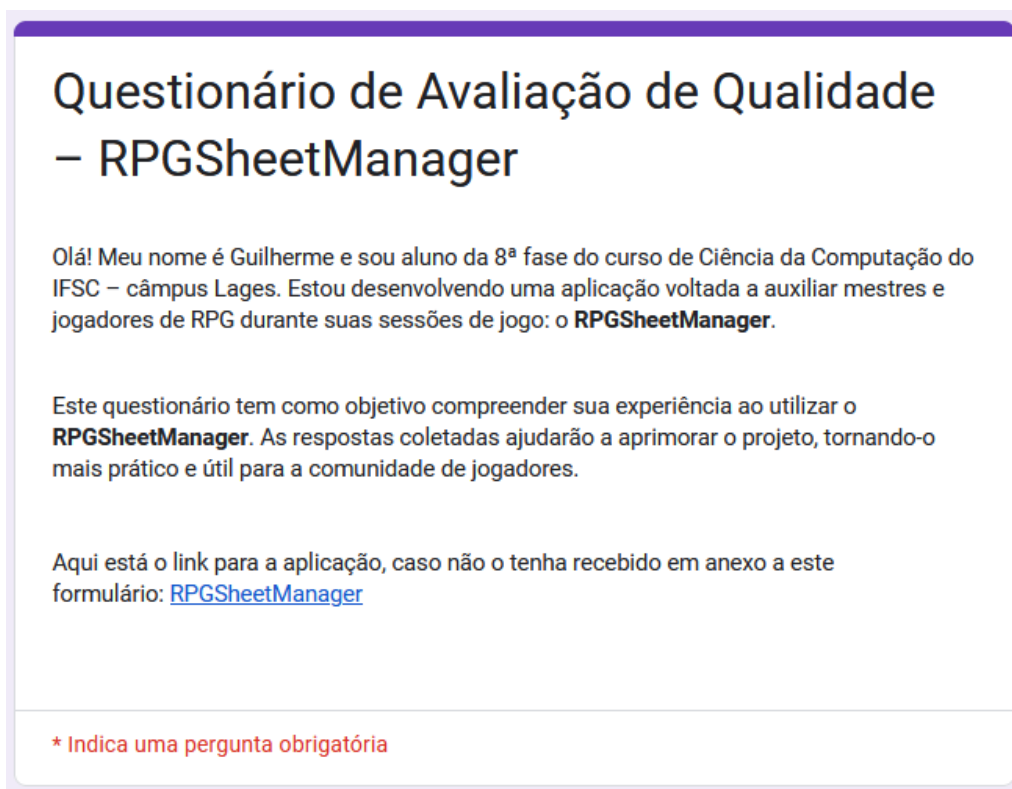
Referências

- Alves, B. M. (2022). Old quest. Publicado pela UICLAP.
- Foundry Gaming, LLC (2024a). About foundry virtual tabletop. <https://foundryvtt.com>. Acesso em: 10 abr. 2025.
- Foundry Gaming, LLC (2024b). Foundry virtual tabletop. <https://foundryvtt.com>. Acesso em: 10 abr. 2025.
- Henning, G. (2021). A terapia analítico comportamental das habilidades sociais em grupo para adolescentes com uso do rpg e da gamificação. Master's thesis, Instituto de Psicologia da Universidade de São Paulo.
- ISO/IEC (2025). Iso/iec 25010 — systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models. <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>. Acesso em: 09 out. 2025.
- Larian Studios (2025). Baldur's gate 3 – official website. <https://baldursgate3.game>. Acesso em: 15 abr. 2025.
- Reddit (2024). Alguém mais acha que o aplicativo móvel do d&d beyond é meio ruim? https://www.reddit.com/r/DnD/comments/1cb074p/anyone_else_think_the_dnd_beyond_mobile_app_kind/. Acesso em: 10 abr. 2025.
- Roll20 (2024a). Roll20. <https://roll20.net>. Acesso em: 10 abr. 2025.
- Roll20 (2024b). Roll20 crash course. https://wiki.roll20.net/Roll20_Crash_Course. Acesso em: 10 abr. 2025.
- RPGamer (2020). The digital future of tabletop rpgs. <https://a16z.com/the-digital-future-of-tabletop-games/>. Acesso em: 10 abr. 2025.
- Shibata, M. N. S., Oliveira, V. S. d., Borges, W. N., e Abe, Y. R. (2023). Lorde dos dados: Sistema para criação de fichas de personagens e gerenciamento de aventuras em jogos de rpg. Trabalho de Conclusão de Curso (Técnico em Informática) – Etec Prof. Armando José Farinazzo, Centro Paula Souza. Orientação: Josilene Franco Pacheco.
- Silva, R. (2014). D&d 40 anos: a origem não tão secreta de dungeons & dragons. <https://www.rederpg.com.br/2014/01/17/dd-40-anos-a-origem-nao-tao-secreta-de-dd/>. Acesso em: 22 maio 2025.
- University of Wisconsin-Parkside (2024). Gen con at uw-parkside. <https://libguides.uwp.edu/gencon>. Acesso em: 15 abr. 2025.
- Wizards of the Coast (2024). D&d beyond. <https://www.dndbeyond.com>. Acesso em: 10 abr. 2025.
- Wizards of the Coast (2025). Dungeons & dragons — official website. <https://dnd.wizards.com/pt-BR>. Acesso em: 24 maio 2025.

Anexos

A. Formulário de Validação

Este anexo contém o formulário utilizado para a coleta de dados durante a validação da aplicação, conforme apresentado na Seção 5. O questionário foi elaborado com base na norma ISO/IEC 25010 e aplicado por meio da plataforma *Google Forms*.



The image shows a screenshot of a Google Forms questionnaire titled "Questionário de Avaliação de Qualidade – RPGSheetManager". The text on the page is as follows:

**Questionário de Avaliação de Qualidade
– RPGSheetManager**

Olá! Meu nome é Guilherme e sou aluno da 8ª fase do curso de Ciência da Computação do IFSC – câmpus Lages. Estou desenvolvendo uma aplicação voltada a auxiliar mestres e jogadores de RPG durante suas sessões de jogo: o **RPGSheetManager**.

Este questionário tem como objetivo compreender sua experiência ao utilizar o **RPGSheetManager**. As respostas coletadas ajudarão a aprimorar o projeto, tornando-o mais prático e útil para a comunidade de jogadores.

Aqui está o link para a aplicação, caso não o tenha recebido em anexo a este formulário: [RPGSheetManager](#)

* Indica uma pergunta obrigatória

Figura 19. Página inicial do formulário de validação.

A seguir são apresentadas as perguntas que compõem o formulário de validação utilizado na pesquisa, conforme disponibilizado originalmente na plataforma *Google Forms*. O formulário completo pode ser acessado em: formulário de validação.

<p>Perfil do Participante</p> <p>Essas perguntas servem para entender um pouco sobre você e seu contato com o RPG.</p>	<p>Com que frequência joga? *</p> <p><input type="radio"/> Mais de uma vez na semana</p> <p><input type="radio"/> Semanalmente</p> <p><input type="radio"/> Mensalmente</p> <p><input type="radio"/> Raramente</p> <p><input type="radio"/> Não joga</p>
<p>Idade *</p> <p>Sua resposta _____</p>	<p>Quais sistemas costuma jogar? (ex: D&D 5e, Tormenta20, Call of Cthulhu, etc.)</p> <p>Sua resposta _____</p>
<p>Gênero *</p> <p><input type="radio"/> Masculino</p> <p><input type="radio"/> Feminino</p> <p><input type="radio"/> Outro</p> <p><input type="radio"/> Prefiro não responder</p>	<p>Qual tipo de jogador você se considera? *</p> <p><input type="radio"/> Mestre</p> <p><input type="radio"/> Jogador</p> <p><input type="radio"/> Ambos</p> <p><input type="radio"/> Nenhum</p>
<p>Nível de escolaridade *</p> <p><input type="radio"/> Ensino Fundamental</p> <p><input type="radio"/> Ensino Médio</p> <p><input type="radio"/> Superior Incompleto</p> <p><input type="radio"/> Superior Completo</p> <p><input type="radio"/> Pós-graduação</p>	<p>Com que frequência você usa computador ou celular para atividades como estudar, trabalhar ou jogar? *</p> <p><input type="radio"/> Nunca</p> <p><input type="radio"/> Raramente</p> <p><input type="radio"/> Às vezes</p> <p><input type="radio"/> Frequentemente</p> <p><input type="radio"/> Diariamente</p>
<p>Há quanto tempo joga RPG de mesa (em anos) ? *</p> <p>Sua resposta _____</p>	

Figura 20. Perfil do participante e caracterização do uso de RPG e tecnologia.

<p>Você costuma acessar sites ou aplicativos para jogos, estudos ou redes sociais? *</p> <p><input type="radio"/> Nunca</p> <p><input type="radio"/> Raramente</p> <p><input type="radio"/> Às vezes</p> <p><input type="radio"/> Frequentemente</p> <p><input type="radio"/> Diariamente</p>	<p>Adequação Funcional - O sistema produz resultados corretos? *</p> <p>1 2 3 4 5</p> <p>Discordo totalmente <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Concordo totalmente</p>
<p>Você já utilizou sites ou aplicativos para criar, armazenar ou jogar fichas de RPG? *</p> <p><input type="radio"/> Sim</p> <p><input type="radio"/> Não</p>	<p>Adequação Funcional - O sistema realmente ajuda a realizar as tarefas do jogo (ex: montar ficha, rolar dados, consultar informações de personagens) *</p> <p>1 2 3 4 5</p> <p>Discordo totalmente <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Concordo totalmente</p>
<p>Se sim, qual(is)?</p> <p>Sua resposta _____</p>	<p>Eficiência de Desempenho - O sistema responde de forma rápida às ações? *</p> <p>1 2 3 4 5</p> <p>Discordo totalmente <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Concordo totalmente</p>
<p>Avaliação da Ferramenta</p> <p>Aqui queremos saber o que você achou do Gerenciador de Fichas: se ele é fácil de usar, funciona bem e ajuda no jogo.</p>	<p>Eficiência de Desempenho - O desempenho geral (carregamento, processamento) é adequado ao que o sistema faz? *</p> <p>1 2 3 4 5</p> <p>Discordo totalmente <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Concordo totalmente</p>
<p>Adequação Funcional - O sistema oferece todas as funções que você espera para apoiar uma sessão de RPG?</p> <p>1 2 3 4 5</p> <p>Discordo totalmente <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Concordo totalmente</p>	<p>Usabilidade - É fácil aprender a usar o sistema sem precisar de ajuda? *</p> <p>1 2 3 4 5</p> <p>Discordo totalmente <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Concordo totalmente</p>

Figura 21. Avaliação funcional e de desempenho do sistema de gerenciamento de fichas.

<p>Usabilidade - As opções e menus são claros e intuitivos? *</p> <p>1 2 3 4 5</p> <p>Discordo totalmente <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Concordo totalmente</p>	<p>Você faria uso deste sistema? Justifique sua resposta *</p> <p>Sua resposta _____</p>
<p>Usabilidade - A aparência do sistema é agradável e coerente? *</p> <p>1 2 3 4 5</p> <p>Discordo totalmente <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Concordo totalmente</p>	<p>Caso tenha respondido "não" à pergunta anterior, o que seria necessário para o sistema se adaptar às suas necessidades?</p> <p>Sua resposta _____</p>
<p>Usabilidade - Pessoas com pouca experiência digital conseguiriam usar o sistema? *</p> <p>1 2 3 4 5</p> <p>Discordo totalmente <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Concordo totalmente</p>	<p>Quais melhorias você sugeriria? (funcionalidades, aparência, usabilidade, etc.)</p> <p>Sua resposta _____</p>
<p>Confiabilidade - O sistema funciona sem travar ou apresentar erros? *</p> <p>1 2 3 4 5</p> <p>Discordo totalmente <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Concordo totalmente</p>	<p>Há algum aspecto positivo que você destacaria no sistema?</p> <p>Sua resposta _____</p>
<p>Confiabilidade - Você encontrou alguma falha ou problema no uso do sistema? Se sim, relate a inconsistência.</p> <p>Sua resposta _____</p>	<p>Há algum aspecto negativo que você destacaria no sistema?</p> <p>Sua resposta _____</p>
<p>Confiabilidade - Você encontrou alguma falha ou problema no uso do sistema? Se sim, relate a inconsistência.</p> <p>Sua resposta _____</p>	<p>Existe algo que tenha dificultado o uso ou que você não tenha entendido sobre o sistema?</p> <p>Sua resposta _____</p>

Figura 22. Avaliação de usabilidade, confiabilidade e feedback geral dos usuários.

B. Resultados e Gráficos da Pesquisa

Este anexo apresenta recortes dos resultados coletados na etapa de validação, com os gráficos gerados pelo *Google Forms*. As respostas completas estão disponíveis na planilha pública: planilha pública de resultados.

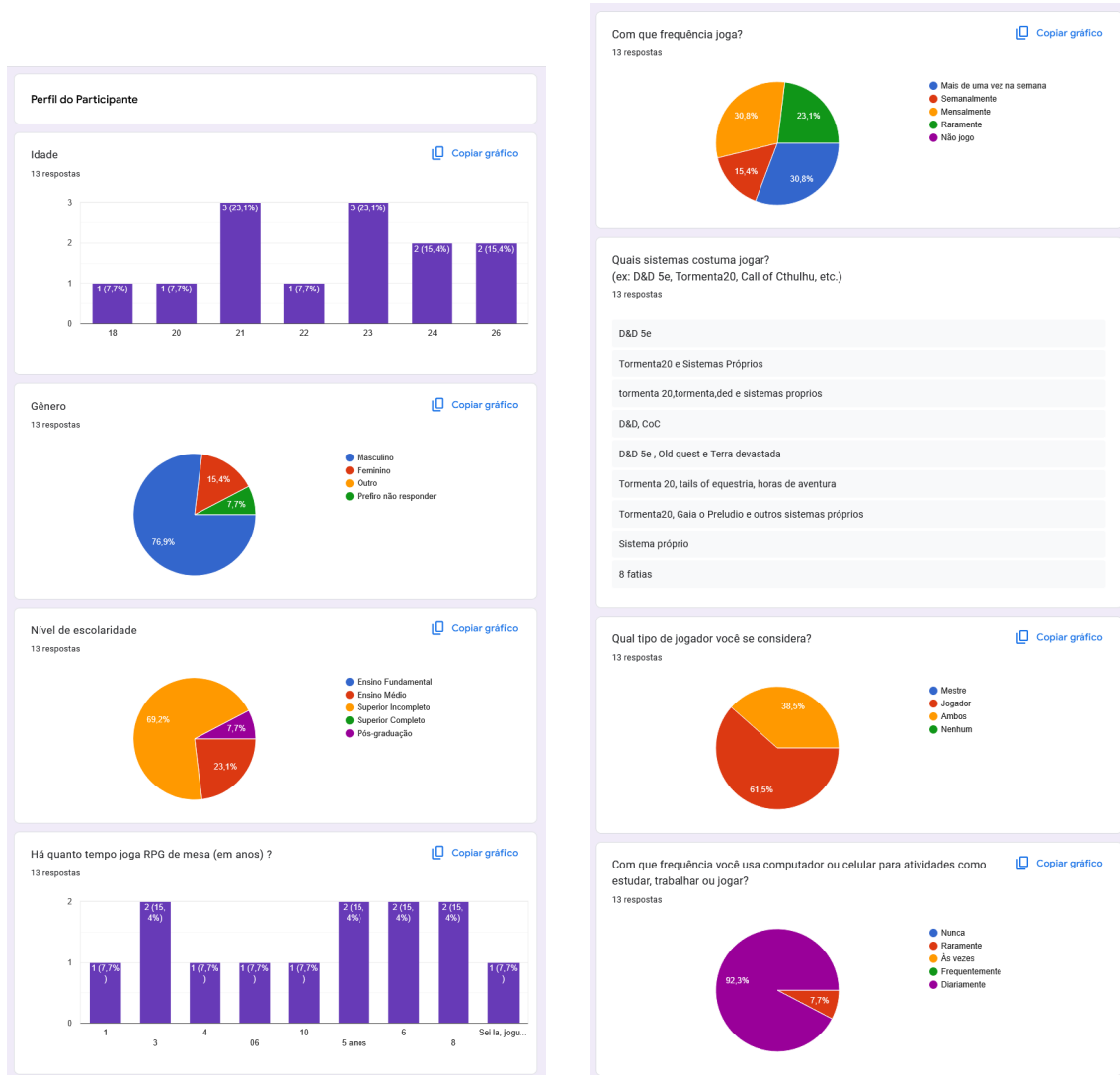


Figura 23. Perfil demográfico e experiência prévia dos participantes com RPG.

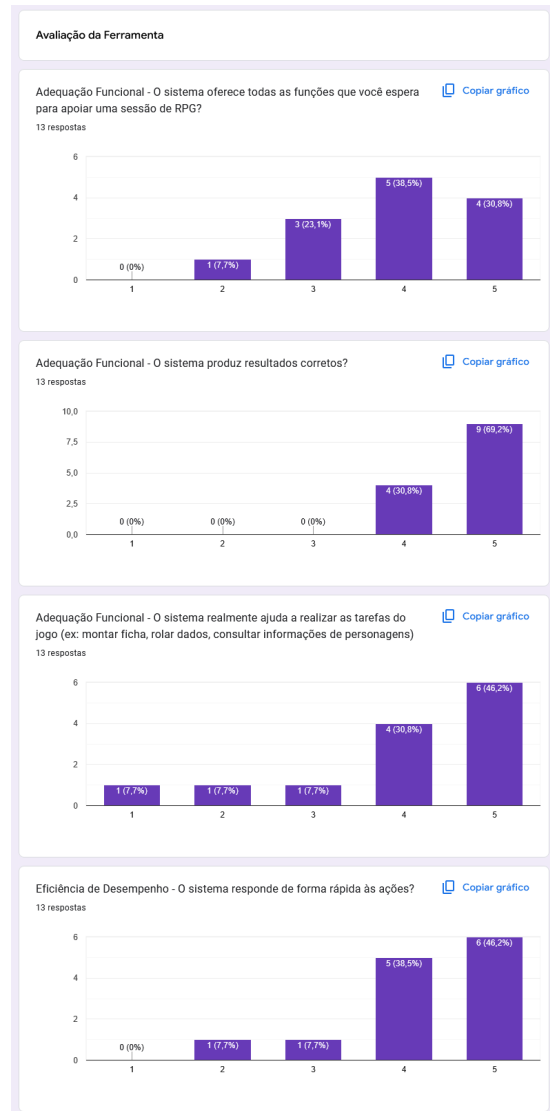
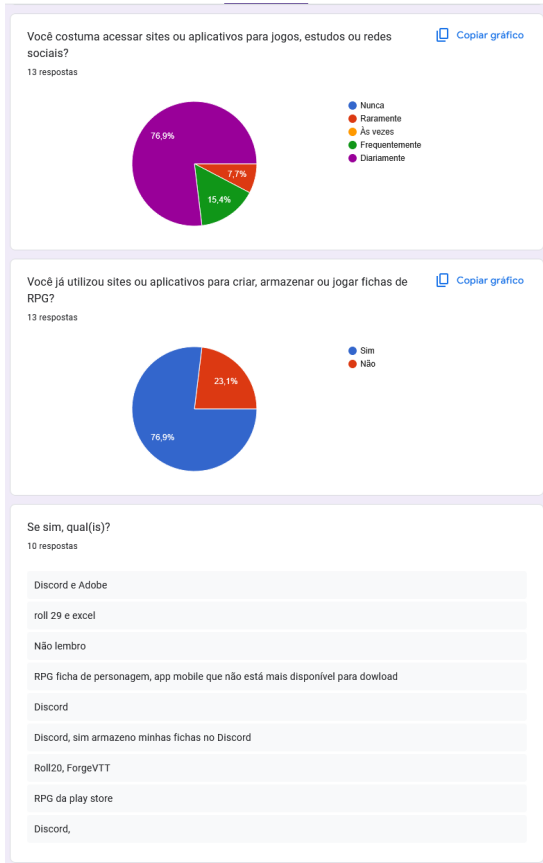


Figura 24. Hábitos de uso de tecnologia e utilização prévia de ferramentas digitais para RPG.

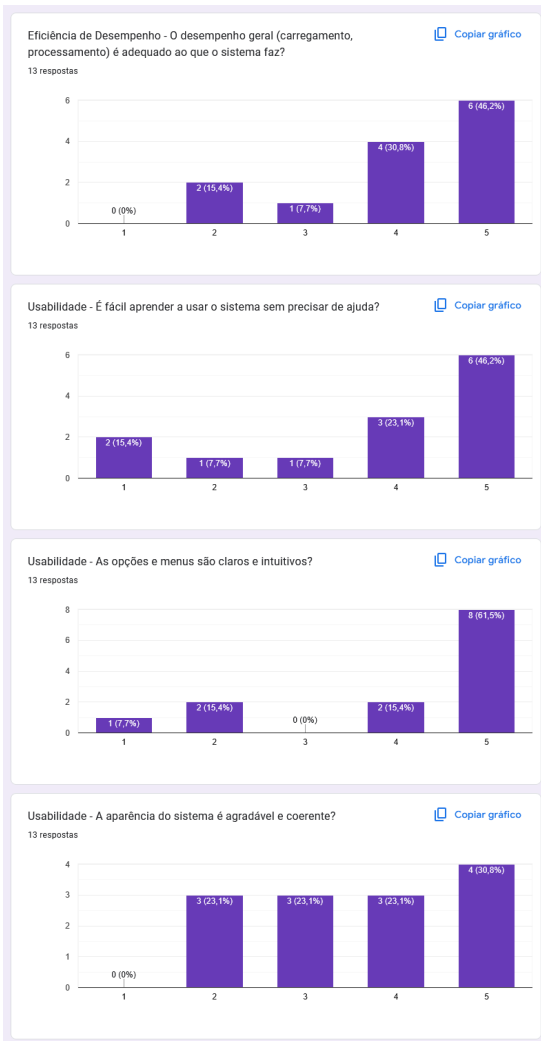


Figura 25. Resultados da avaliação de adequação funcional do sistema.

Você faria uso deste sistema? Justifique sua resposta

13 respostas

Provavelmente não, pois o site é incompleto em vários fatores, e não apresenta os sistemas que é necessário, por exemplo mesmo que já exista 3 sistemas no site e necessário salvar ele, caso não ele nem mesmo aparece

Seria bem util para guardar minhas fichas de personagens em um lugar mais organizado

nao, ainda falta alguns sistemas como por exemplo tormenta 20

Sim ele é bom

Sim, o app que usava não está mais no ar e esse atende tudo que preciso

Sim, antes de crashar rodava lisinho no celular

Se eu jogasse rpg sempre, sim eu usaria

Sim, a ideia inicial é interessante e com melhorias tem boas chances de ser sensacional

Caso tenha respondido "não" à pergunta anterior, o que seria necessário para o sistema se adaptar às suas necessidades?

2 respostas

Os Sistemas adicionados estarem mais completos, por exemplo o fato de ter o sistema de D&D 5e e vc ainda precisar escrever cada magia e adicionar o espaço pra ela manualmente faz vc não ter incentivo a usar o site

Adicionar mais informações

Quais melhorias você sugeriria? (funcionalidades, aparência, usabilidade, etc.)

11 respostas

Tornar a acessibilidade do site melhor, com destaques nas partes importantes, pois quando abri o site pensei que só havia um sistema e não tinha nada indicando que teria outros até eu acessar outra aba

Mais opções e visualização visual parecida com a do sistema de mesa

Trazer mais identidade para o site, criar uma identidade visual e branding

Manual de uso

Um modelador de personagens, para q vc possa salvar o seu personagem do jeito q vc curte

Portar outros sistemas já feitos, e explicar como funciona e se funciona a portabilidade de sistemas próprios, talvez criar uma maneira de portar magias direto para as fichas como o site de Grimório de T20.

Aumentar tooltips nos campos da ficha, para ter uma breve explicação do que impacta na ficha do jogador aquilo.

Trocar o redirecionamento do botão de notificação por um popup, o redirecionamento que ele faz quando clicado é bastante incomodo.

Há algum aspecto positivo que você destacaria no sistema?

11 respostas

A possibilidade de adicionar os próprios sistemas

Facilidade de cadastro

Intuívo e personalizável

A possibilidade de adição de novos sistemas

Ele é fácil de se usar

Facilidade de uso

O layout se mostrou bastante convidativo e amigável.

facilidade de acesso ao personagem

Ele responde rápido sem travamentos

Há algum aspecto negativo que você destacaria no sistema?

9 respostas

a falta de informação

Travou meu Chrome e no celular os textos sumiam

Não sei

Inicial demais para pontuar, posso dizer que faltam sistemas para testes

A falta de explicações ou um "guia" para utilizar recursos, como a criação de um sistema.

aparência meio crua

não

Falta de informações

Lento

Existe algo que tenha dificultado o uso ou que você não tenha entendido sobre o sistema?

8 respostas

a falta de informação

Não

A princípio nada

Não

Acho que o único ponto que se mostrou uma dificuldade é o fato de não ter responsividade quando utilizado em mobile.

não

Falta de informações

Muitas opções desativadas, sem nenhum feedback do motivo

Figura 26. Resultados da avaliação de desempenho, usabilidade e confiabilidade do sistema.