



DENNYS KAYKHE JUSTINO DE ASSIS

JULIO CEZAR ARAUJO GOMES

LUCAS GABRIEL DE SOUSA

VINÍCIUS FERREIRA DE AQUINO

RPG Ruby

Site de RPG de mesa

Equipe Panteon Table

Orientadores Profa. Graciete Henriques dos Santos

Prof. Israel Nuncio Dias Lucania e

Co-orientador Prof. Kleyton Sartori Leite

Mongaguá

07/2026

DENNYS KAYKHE JUSTINO DE ASSIS

JULIO CEZAR ARAUJO GOMES

LUCAS GABRIEL DE SOUSA

VINÍCIUS FERREIRA DE AQUINO

RPG Ruby

Site de RPG de mesa

Equipe Panteon Table

Trabalho de Conclusão de Curso
apresentado à Escola Técnica Adolpho
Berezin, como parte dos requisitos para a
obtenção do título de Técnico em Informática.
Orientadores Profa. Graciete Henriques dos
Santos e Prof. Israel Nuncio Dias Lucania

Mongaguá

07/2026

Dedicamos

Dedicamos este trabalho a todos aqueles que acreditam no poder do conhecimento, da criatividade e da perseverança. Aos familiares, amigos e professores que, direta ou indiretamente, contribuíram para nossa formação pessoal e acadêmica, oferecendo apoio, incentivo e inspiração ao longo desta jornada.

Também dedicamos este projeto a todos os apaixonados pelo universo do RPG de mesa, cuja imaginação, colaboração e espírito criativo motivaram a concepção e o desenvolvimento desta proposta.

Por fim, dedicamos este trabalho a todos aqueles que acreditam que a tecnologia pode aproximar pessoas, fortalecer comunidades e transformar ideias em realidade.

AGRADECIMENTOS

Agradecemos, primeiramente, aos nossos orientadores, Prof.^a Graciete Henriques dos Santos e Prof. Kleyton Sartori Leite, pela dedicação, paciência e apoio ao longo do desenvolvimento deste trabalho. Seus conhecimentos, orientações e contribuições foram fundamentais para a construção deste projeto, auxiliando-nos na superação dos desafios encontrados durante sua elaboração.

Expressamos nossa gratidão pela confiança depositada em nossa equipe, pelas sugestões fornecidas em cada etapa do processo e pelo comprometimento em compartilhar experiências e conhecimentos que contribuíram significativamente para nossa formação acadêmica e profissional.

Por fim, registramos nosso sincero reconhecimento pelo incentivo e pela disponibilidade demonstrados durante todo o período de desenvolvimento deste Trabalho de Conclusão de Curso, tornando possível a concretização desta pesquisa e da proposta apresentada.

Muito obrigado.

"Não ousamos muitas coisas porque são difíceis, mas são difíceis porque não usamos." - Sêneca

RESUMO

O presente trabalho tem como objetivo analisar a evolução do RPG de mesa no contexto digital e desenvolver uma proposta de solução tecnológica voltada às necessidades da comunidade de jogadores e mestres. Inicialmente, foi realizada uma pesquisa bibliográfica de caráter qualitativo sobre a origem do RPG de mesa, seu desenvolvimento histórico, impacto social e cultural, bem como sua adaptação às plataformas digitais e às redes sociais de nicho. A análise permitiu identificar que, apesar da existência de diversas ferramentas voltadas ao RPG online, as funcionalidades necessárias para a organização e realização de campanhas encontram-se distribuídas entre múltiplas plataformas, exigindo que os usuários utilizem diferentes sistemas simultaneamente. Diante dessa problemática, foi proposta a plataforma Ruby RPG | Panteon Table, uma rede social especializada destinada à comunidade de RPG de mesa, integrando recursos de interação social, gerenciamento de campanhas e ferramentas de apoio à realização de sessões virtuais. Para o desenvolvimento do projeto, foram utilizadas tecnologias modernas de desenvolvimento web, incluindo TypeScript, React, Vite, Node.js, Express e MySQL, permitindo a construção de uma aplicação escalável e alinhada às necessidades do público-alvo. Em razão das limitações de tempo e recursos disponíveis para a execução do Trabalho de Conclusão de Curso, o escopo inicial foi reduzido, priorizando a implementação das funcionalidades essenciais para validação da proposta. Os resultados obtidos demonstram a viabilidade técnica da solução e evidenciam o potencial da plataforma para centralizar recursos atualmente dispersos em diferentes serviços, contribuindo para a organização, acessibilidade e fortalecimento da comunidade de RPG de mesa em ambiente digital.

PALAVRAS-CHAVES: RPG de mesa. Rede social de nicho. Virtual Tabletop. Desenvolvimento web. Comunidades digitais. TypeScript.

ABSTRACT

This study aims to analyze the evolution of tabletop role-playing games (RPGs) in the digital environment and to develop a technological solution focused on the needs of players and game masters. Initially, a qualitative bibliographic research was conducted regarding the origins of tabletop RPGs, their historical development, social and cultural impacts, as well as their adaptation to digital platforms and niche social networks. The analysis revealed that, despite the existence of several tools designed for online RPG activities, the functionalities required to organize and manage campaigns are currently distributed across multiple platforms, forcing users to rely on different systems simultaneously. To address this issue, the Ruby RPG | Panteon Table platform was proposed as a specialized social network for the tabletop RPG community, integrating social interaction features, campaign management tools, and resources to support virtual gaming sessions. The project was developed using modern web technologies, including TypeScript, React, Vite, Node.js, Express, and MySQL, enabling the creation of a scalable application aligned with the needs of its target audience. Due to time and resource limitations associated with the completion of the Course Conclusion Project, the original scope was reduced, prioritizing the implementation of essential features required to validate the proposal. The results demonstrate the technical feasibility of the solution and highlight its potential to centralize resources that are currently fragmented across different services, contributing to the organization, accessibility, and strengthening of the tabletop RPG community in the digital environment.

Key Words: Tabletop RPG. Niche social network. Virtual Tabletop. Web development. Digital communities. TypeScript.

LISTA DE FIGURAS

Figura 1 - Logo da Equipe	33
Figura 2 - Logo do Projeto	33
Figura 3 - MER	38
Figura 4 - Tela inicial (modo escuro) Fonte: Wireframe das Telas.....	49
Figura 5 - Tela inicial (modo claro)	49
Figura 6 - Tela inicial (status de perfil)	50
Figura 7 - Perfil do usuário.....	50
Figura 8 - Tela de configuração de perfil	51
Figura 9 - Amigos (meus amigos)	51
Figura 10 - Tela de amigos (Explorar afinidade)	52
Figura 11 - Notificações	52
Figura 12 - Tela de comunidades	53
Figura 13 - Tela de uma comunidade (Salão de mesas)	53
Figura 14 - Tela de crônicas (feed da comunidade)	54
Figura 15 - Tela da Wiki da comunidade (fórum)	54
Figura 16 - Tela do calendário da comunidade.....	55
Figura 17 - Tela dos canais de texto da comunidade	55
Figura 18 - Tela de criador de fichas para sistemas	56
Figura 19 - Tela de editor de ficha para sistemas	56
Figura 20 - Tela de criação de fichas para sistemas	57
Figura 21 - Tela de sistemas de RPG já existentes.....	57
Figura 22 - Tela de criação de sistemas de RPG's.....	58
Figura 23 - Tela de mensagens	58
Figura 24 - Tela de calendário pessoal.....	59
Figura 25 - Tela de storys de perfil	59
Figura 26 - Tela de mesas	60
Figura 27 - Tela inicial da mesa (com o chat da mesa aberto)	60
Figura 28 - Tela da mesa (fichas da mesa)	61
Figura 29 - Tela de mesa (criação de fichas/status)	61
Figura 30 - Tela de mesa (criação de fichas/biografia)	62
Figura 31 - Tela de mesa (criação de fichas/skills)	62
Figura 32 - Tela de mesa (criação de fichas/skills/criação)	63

Figura 33 - Tela de mesa (criação de fichas/anotações)	63
Figura 34 - Tela de mesa (mapas).....	64
Figura 35 - Tela de mesa (mapas/visualização de mapas).....	64
Figura 36 - Tela de mesa (mapas/visualização de mapas/privacidade)	65
Figura 37 - Tela de mesa (anotações pessoais).....	65
Figura 38 - Tela de mesa (tria sonora)	66
Figura 39 - Tela de mesa (tria sonora/criar faixa)	66
Figura 40 - Tela de mesa (Configuração da mesa)	67
Figura 41 - Tela de configuração de conta	67
Figura 42 - Login.....	68
Figura 43 - Cadastro	68
Figura 44 – Recuperação de senha.....	69
Figura 45 – Recuperação de conta (redefinição de senha)	69
Figura 46 – Tela inicial.....	70
Figura 47 – Tela inicial (criação de post/upload)	70
Figura 48 – Tela inicial (criação de post/link).....	71
Figura 49 – Post e comentários	71
Figura 50 – Perfil de usuário.....	72
Figura 51 – Status de perfil.....	72
Figura 52 – Edição de perfil	73
Figura 53 – Amigos do usuário	73
Figura 54 – Solicitações de amizade	74
Figura 55 – Usuários.....	74
Figura 56 – Pesquisa de usuários	75
Figura 57 – Chat de conversa.....	75
Figura 58 - Notificações	76
Figura 59 – Ficha de personagem	76
Figura 60 – Modelo de ficha	77
Figura 61 – Criação de ficha.....	77
Figura 62 – Criação de ficha.....	78
Figura 63 – Criação de ficha (seleção de sistema)	78
Figura 64 – Ficha de personagem	79
Figura 65 – Ficha de personagem	79
Figura 66 – Edição de ficha de personagem	80

Figura 67 – Edição de ficha de personagem (adição de funcionalidades).....	80
Figura 68 - Sistemas.....	81
Figura 69 – Perfil de sistema	81
Figura 70 – Modelos de fichas de sistemas.....	82
Figura 71 – Criação de sistema	82
Figura 72 – Manual de usuário	83
Figura 73 – Configuração de conta geral.....	83
Figura 74 – Configuração de conta (redes sociais)	84
Figura 75 – Configuração de conta (visibilidade/privacidade).....	84
Figura 76 – Configuração de conta (notificações)	85

SUMÁRIO

LISTA DE FIGURAS	7
INTRODUÇÃO	14
Fundamentação Teórica	15
O que é RPG de mesa	15
Origem do RPG	15
Evolução do RPG para o ambiente digital	16
Impacto social e cultural do RPG	17
Virtual Tabletop (VTT) como conceito	17
Referencial teórico	18
História do RPG de mesa	18
RPG e desenvolvimento social	18
RPG no ambiente digital	19
Redes sociais de nicho	19
Plataformas VTT no cenário atual	21
Justificativa	21
Metodologia de Desenvolvimento	22
1. NICHOS DE MERCADO DO PROJETO	24
1.1 Público-alvo	24
1.2 Problema	25
1.3 Solução Proposta	25
2. TECNOLOGIAS UTILIZADAS	27
2.1 Linguagem de Programação	27
TypeScript	27
2.2 Tecnologias e Frameworks Front-end	27
React	27
Vite	28

Tailwind CSS	28
2.3 Tecnologias e Arquitetura Back-end	28
Node.js	28
Express.....	29
TSX.....	29
2.4 Banco de Dados e Gerenciamento de Arquivos.....	29
MySQL.....	29
MySQL2 Connector	29
Multer.....	30
2.5 Bibliotecas de Integração, Segurança e Interface	30
Google Generative AI SDK.....	30
Framer Motion	30
Nodemailer	30
Bcrypt.js.....	30
Lucide React.....	31
React Hot Toast.....	31
3. EMPRESA.....	32
3.1 Missão	32
3.2 Visão	32
3.3 Valores	32
3.4 Logos	33
3.4.1 Logo da Equipe.....	33
3.4.2 Logo do Projeto	33
3.5 Slogan da empresa	34
3.6 Slogan do projeto	34
4. ANÁLISE.....	35
4.1. Descrição das funcionalidades.....	35

4.1.1 Módulo de Rede Social e Comunidade	35
4.1.2 Módulo de Gerenciamento e Preparação (O "Hub")	35
4.1.3 Módulo de Mesa Virtual (VTT - Virtual Tabletop)	36
4.1.4 Funcionalidades de Sistema e Customização	36
4.2. MER – Modelo do Banco de Dados	38
4.3. Create do Banco de Dados	39
4.4 Principais Selects do Banco de Dados	45
Select 1 – Consulta de Perfil de Usuário	45
Select 2 – Feed de Publicações	46
Select 3 – Fichas de Personagens	46
Select 4 – Sistemas Favoritados pelos Usuários	46
Select 5 – Conversas e Participantes	47
Select 6 – Mensagens Enviadas	47
Select 7 – Notificações dos Usuários	47
Select 8 – Ranking de Sistemas Mais Utilizados	48
4.5. Wireframe das Telas	49
4.6 Prints das Telas	68
4.7 Trecho do Código Fonte	85
1. MANUAL DO USUÁRIO	96
1. INTRODUÇÃO E ESCRIVENTE DE AVENTURAS	96
1.1 Propósito do Sistema Ruby RPG	96
1.2 Visão Geral da Plataforma	96
2. ARQUITETURA DO PORTAL, CADASTRO E CONTROLE DE ACESSO	96
2.1 Processo de Registro e Autenticação Criptográfica	96
2.2 Recuperação de Acesso	96
3. A REDE SOCIAL E FEED OPERACIONAL (PÁGINA INICIAL)	96

3.1 Criação de Publicações Multi-mídia.....	97
3.2 Níveis de Privacidade de Postagens	97
4. DIRETÓRIO GLOBAL DE REINOS (SISTEMAS)	97
4.1 Navegação de Sistemas e Classificações	97
4.2 Cadastro de Novos Reinos e Ajuste de Imagem sem Tarja Preta.....	97
5. GERENCIADOR DINÂMICO DE FICHAS DE PERSONAGENS	97
5.1 Criação Baseada em Template ou Criação Livre	97
5.2 Estrutura Modular: Seções, Linhas e Colunas.....	98
5.3 Atributos, Matemática Relativa (+5, -3) e Fórmulas.....	98
5.4 Rolador Integrado de Dados.....	98
5.5 Logs Históricos de Desfazer/Refazer (Undo/Redo) e Auto-Save	98
4. CONCLUSÃO.....	99
5. REFERÊNCIA BIBLIOGRÁFICA.....	101
6. ANEXOS	Erro! Indicador não definido.

INTRODUÇÃO

O Role Playing Game (RPG), ou jogo de interpretação de papéis, constitui uma modalidade de jogo baseada na construção coletiva de narrativas, na interpretação de personagens fictícios e na interação entre os participantes por meio de regras previamente estabelecidas. Diferentemente de jogos competitivos tradicionais, o RPG caracteriza-se pela cooperação entre os jogadores e pelo desenvolvimento compartilhado de histórias, permitindo a criação de experiências narrativas complexas e imersivas (WITWER, 2016).

Desde sua criação com o lançamento de Dungeons & Dragons em 1974, o RPG passou por um processo contínuo de expansão e adaptação, consolidando-se como um importante fenômeno cultural e social. Ao longo das décadas, novos sistemas, cenários e comunidades surgiram, ampliando significativamente o alcance da prática e atraindo públicos de diferentes faixas etárias e contextos socioculturais (WITWER, 2016).

O avanço das tecnologias digitais promoveu profundas transformações na forma como os jogadores interagem e organizam suas atividades relacionadas ao RPG. Ferramentas de comunicação online, plataformas especializadas e sistemas de Virtual Tabletop (VTT) passaram a desempenhar papel fundamental na realização de campanhas remotas, permitindo que participantes de diferentes localidades compartilhem experiências de jogo em tempo real (CHIAPPA; RIBEIRO, 2023).

Apesar dos avanços tecnológicos, observa-se que a experiência digital do RPG ainda se encontra fragmentada entre diversas plataformas e serviços distintos. Frequentemente, os usuários precisam utilizar simultaneamente redes sociais, aplicativos de comunicação por voz, sistemas de compartilhamento de arquivos e plataformas VTT para organizar e executar suas campanhas. Essa dispersão de ferramentas pode gerar dificuldades de gerenciamento, reduzir a acessibilidade para novos jogadores e comprometer a experiência de utilização dos recursos disponíveis.

Nesse contexto, surge a proposta da plataforma Ruby RPG, desenvolvida pela equipe Panteon Table, com o objetivo de integrar recursos sociais e ferramentas especializadas para RPG em um único ambiente digital. A proposta busca oferecer funcionalidades de interação social, gerenciamento de campanhas, compartilhamento de conteúdos e execução de sessões virtuais, promovendo maior praticidade, acessibilidade e integração para a comunidade de jogadores.

Dessa forma, o presente trabalho tem como objetivo analisar o cenário atual do RPG de mesa no ambiente digital, compreender as necessidades existentes dentro desse nicho e apresentar o desenvolvimento de uma plataforma voltada à centralização das principais ferramentas utilizadas pelos praticantes de RPG online. Além disso, busca-se demonstrar como a integração entre recursos de rede social e funcionalidades de Virtual Tabletop pode contribuir para a ampliação e fortalecimento das comunidades especializadas nesse segmento.

Fundamentação Teórica

A fundamentação teórica deste trabalho tem como objetivo apresentar os conceitos centrais relacionados ao RPG de mesa, sua origem, seu desenvolvimento ao longo do tempo e seu impacto social e cultural, com base na literatura acadêmica e social pesquisada.

O que é RPG de mesa

O RPG de mesa, sigla para Role Playing Game, consiste em um jogo de interpretação no qual os participantes assumem papéis de personagens fictícios inseridos em um universo narrativo compartilhado. Durante as sessões, os jogadores tomam decisões e realizam ações por meio de seus personagens, enquanto um participante, geralmente denominado Mestre ou Narrador, é responsável por conduzir a história, apresentar desafios e interpretar o ambiente de jogo (WITWER, 2016).

A dinâmica do RPG baseia-se na cooperação entre os participantes e na construção coletiva de narrativas. As ações realizadas pelos personagens são mediadas por regras específicas definidas pelo sistema utilizado, frequentemente associadas à utilização de dados para determinar o sucesso ou fracasso das ações executadas. Essa combinação entre narrativa, estratégia e interpretação torna o RPG uma experiência singular dentro do universo dos jogos (CHIAPPA; RIBEIRO, 2023).

Além de seu caráter recreativo, o RPG também estimula habilidades relacionadas à criatividade, comunicação, resolução de problemas e trabalho em equipe, fatores que contribuem para sua relevância social e cultural ao longo das últimas décadas.

Origem do RPG

As origens do RPG de mesa estão diretamente relacionadas aos jogos de guerra estratégicos, conhecidos como *wargames*, que simulavam conflitos militares por meio de miniaturas e conjuntos de regras específicas. A partir dessas simulações

surgiu a ideia de controlar personagens individuais inseridos em narrativas mais elaboradas, estabelecendo as bases do que viria a se tornar o RPG moderno.

O principal marco histórico do gênero ocorreu em 1974 com a publicação de *Dungeons & Dragons*, criado por Gary Gygax e Dave Arneson. O sistema inovou ao combinar elementos narrativos, interpretação de personagens e mecânicas estruturadas de jogo, influenciando praticamente todos os sistemas de RPG desenvolvidos posteriormente (WITWER, 2016).

Nas décadas seguintes, o RPG expandiu-se internacionalmente, originando novos sistemas, cenários e comunidades de jogadores. Com o crescimento da internet e o desenvolvimento das tecnologias digitais, a prática também passou a incorporar ferramentas online, possibilitando a realização de campanhas remotas e a formação de comunidades globais dedicadas ao hobby (CHIAPPA; RIBEIRO, 2023).

Evolução do RPG para o ambiente digital

O desenvolvimento das tecnologias da informação transformou significativamente a forma como diversas atividades sociais são realizadas, incluindo os jogos de interpretação de papéis. Tradicionalmente praticado em ambientes presenciais, o RPG passou a incorporar recursos digitais que permitem a interação remota entre jogadores, ampliando seu alcance e possibilitando a formação de grupos compostos por participantes de diferentes regiões geográficas.

Segundo Chiappa e Ribeiro (2023), a utilização de plataformas digitais permitiu que o RPG mantivesse suas características fundamentais mesmo quando praticado a distância. Os autores destacam que sistemas modernos possibilitam a integração entre comunicação em tempo real, armazenamento de informações e gerenciamento de recursos utilizados durante as campanhas.

A evolução tecnológica observada no RPG acompanha um fenômeno mais amplo relacionado ao crescimento dos sistemas computacionais. Conforme afirma Sommerville (2011), o software tornou-se elemento essencial da sociedade contemporânea, estando presente em praticamente todos os serviços e processos digitais. Essa realidade também influenciou diretamente a forma como comunidades de RPG passaram a se organizar e interagir.

A adaptação do RPG ao ambiente digital intensificou-se principalmente durante o período de isolamento social decorrente da pandemia de COVID-19, quando plataformas virtuais passaram a desempenhar papel fundamental na continuidade das campanhas. Como resultado, houve crescimento significativo da utilização de

ferramentas online especializadas, consolidando um novo modelo de interação para os praticantes do hobby (CHIAPPA; RIBEIRO, 2023).

Impacto social e cultural do RPG

O RPG de mesa caracteriza-se por ser uma atividade baseada na cooperação e na construção coletiva de narrativas. Diferentemente de jogos que possuem foco exclusivo na competição, o RPG incentiva a comunicação constante entre os participantes, promovendo o desenvolvimento de habilidades sociais, criativas e estratégicas.

Conforme descrevem Chiappa e Ribeiro (2023, p. 2), o RPG de mesa é um “jogo de interpretação de personagens que possuem emoções, diálogos e habilidades escolhidos pelo jogador”, sendo conduzido por uma narrativa construída coletivamente. Essa característica torna o RPG uma atividade que ultrapassa o entretenimento, contribuindo para o desenvolvimento da criatividade e da interação social.

Além disso, a necessidade de assumir diferentes papéis durante as campanhas favorece o desenvolvimento da empatia e da capacidade de compreender perspectivas distintas. Os participantes são constantemente estimulados a resolver problemas, trabalhar em equipe e tomar decisões em conjunto, competências que podem ser aplicadas em diversos contextos sociais e profissionais (WITWER, 2016).

Ao longo das últimas décadas, o RPG consolidou-se como importante manifestação cultural, influenciando produções literárias, cinematográficas, jogos digitais e comunidades especializadas. Seu crescimento demonstra a capacidade da prática de se adaptar às transformações sociais e tecnológicas sem perder sua essência colaborativa (WITWER, 2016).

Virtual Tabletop (VTT) como conceito

O Virtual Tabletop (VTT), ou mesa virtual, consiste em plataformas digitais desenvolvidas para reproduzir a experiência do RPG de mesa em ambiente online. Essas ferramentas disponibilizam recursos específicos para a realização das sessões, incluindo mapas interativos, fichas digitais, sistemas de rolagem automática de dados e mecanismos de comunicação em tempo real.

O surgimento dos sistemas VTT está diretamente relacionado à necessidade de adaptar as atividades tradicionalmente presenciais para ambientes virtuais. Segundo Santos, Sant'Anna e Paim (2023), essas plataformas ampliam a

acessibilidade do RPG ao permitir que participantes localizados em diferentes regiões possam compartilhar a mesma campanha de forma simultânea.

Além disso, os sistemas VTT preservam aspectos fundamentais da experiência tradicional do RPG, como a interpretação de personagens, a construção coletiva da narrativa e a aplicação das regras do sistema utilizado. Dessa forma, representam uma importante convergência entre tecnologia e entretenimento colaborativo, consolidando-se como ferramentas amplamente utilizadas pela comunidade de RPG contemporânea.

Referencial teórico

História do RPG de mesa

O RPG de mesa tem suas origens nos jogos de guerra estratégicos (*wargames*), que simulavam batalhas militares por meio de regras e miniaturas. A partir dessas simulações, desenvolveu-se a proposta de controlar personagens individuais inseridos em narrativas ficcionais mais complexas.

O marco histórico do RPG moderno foi a criação de *Dungeons & Dragons*, publicado em 1974 por Gary Gygax e Dave Arneson. Conforme descrito em *Dungeons & Dragons: O Império da Imaginação*, o sistema introduziu a combinação entre interpretação de personagens e regras estruturadas, estabelecendo um novo modelo de experiência lúdica.

Segundo Witwer (2016), o sucesso inicial do jogo impulsionou o crescimento do mercado editorial especializado, promovendo a expansão do RPG para diferentes países e consolidando-o como fenômeno cultural. No Brasil, o RPG ganhou maior visibilidade a partir da década de 1990, com a tradução de sistemas estrangeiros e o surgimento de produções nacionais, fortalecendo a comunidade de jogadores.

Dessa forma, o RPG evoluiu de uma prática derivada de simulações militares para um modelo narrativo colaborativo, centrado na criatividade e na interação social.

RPG e desenvolvimento social

O RPG de mesa possui relevante dimensão social, uma vez que sua dinâmica exige comunicação constante, cooperação e construção coletiva de narrativas. Diferentemente de jogos exclusivamente competitivos, o RPG depende da colaboração entre os participantes para o desenvolvimento da história.

Conforme afirmam Chiappa e Ribeiro (2023, p. 2):

“o RPG de mesa [...] é um jogo de interpretação de personagens [...] determinados por uma história criada por um dos jogadores, denominado ‘Mestre’”.

Essa estrutura evidencia a necessidade de interação entre os jogadores, promovendo habilidades como argumentação, criatividade e resolução de problemas.

Além disso, a prática do RPG contribui para o desenvolvimento da empatia, pois os participantes assumem papéis distintos e exploram diferentes perspectivas dentro da narrativa. Witwer (2016) destaca que o RPG se consolidou como uma forma de expressão cultural e social, ultrapassando a condição de simples entretenimento.

Assim, o RPG pode ser compreendido como ferramenta que estimula competências sociais e favorece a construção de vínculos interpessoais.

RPG no ambiente digital

Com o avanço das tecnologias da informação, o RPG de mesa passou a incorporar recursos digitais que possibilitam sua realização de forma remota. Esse processo intensificou-se durante o período de isolamento social causado pela pandemia de COVID-19.

De acordo com Chiappa e Ribeiro (2023), a adaptação do RPG ao ambiente online permitiu que jogadores mantivessem a prática do jogo por meio de plataformas digitais, chats e sistemas automatizados de rolagem de dados. Os autores destacam que ferramentas web foram desenvolvidas com o objetivo de replicar funcionalidades da mesa presencial.

Nesse contexto, observa-se que:

A tecnologia utilizada para o front-end foi ReactJs e, para o back-end, foi o Firebase. O Firebase permite a armazenagem e a sincronização de dados em tempo real, enquanto o ReactJs possibilita a atualização dinâmica da interface (CHIAPPA; RIBEIRO, 2023, p. 3).

A integração entre essas tecnologias demonstra que o RPG pode ser adaptado às transformações tecnológicas contemporâneas, preservando seus elementos essenciais enquanto amplia sua acessibilidade.

Redes sociais de nicho

As redes sociais de nicho são plataformas digitais desenvolvidas para atender grupos específicos de usuários que compartilham interesses, atividades ou objetivos em comum. Diferentemente das redes sociais generalistas, como Facebook e Instagram, essas plataformas concentram-se em comunidades especializadas,

oferecendo funcionalidades direcionadas às necessidades de determinado público e promovendo interações mais relevantes entre seus membros.

Segundo Recuero (2009), as redes sociais na internet constituem estruturas formadas por atores e conexões sociais, possibilitando a formação de comunidades baseadas em interesses compartilhados. Nesse contexto, as redes de nicho surgem como ambientes capazes de fortalecer vínculos entre indivíduos que possuem afinidades específicas, favorecendo a troca de conhecimentos e experiências relacionadas ao tema central da comunidade.

No universo do RPG de mesa, as redes sociais de nicho desempenham papel importante na divulgação de campanhas, recrutamento de jogadores, compartilhamento de materiais e desenvolvimento de discussões sobre sistemas, cenários e mecânicas de jogo. Essas plataformas permitem que praticantes de diferentes regiões geográficas mantenham contato constante e encontrem grupos compatíveis com seus interesses e disponibilidade de horários (SANTOS; SANT'ANNA; PAIM, 2023).

O crescimento das comunidades digitais especializadas está diretamente relacionado à expansão da internet e ao aumento da conectividade entre usuários. Conforme observam Silva e Souza (2020), as ferramentas online possibilitam a manutenção das interações sociais e das atividades colaborativas mesmo quando os participantes não compartilham o mesmo espaço físico. No caso do RPG, essa característica foi fundamental para a expansão da prática e para o fortalecimento de comunidades virtuais dedicadas ao hobby.

Além da interação entre usuários, as redes sociais de nicho também contribuem para a preservação e disseminação do conhecimento produzido pela comunidade. Por meio delas, jogadores e mestres podem compartilhar sistemas próprios, regras adaptadas, campanhas autorais, materiais de apoio e conteúdos educativos relacionados ao RPG. Esse compartilhamento constante favorece a produção coletiva de conhecimento e fortalece a identidade da comunidade.

Nesse cenário, observa-se uma crescente demanda por plataformas que integrem recursos sociais e ferramentas especializadas em um único ambiente digital. A centralização dessas funcionalidades pode contribuir para melhorar a experiência dos usuários, reduzir a fragmentação entre diferentes serviços e ampliar o acesso às

atividades relacionadas ao RPG de mesa, aspecto diretamente relacionado à proposta desenvolvida neste trabalho.

Plataformas VTT no cenário atual

O *Virtual Tabletop* (VTT) consiste em plataformas digitais desenvolvidas para simular a experiência do RPG de mesa em ambiente virtual. Essas ferramentas reproduzem elementos tradicionais do jogo presencial, como mapas, fichas de personagem, rolagem de dados e comunicação em tempo real.

Segundo Santos, Sant'Anna e Paim (2023), as plataformas digitais especializadas ampliaram a acessibilidade do RPG, permitindo que jogadores participem de campanhas independentemente da localização física. A incorporação de recursos como rolagem automática de dados e compartilhamento de mapas interativos demonstra a adaptação tecnológica do jogo às demandas contemporâneas.

Silva e Souza (2020) observam que o uso dessas ferramentas não descaracteriza o RPG tradicional, mas representa uma extensão de suas possibilidades narrativas e organizacionais. O ambiente virtual mantém a estrutura colaborativa do jogo, ao mesmo tempo em que oferece suporte técnico que facilita a condução das sessões.

Entre as plataformas mais conhecidas estão Roll20 e *Foundry Virtual Tabletop*, que disponibilizam recursos integrados para mestres e jogadores. Tais sistemas evidenciam a consolidação do VTT como solução específica para o nicho do RPG de mesa.

Dessa forma, o *Virtual Tabletop* configura-se como uma evolução do RPG tradicional, integrando narrativa colaborativa e infraestrutura digital, consolidando-se como elemento central na adaptação do jogo à cultura digital.

Justificativa

O RPG de mesa consolidou-se como prática cultural relevante, caracterizada pela construção coletiva de narrativas, desenvolvimento da criatividade e fortalecimento da interação social. Conforme discutido por Chiappa e Ribeiro (2023), o RPG envolve interpretação de personagens, cooperação entre participantes e mediação por regras estruturadas, constituindo-se como atividade que estimula habilidades cognitivas e sociais.

Além disso, estudos recentes apontam que o RPG passou por um processo de adaptação ao ambiente digital, ampliando seu alcance por meio de ferramentas on-line e plataformas especializadas (SANTOS; SANT'ANNA; PAIM, 2023). Essa transformação tecnológica possibilitou a continuidade da prática mesmo em contextos de distanciamento físico, demonstrando sua capacidade de adaptação às novas dinâmicas sociais.

Diante desse cenário, torna-se relevante investigar o RPG de mesa sob a perspectiva histórica, social e tecnológica, compreendendo sua evolução e seu impacto na formação de comunidades de nicho. O presente trabalho justifica-se pela necessidade de sistematizar essas informações e analisar como o RPG mantém sua essência colaborativa ao mesmo tempo em que incorpora recursos digitais contemporâneos.

Assim, o estudo contribui para a compreensão do RPG como fenômeno cultural, educacional e tecnológico, evidenciando sua importância no contexto atual.

Metodologia de Desenvolvimento

A metodologia adotada neste trabalho caracteriza-se como pesquisa bibliográfica, de natureza qualitativa, fundamentada na análise de livros e artigos científicos relacionados ao RPG de mesa, sua história, seu impacto social e sua adaptação ao ambiente digital.

Segundo os princípios da pesquisa bibliográfica, esse tipo de abordagem permite a sistematização de conhecimentos já produzidos, possibilitando a construção de um referencial teórico consistente a partir de fontes confiáveis e reconhecidas academicamente.

Foram analisadas obras que abordam a origem do RPG, como estudos históricos sobre o surgimento do sistema *Dungeons & Dragons*, bem como artigos que discutem sua evolução para o ambiente digital e o uso de ferramentas on-line. Também foram considerados trabalhos que tratam do impacto social do RPG e da formação de comunidades digitais especializadas.

A análise dos materiais ocorreu por meio da leitura interpretativa e comparação entre autores, buscando identificar convergências e complementaridades teóricas. A partir dessa análise, os conteúdos foram organizados em seções temáticas, estruturando o referencial teórico do trabalho.

Dessa forma, a metodologia permitiu fundamentar teoricamente a pesquisa e garantir embasamento científico às discussões apresentadas.

1. NICHOS DE MERCADO DO PROJETO

O projeto foca no nicho RPG de mesa (jogos virtuais/online), tendo a proposta de criar uma rede social com gerenciamento de mesas de RPG simultâneas em comunidades para criar um ambiente acolhedor e que despertem nos jogadores atuais pela falta de uma rede referente nesse nicho, a vontade de querer conciliar as ferramentas virtuais de ligação, gerenciamento de mesa, organização de informações, comunhão de comunidades para diferentes sistemas com sistemas básicos de gerenciar perfil, amizades, e encontrar pessoas com os mesmos gostos tudo em apenas um único lugar.

Com a alta de jogadores de RPG frequentes derivadas principalmente do sucesso recente de sistemas como Ordem Paranormal, o projeto chega como um refúgio e lugar onde novos jogadores ainda inexperientes e até mesmos seus veteranos consigam ter acesso a todas as ferramentas necessárias para suas atividades frequentes nesse universo e ter uma solução para um problema neste nicho que não teve uma resposta ou solução até hoje, igual McDonald's em seu início conciliou e popularizou a ideia de restaurante com lanches e bebidas, conhecido como combo meal, algo revolucionário em sua época, a Ruby RPG busca criar algo inovador que deixará seu legado nesse nicho de mercado, sendo a primeira a fazer este conjunto de ferramentas.

1.1 Público-alvo

O público-alvo do projeto é composto por jogadores e mestres de RPG de mesa que utilizam ferramentas digitais para organizar e participar de campanhas online. Incluem-se nesse grupo tanto usuários iniciantes quanto praticantes experientes, pertencentes em sua maioria de 16 a 30 anos, dados provenientes da pesquisa de campo aplicada pela equipe Panteon Table para tal projeto.

Também fazem parte do público-alvo criadores de conteúdo, desenvolvedores de sistemas próprios de RPG e integrantes de comunidades especializadas que buscam ambientes digitais voltados à interação social e ao compartilhamento de informações relacionadas ao hobby.

A crescente adoção de plataformas digitais para a realização de sessões remotas ampliou significativamente o número de usuários que dependem de recursos

tecnológicos para organizar suas atividades. Dessa forma, o projeto busca atender indivíduos que necessitam de uma solução centralizada para gerenciamento de campanhas, comunicação entre participantes e compartilhamento de materiais relacionados ao RPG.

1.2 Problema

O RPG de mesa consolidou-se como prática cultural baseada na interação social, na construção coletiva de narrativas e na cooperação entre participantes. Entretanto, com o avanço das tecnologias digitais e o crescimento das plataformas on-line, essa prática passou por transformações significativas, especialmente no que se refere à forma de organização das sessões e à mediação da interação entre os jogadores.

A adaptação do RPG ao ambiente digital, por meio do uso de ferramentas on-line e plataformas de Virtual Tabletop (VTT), ampliou seu alcance e acessibilidade. Contudo, essa transição levanta questionamentos acerca da preservação de suas características fundamentais, como a interação presencial, a espontaneidade narrativa e a dinâmica social tradicional.

Dessa forma, surge a seguinte problemática: **“de que maneira o RPG de mesa consegue manter sua essência narrativa, colaborativa, comunicativa e social ao mesmo tempo em que se adapta às transformações tecnológicas e às plataformas digitais contemporâneas?”**

A partir dessa questão central, o presente trabalho busca analisar as mudanças estruturais ocorridas no RPG, investigando seu percurso histórico, seu impacto social e sua consolidação no ambiente digital, especialmente no contexto das redes sociais de nicho e das plataformas especializadas.

1.3 Solução Proposta

A solução proposta consiste no desenvolvimento da plataforma Ruby RPG, uma rede social especializada voltada à comunidade de RPG de mesa. O sistema tem como objetivo integrar recursos de interação social, gerenciamento de campanhas e ferramentas de apoio às sessões virtuais em um único ambiente digital.

A plataforma permitirá que usuários criem perfis personalizados, estabeleçam conexões com outros jogadores, participem de comunidades temáticas e organizem campanhas de RPG por meio de funcionalidades específicas para esse segmento.

Além disso, serão disponibilizados recursos para gerenciamento de fichas, compartilhamento de informações e comunicação entre participantes.

Considerando o tempo disponível para desenvolvimento e os recursos da equipe, o escopo inicial do projeto foi definido com foco na implementação das funcionalidades essenciais para validação da proposta. Dessa forma, a versão apresentada neste trabalho concentra-se principalmente nos recursos fundamentais de rede social e gerenciamento de fichas de personagem, permitindo demonstrar a viabilidade técnica da solução e estabelecer uma base para futuras expansões.

2. TECNOLOGIAS UTILIZADAS

Para o desenvolvimento da plataforma Ruby RPG | Panteon Table, foi adotado um conjunto de tecnologias modernas voltadas ao desenvolvimento de aplicações web escaláveis, responsivas e interativas. A seleção dessas ferramentas considerou aspectos relacionados à performance, segurança, manutenção do código, experiência do usuário e integração entre os diferentes componentes do sistema. Segundo Sommerville (2011), a escolha adequada das tecnologias é um fator determinante para a qualidade e sustentabilidade de um software ao longo de seu ciclo de vida.

2.1 Linguagem de Programação

TypeScript

O TypeScript foi utilizado como linguagem principal de desenvolvimento em toda a aplicação, tanto no frontend quanto no backend. Desenvolvido pela Microsoft, o TypeScript expande as funcionalidades do JavaScript ao adicionar tipagem estática, permitindo maior controle sobre estruturas de dados, funções e componentes da aplicação.

Sua utilização contribui para a redução de erros durante o desenvolvimento, melhora a legibilidade do código e facilita a manutenção de projetos de grande porte. Além disso, os mecanismos de verificação em tempo de compilação auxiliam na identificação precoce de falhas, reduzindo custos de correção durante o ciclo de desenvolvimento do software (SOMMERVILLE, 2011).

2.2 Tecnologias e Frameworks Front-end

2.2.1 React

O React é uma biblioteca JavaScript voltada para a construção de interfaces de usuário baseadas em componentes reutilizáveis. Sua arquitetura permite dividir a interface em elementos independentes, facilitando a manutenção e a escalabilidade do sistema.

No projeto Ruby RPG, o React foi empregado na construção das telas da plataforma, incluindo sistemas de perfil, comunidades, chats, publicações, gerenciamento de fichas e demais recursos disponíveis aos usuários. Sua utilização contribui para a atualização dinâmica dos elementos da interface sem a necessidade

de recarregar toda a página, proporcionando uma experiência mais fluida ao usuário (CHIAPPA; RIBEIRO, 2023).

2.2.2 Vite

O Vite é uma ferramenta moderna de desenvolvimento e empacotamento de aplicações web. Seu principal diferencial consiste na velocidade de inicialização do ambiente de desenvolvimento e no sistema de Hot Module Replacement (HMR), que atualiza instantaneamente as alterações realizadas no código sem necessidade de recompilação completa da aplicação.

Essa característica contribui para aumentar a produtividade da equipe durante o desenvolvimento e otimizar o processo de construção da aplicação para ambiente de produção.

2.2.3 Tailwind CSS

O Tailwind CSS é um framework de estilização baseado no conceito utility-first, permitindo que os estilos sejam aplicados diretamente nos elementos da interface por meio de classes utilitárias.

Conforme estudado nos fundamentos de CSS, a separação entre estrutura e apresentação é um princípio fundamental do desenvolvimento web (SILVA, 2024). Nesse contexto, o Tailwind CSS possibilita a construção de interfaces modernas, responsivas e padronizadas, reduzindo a necessidade de extensos arquivos de estilização personalizados.

2.3 Tecnologias e Arquitetura Back-end

2.2.1 Node.js

O Node.js é um ambiente de execução JavaScript que permite executar código fora do navegador, possibilitando o desenvolvimento de aplicações do lado do servidor.

Sua arquitetura orientada a eventos e baseada em operações assíncronas oferece elevado desempenho em aplicações que necessitam processar múltiplas requisições simultaneamente. No projeto Ruby RPG, o Node.js foi utilizado como base para toda a infraestrutura do servidor responsável pelo gerenciamento das informações da plataforma.

2.3.2 Express

O Express é um framework minimalista para Node.js utilizado para a criação de APIs e gerenciamento das rotas da aplicação. Sua estrutura simplificada permite organizar as funcionalidades do sistema em módulos independentes, facilitando a manutenção e expansão do software.

No Ruby RPG, o Express é responsável pelo gerenciamento das requisições HTTP, autenticação de usuários, controle de sessões, processamento de uploads e comunicação entre cliente e servidor.

2.3.3 TSX

O TSX é uma ferramenta utilizada para executar arquivos TypeScript diretamente durante o desenvolvimento, dispensando etapas intermediárias de compilação. Sua utilização acelera os testes e a depuração do backend, contribuindo para maior produtividade da equipe de desenvolvimento.

2.4 Banco de Dados e Gerenciamento de Arquivos

2.4.1 MySQL

O MySQL é um Sistema Gerenciador de Banco de Dados Relacional (SGBDR) amplamente utilizado em aplicações corporativas e sistemas web. Sua principal função é armazenar, organizar e recuperar informações de forma estruturada e segura.

Segundo Date (2012), os bancos de dados relacionais permitem garantir integridade, consistência e eficiência no gerenciamento das informações. No projeto Ruby RPG, o MySQL é responsável pelo armazenamento dos dados dos usuários, comunidades, campanhas, fichas de personagens, mensagens, publicações e demais registros necessários ao funcionamento da plataforma.

2.4.2 MySQL2 Connector

O MySQL2 é uma biblioteca utilizada para estabelecer a comunicação entre a aplicação Node.js e o banco de dados MySQL. Sua implementação possibilita a execução de consultas SQL, procedimentos de inserção, atualização e remoção de dados, além do gerenciamento eficiente das conexões com o banco de dados.

2.4.3 Multer

O Multer é um middleware especializado no processamento de requisições do tipo multipart/form-data, sendo utilizado para realizar uploads de arquivos enviados pelos usuários.

Por meio dessa ferramenta, a plataforma possibilita o envio de imagens de perfil, tokens de personagens, documentos PDF, mapas e demais arquivos relacionados às campanhas de RPG.

2.5 Bibliotecas de Integração, Segurança e Interface

2.5.1 Google Generative AI SDK

A biblioteca Google Generative AI SDK foi utilizada para integrar recursos de inteligência artificial generativa baseados nos modelos Gemini. Essa integração permite a implementação futura de ferramentas de auxílio à criação de narrativas, geração de personagens, sugestões de aventuras e suporte aos mestres durante a condução das campanhas.

2.5.2 Framer Motion

O Framer Motion é uma biblioteca especializada na criação de animações para aplicações React. Sua utilização possibilita implementar transições suaves, efeitos visuais, movimentação de elementos e interações avançadas na interface do usuário.

Esses recursos contribuem para aumentar a sensação de fluidez da navegação e proporcionar maior imersão ao ambiente virtual da plataforma.

2.5.3 Nodemailer

O Nodemailer é uma biblioteca para Node.js destinada ao envio automatizado de correios eletrônicos. No sistema Ruby RPG, essa ferramenta é utilizada para envio de mensagens relacionadas à recuperação de senhas, validação de contas e notificações importantes aos usuários.

2.5.4 Bcrypt.js

A segurança das credenciais dos usuários é realizada por meio do Bcrypt.js, biblioteca responsável pela geração de hashes criptográficos das senhas antes de seu armazenamento no banco de dados.

Segundo Sommerville (2011), a proteção das informações dos usuários constitui um dos pilares fundamentais da segurança em sistemas computacionais. Dessa forma, a utilização de algoritmos criptográficos reduz significativamente os riscos relacionados ao vazamento de credenciais.

2.5.5 Lucide React

O Lucide React é uma biblioteca de ícones vetoriais utilizada na construção da interface gráfica da plataforma. Sua adoção contribui para a padronização visual do sistema e para a melhoria da usabilidade dos recursos disponibilizados aos usuários.

2.5.6 React Hot Toast

O React Hot Toast é uma biblioteca responsável pela exibição de notificações em tempo real na interface da aplicação. Sua utilização permite informar ao usuário o sucesso ou falha de ações realizadas no sistema, contribuindo para uma experiência mais intuitiva e interativa.

As tecnologias selecionadas permitiram o desenvolvimento de uma plataforma moderna, responsiva e escalável, alinhada às necessidades da comunidade de RPG de mesa. A integração entre *frontend*, *backend*, banco de dados e bibliotecas especializadas possibilitou a construção de um ambiente capaz de centralizar funcionalidades sociais e ferramentas de apoio às campanhas de RPG em um único sistema.

3. EMPRESA

A Panteon Table é uma equipe inovadora dedicada ao universo do RPG de mesa, formada por 4 integrantes do curso técnico de informática da instituição Etec Adolpho Berezin gerida pelo Centro Paula Souza. Criada por uma equipe apaixonada por jogos narrativos, tecnologia e criação de comunidades, nossa empresa busca unificar, fortalecer e expandir o cenário do RPG, oferecendo aos jogadores, mestres e criadores um ecossistema completo, em busca de monopolizar esse novo nicho de mercado não explorado pelas empresas atuais que são especializadas em modelos VTT & RPG de mesa virtual, sendo o principal projeto da empresa para concretizar esse ideal a Ruby RPG, que terá as seguintes funcionalidades: rede social, ferramentas de mesa virtual (VTT)

3.1 Missão

Nossa proposta é oferecer um espaço onde interação, criatividade e imersão se conectam — tudo em um único lugar.

3.2 Visão

Proporcionar um ambiente digital completo e acessível que reúna, fortaleça e amplie a comunidade de RPG de mesa, oferecendo ferramentas, conteúdos e recursos especializados que permitam que jogadores e mestres desenvolvam suas experiências, criem conexões e consolidem o RPG em seu próprio espaço.

3.3 Valores

- Comunidade: Fortalecer o senso de pertencimento entre jogadores, mestres e criadores, incentivando colaboração e respeito.
- Criatividade: Promover ferramentas e espaços que estimulem imaginação, narrativa e liberdade criativa.
- Acessibilidade: Garantir que pessoas de diferentes regiões, níveis de conhecimento e condições possam acessar e desfrutar do universo do RPG.
- Inovação: Integrar tecnologia moderna para aprimorar a experiência de RPG, desde rolagem de dados sincronizada até mesas virtuais completas.
- Transparência: Agir com clareza, honestidade e responsabilidade em todas as relações e processos.

- **Diversidade e Inclusão:** Valorizar diferentes estilos de jogo, temáticas, sistemas e grupos, reforçando um ambiente acolhedor para todos.
- **Qualidade:** Oferecer produtos, ferramentas e conteúdos confiáveis, organizados e preparados para elevar a experiência dos usuários.

3.4 Logos

3.4.1 Logo da Equipe



Figura 1 - Logo da Equipe

Fonte: Criado pela equipe

3.4.2 Logo do Projeto



Figura 2 - Logo do Projeto

Fonte: Criado pela equipe

3.5 Slogan da empresa

“Todos os deuses, todos os jogadores, todos os mundos, uma só mesa, uma só empresa!”

3.6 Slogan do projeto

“A sua rede social para a sua mesa de RPG!”

4. ANÁLISE

4.1. Descrição das funcionalidades

4.1.1 Módulo de Rede Social e Comunidade

Este módulo foca na interação entre jogadores e na construção de um ecossistema para o RPG.

- **Perfil do Usuário:** Personalização de avatar, biografia, sistemas de RPG favoritos, storys, histórico de campanhas.
- **Sistema de Amizades:** Busca, adição e gerenciamento de lista de amigos.
- **Notificações em Tempo Real:** Alertas de convites para mesas, novas mensagens e lembretes de sessões.
- **Comunidades (Estilo Servidores):** Criação de espaços dedicados a grupos ou sistemas específicos, com divisões por canais.
- **Canais de Texto e Voz:** Espaços para discussão de regras, *lore* ou conversas casuais.
- **Mensagens Diretas (DMs):** Chat privado entre usuários com suporte a envio de referências de fichas e crônicas.
- **Feed de Crônicas:** Um mural onde mestres e jogadores podem postar "diários de campanha" com imagens e textos, visíveis para a comunidade ou apenas para o grupo.

4.1.2 Módulo de Gerenciamento e Preparação (O "Hub")

Ferramentas para organizar o jogo antes de ele começar.

- **Dashboard Dinâmico:** Painel central que mostra as mesas ativas, as próximas sessões e atividades recentes dos amigos.
- **Calendário de Sessões:** Agendamento de partidas com confirmação de presença e integração de fuso horário.
- **Explorador de Sistemas:** Biblioteca para navegar por sistemas de RPG existentes (D&D, Ordem Paranormal, Tormenta, etc.).
- **Localizador de Mesas (LFG):** Filtro para encontrar mesas que precisam de jogadores, baseado em sistema, estilo de jogo e horário.

- **Construtor de Sistemas (System Builder):** Ferramenta *no-code* que permite ao usuário criar seu próprio sistema de RPG do zero, definindo atributos, perícias e fórmulas de cálculo.
- **Database Schema:** Visualização técnica da estrutura de dados para usuários avançados que desejam entender como as fichas são processadas.

4.1.3 Módulo de Mesa Virtual (VTT - Virtual Tabletop)

O coração da plataforma, onde o jogo acontece em tempo real.

- **Mapa Interativo:** Upload de mapas com suporte a zoom, pan e grade (grid) ajustável.
- **Rolador de Dados:** Chat integrado que processa comandos de dados (ex: /r 1d20+5) com histórico visível para todos.
- **Painel de Notas:** Bloco de notas para jogadores registrarem pistas e informações importantes.
- **Fichas de Personagem Integradas:** Fichas que conversam com o mapa; ao clicar em uma perícia na ficha, o dado é rolado automaticamente no chat.
- **Jukebox (Áudio Compartilhado):** Sistema de som onde o mestre controla a trilha sonora e efeitos ambientais para todos os jogadores simultaneamente.
- **Gerenciador de Tokens:** Colocação de imagens representativas de personagens e monstros sobre o mapa.
- **Sincronização Multi-Usuário (Socket.io):** Movimentação de tokens e atualizações de vida/status que aparecem instantaneamente para todos os jogadores sem atraso.
- **Névoa de Guerra (Fog of War):** O mestre controla quais partes do mapa os jogadores podem ver, revelando o cenário conforme eles exploram.
- **Painel do Mestre:** Ferramentas exclusivas para o narrador gerenciar NPCs, esconder rolagens de dados e controlar o fluxo da narrativa.

4.1.4 Funcionalidades de Sistema e Customização

Recursos que garantem a usabilidade e a modernidade da plataforma.

- **Temas Dark e Light:** Interface adaptável para conforto visual em diferentes ambientes.
- **Design Responsivo:** O site funciona perfeitamente em desktops, tablets e smartphones.
- **Login Social (Google OAuth):** Acesso rápido e seguro utilizando a conta do Google.
- **Armazenamento em Nuvem (Blob Storage):** Salvamento automático de mapas, imagens de tokens e fichas, acessíveis de qualquer lugar.

4.2. MER – Modelo do Banco de Dados

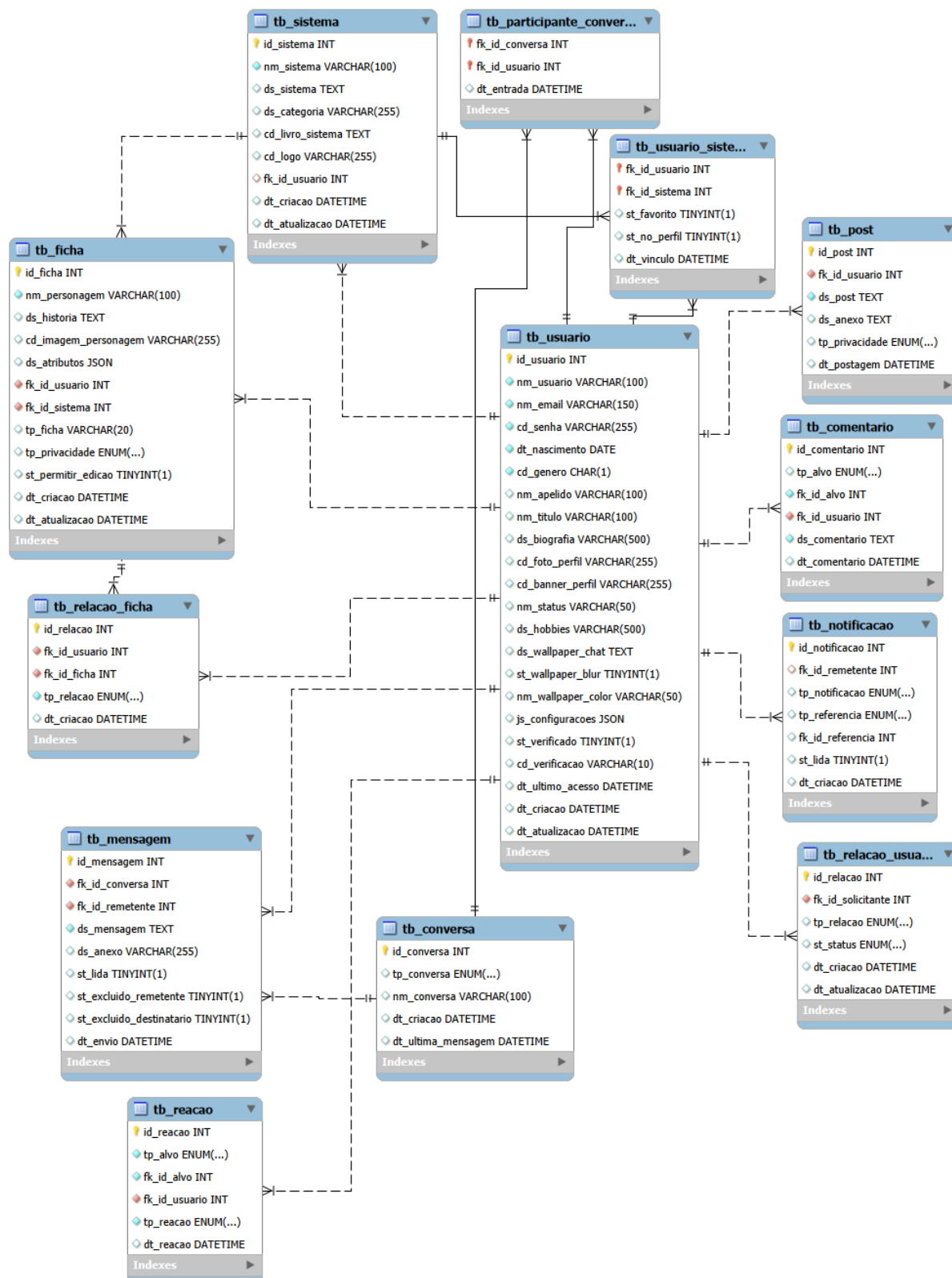


Figura 3 - MER

Fonte: Criado pela equipe

4.3. Create do Banco de Dados

```
CREATE DATABASE IF NOT EXISTS db_ruby_rpg;
```

```
USE db_ruby_rpg;
```

```
-- ♦ TB_USUARIO: Modelo Unificado (Perfil + Configurações Integrados)
```

```
CREATE TABLE IF NOT EXISTS tb_usuario (
```

```
    id_usuario INT PRIMARY KEY AUTO_INCREMENT,
```

```
    nm_usuario VARCHAR(100) NOT NULL,
```

```
    nm_email VARCHAR(150) NOT NULL UNIQUE,
```

```
    cd_senha VARCHAR(255) NOT NULL,
```

```
    dt_nascimento DATE NOT NULL,
```

```
    cd_genero CHAR(1) NOT NULL,
```

```
-- Perfil
```

```
    nm_apelido VARCHAR(100),
```

```
    nm_titulo VARCHAR(100),
```

```
    ds_biografia VARCHAR(500),
```

```
    cd_foto_perfil          VARCHAR(255)          DEFAULT
```

```
'https://api.dicebear.com/7.x/avataaars/svg?seed=default',
```

```
    cd_banner_perfil VARCHAR(255),
```

```
    nm_status VARCHAR(50) DEFAULT 'explorando',
```

```
    ds_hobbies VARCHAR(500),
```

```
    ds_wallpaper_chat TEXT,
```

```
    st_wallpaper_blur BOOLEAN DEFAULT FALSE,
```

```
    nm_wallpaper_color VARCHAR(50) DEFAULT '#000000',
```

```
-- Configurações (JSON)
```

```
    js_configuracoes JSON,
```

```
-- Sistema
```

```
    st_verificado BOOLEAN DEFAULT FALSE,
```

```
    cd_verificacao VARCHAR(10),
```

```
    dt_ultimo_acesso DATETIME DEFAULT CURRENT_TIMESTAMP,
```

```

        dt_criacao DATETIME DEFAULT CURRENT_TIMESTAMP,
        dt_atualizacao DATETIME DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP
    ) ENGINE=InnoDB;

```

```

-- ♦ TB_RELACAO_USUARIO (Amizade, Bloqueio, Seguindo)
CREATE TABLE IF NOT EXISTS tb_relacao_usuario (
    id_relacao INT PRIMARY KEY AUTO_INCREMENT,
    fk_id_usuario_1 INT NOT NULL, -- Menor ID para amizade 1-1
    fk_id_usuario_2 INT NOT NULL, -- Maior ID para amizade 1-1
    fk_id_solicitante INT NOT NULL,
    tp_relacao ENUM('amigo', 'bloqueio', 'seguindo') DEFAULT 'amigo',
    st_status ENUM('pendente', 'aceito', 'recusado') DEFAULT 'pendente',
    dt_criacao DATETIME DEFAULT CURRENT_TIMESTAMP,
    dt_atualizacao DATETIME DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
    UNIQUE(fk_id_usuario_1, fk_id_usuario_2, tp_relacao),
    FOREIGN KEY (fk_id_usuario_1) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE,
    FOREIGN KEY (fk_id_usuario_2) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE,
    FOREIGN KEY (fk_id_solicitante) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE
    ) ENGINE=InnoDB;

```

```

-- ♦ TB_NOTIFICACAO
CREATE TABLE IF NOT EXISTS tb_notificacao (
    id_notificacao INT PRIMARY KEY AUTO_INCREMENT,
    fk_id_usuario INT NOT NULL,
    fk_id_remetente INT,
    tp_notificacao ENUM('like', 'comentario', 'amizade', 'mensagem', 'sistema',
'post'),

```

```

        tp_referencia ENUM('post', 'comentario', 'ficha', 'sistema', 'conversa',
'usuario'),
        fk_id_referencia INT,
        st_lida BOOLEAN DEFAULT FALSE,
        dt_criacao DATETIME DEFAULT CURRENT_TIMESTAMP,
        FOREIGN KEY (fk_id_usuario) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE,
        FOREIGN KEY (fk_id_remetente) REFERENCES tb_usuario(id_usuario) ON
DELETE SET NULL
    ) ENGINE=InnoDB;

```

```
-- ♦ TB_CONVERSA
```

```

CREATE TABLE IF NOT EXISTS tb_conversa (
    id_conversa INT PRIMARY KEY AUTO_INCREMENT,
    tp_conversa ENUM('individual', 'grupo') DEFAULT 'individual',
    nm_conversa VARCHAR(100),
    dt_criacao DATETIME DEFAULT CURRENT_TIMESTAMP,
    dt_ultima_mensagem DATETIME DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP
) ENGINE=InnoDB;

```

```
-- ♦ TB_PARTICIPANTE_CONVERSA
```

```

CREATE TABLE IF NOT EXISTS tb_participante_conversa (
    fk_id_conversa INT NOT NULL,
    fk_id_usuario INT NOT NULL,
    dt_entrada DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (fk_id_conversa, fk_id_usuario),
    FOREIGN KEY (fk_id_conversa) REFERENCES tb_conversa(id_conversa)
ON DELETE CASCADE,
    FOREIGN KEY (fk_id_usuario) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE
) ENGINE=InnoDB;

```

```

-- ♦ TB_MENSAGEM
CREATE TABLE IF NOT EXISTS tb_mensagem (
    id_mensagem INT PRIMARY KEY AUTO_INCREMENT,
    fk_id_conversa INT NOT NULL,
    fk_id_remetente INT NOT NULL,
    ds_mensagem TEXT NOT NULL,
    ds_anexo VARCHAR(255),
    st_lida BOOLEAN DEFAULT FALSE,
    st_excluido_remetente BOOLEAN DEFAULT FALSE,
    st_excluido_destinatario BOOLEAN DEFAULT FALSE,
    dt_envio DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (fk_id_conversa) REFERENCES tb_conversa(id_conversa)
ON DELETE CASCADE,
    FOREIGN KEY (fk_id_remetente) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE
) ENGINE=InnoDB;

-- ♦ TB_SISTEMA
CREATE TABLE IF NOT EXISTS tb_sistema (
    id_sistema INT PRIMARY KEY AUTO_INCREMENT,
    nm_sistema VARCHAR(100) NOT NULL,
    ds_sistema TEXT,
    ds_categoria VARCHAR(255),
    cd_livro_sistema TEXT,
    cd_logo VARCHAR(255),
    fk_id_usuario INT, -- Criador
    dt_criacao DATETIME DEFAULT CURRENT_TIMESTAMP,
    dt_atualizacao DATETIME DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (fk_id_usuario) REFERENCES tb_usuario(id_usuario) ON
DELETE SET NULL
) ENGINE=InnoDB;

```

```

-- ♦ TB_FICHA
CREATE TABLE IF NOT EXISTS tb_ficha (
    id_ficha INT PRIMARY KEY AUTO_INCREMENT,
    nm_personagem VARCHAR(100) NOT NULL,
    ds_historia TEXT,
    cd_imagem_personagem VARCHAR(255),
    ds_atributos JSON,
    fk_id_usuario INT NOT NULL,
    fk_id_sistema INT NOT NULL,
    tp_ficha VARCHAR(20) DEFAULT 'personagem',
    tp_privacidade ENUM('publico', 'privado', 'amigos') DEFAULT 'publico',
    st_permitir_edicao BOOLEAN DEFAULT FALSE,
    dt_criacao DATETIME DEFAULT CURRENT_TIMESTAMP,
    dt_atualizacao DATETIME DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (fk_id_usuario) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE,
    FOREIGN KEY (fk_id_sistema) REFERENCES tb_sistema(id_sistema) ON
DELETE CASCADE
) ENGINE=InnoDB;

-- ♦ TB_RELACAO_FICHA (Unified: FAVORITO, CURTIDA, EDICAO,
VISUALIZACAO)
CREATE TABLE IF NOT EXISTS tb_relacao_ficha (
    id_relacao INT PRIMARY KEY AUTO_INCREMENT,
    fk_id_usuario INT NOT NULL,
    fk_id_ficha INT NOT NULL,
    tp_relacao ENUM('FAVORITO', 'CURTIDA', 'EDICAO', 'VISUALIZACAO')
NOT NULL,
    dt_criacao DATETIME DEFAULT CURRENT_TIMESTAMP,
    UNIQUE(fk_id_usuario, fk_id_ficha, tp_relacao),
    FOREIGN KEY (fk_id_usuario) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE,

```

```
FOREIGN KEY (fk_id_ficha) REFERENCES tb_ficha(id_ficha) ON DELETE
CASCADE
```

```
) ENGINE=InnoDB;
```

```
-- ♦ TB_POST
```

```
CREATE TABLE IF NOT EXISTS tb_post (
```

```
    id_post INT PRIMARY KEY AUTO_INCREMENT,
```

```
    fk_id_usuario INT NOT NULL,
```

```
    ds_post TEXT NOT NULL,
```

```
    ds_anexo TEXT,
```

```
    tp_privacidade ENUM('publico', 'privado', 'amigos') DEFAULT 'publico',
```

```
    dt_postagem DATETIME DEFAULT CURRENT_TIMESTAMP,
```

```
    FOREIGN KEY (fk_id_usuario) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE
```

```
) ENGINE=InnoDB;
```

```
-- ♦ TB_COMENTARIO (Posts, Fichas, etc)
```

```
CREATE TABLE IF NOT EXISTS tb_comentario (
```

```
    id_comentario INT PRIMARY KEY AUTO_INCREMENT,
```

```
    tp_alvo ENUM('post', 'ficha', 'comentario') DEFAULT 'post',
```

```
    fk_id_alvo INT NOT NULL,
```

```
    fk_id_usuario INT NOT NULL,
```

```
    ds_comentario TEXT NOT NULL,
```

```
    fk_id_parent INT DEFAULT NULL,
```

```
    dt_comentario DATETIME DEFAULT CURRENT_TIMESTAMP,
```

```
    FOREIGN KEY (fk_id_usuario) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE,
```

```
    FOREIGN KEY (fk_id_parent) REFERENCES tb_comentario(id_comentario)
ON DELETE SET NULL
```

```
) ENGINE=InnoDB;
```

```
-- ♦ TB_REACAO (Unified)
```

```
CREATE TABLE IF NOT EXISTS tb_reacao (
```

```

id_reacao INT PRIMARY KEY AUTO_INCREMENT,
tp_alvo ENUM('post', 'comentario', 'ficha') NOT NULL,
fk_id_alvo INT NOT NULL,
fk_id_usuario INT NOT NULL,
tp_reacao ENUM('like', 'love', 'haha', 'wow', 'sad', 'angry') NOT NULL,
dt_reacao DATETIME DEFAULT CURRENT_TIMESTAMP,
UNIQUE(tp_alvo, fk_id_alvo, fk_id_usuario),
FOREIGN KEY (fk_id_usuario) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE
) ENGINE=InnoDB;

```

```

-- ♦ TB_USUARIO_SISTEMA (Vínculos rápidos)
CREATE TABLE IF NOT EXISTS tb_usuario_sistema (
    fk_id_usuario INT NOT NULL,
    fk_id_sistema INT NOT NULL,
    st_favorito BOOLEAN DEFAULT FALSE,
    st_no_perfil BOOLEAN DEFAULT FALSE,
    dt_vinculo DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (fk_id_usuario, fk_id_sistema),
    FOREIGN KEY (fk_id_usuario) REFERENCES tb_usuario(id_usuario) ON
DELETE CASCADE,
    FOREIGN KEY (fk_id_sistema) REFERENCES tb_sistema(id_sistema) ON
DELETE CASCADE
) ENGINE=InnoDB;

```

4.4 Principais Selects do Banco de Dados

Select 1 – Consulta de Perfil de Usuário

Responsável por exibir as informações principais de um usuário cadastrado na plataforma.

```

SELECT
    id_usuario AS "ID",
    nm_usuario AS "Nome de Usuário",
    nm_apelido AS "Apelido",
    nm_email AS "E-mail",

```

```

    nm_titulo AS "Título",
    nm_status AS "Status",
    dt_criacao AS "Data de Cadastro"
FROM tb_usuario;

```

Select 2 – Feed de Publicações

Utilizado para exibir as publicações realizadas pelos usuários da rede social.

```

SELECT
    p.id_post AS "ID da Publicação",
    u.nm_usuario AS "Autor",
    p.ds_post AS "Conteúdo",
    p.tp_privacidade AS "Privacidade",
    p.dt_postagem AS "Data da Publicação"
FROM tb_post p
INNER JOIN tb_usuario u
ON p.fk_id_usuario = u.id_usuario;

```

Select 3 – Fichas de Personagens

Responsável por listar as fichas cadastradas na plataforma juntamente com seus respectivos sistemas.

```

SELECT
    f.id_ficha AS "ID da Ficha",
    f.nm_personagem AS "Personagem",
    s.nm_sistema AS "Sistema de RPG",
    f.tp_ficha AS "Tipo",
    f.tp_privacidade AS "Privacidade",
    f.dt_criacao AS "Data de Criação"
FROM tb_ficha f
INNER JOIN tb_sistema s
ON f.fk_id_sistema = s.id_sistema;

```

Select 4 – Sistemas Favoritados pelos Usuários

Consulta utilizada para identificar os sistemas marcados como favoritos.

```

SELECT
    u.nm_usuario AS "Usuário",
    s.nm_sistema AS "Sistema Favorito",
    us.dt_vinculo AS "Data do Vínculo"

```

```

FROM tb_usuario_sistema us
INNER JOIN tb_usuario u
ON us.fk_id_usuario = u.id_usuario
INNER JOIN tb_sistema s
ON us.fk_id_sistema = s.id_sistema
WHERE us.st_favorito = TRUE;

```

Select 5 – Conversas e Participantes

Responsável por exibir as conversas e seus respectivos participantes.

```

SELECT
    c.id_conversa AS "ID da Conversa",
    c.nm_conversa AS "Nome da Conversa",
    c.tp_conversa AS "Tipo",
    u.nm_usuario AS "Participante"
FROM tb_conversa c
INNER JOIN tb_participante_conversa pc
ON c.id_conversa = pc.fk_id_conversa
INNER JOIN tb_usuario u
ON pc.fk_id_usuario = u.id_usuario;

```

Select 6 – Mensagens Enviadas

Consulta utilizada para visualizar mensagens trocadas dentro das conversas.

```

SELECT
    m.id_mensagem AS "ID da Mensagem",
    u.nm_usuario AS "Remetente",
    c.nm_conversa AS "Conversa",
    m.ds_mensagem AS "Mensagem",
    m.dt_envio AS "Data de Envio"
FROM tb_mensagem m
INNER JOIN tb_usuario u
ON m.fk_id_remetente = u.id_usuario
INNER JOIN tb_conversa c
ON m.fk_id_conversa = c.id_conversa;

```

Select 7 – Notificações dos Usuários

Consulta responsável por exibir notificações geradas pelo sistema.

```

SELECT

```

```

n.id_notificacao AS "ID",
u.nm_usuario AS "Destinatário",
r.nm_usuario AS "Remetente",
n.tp_notificacao AS "Tipo",
n.tp_referencia AS "Referência",
n.st_lida AS "Lida",
n.dt_criacao AS "Data"
FROM tb_notificacao n
INNER JOIN tb_usuario u
ON n.fk_id_usuario = u.id_usuario
LEFT JOIN tb_usuario r
ON n.fk_id_remetente = r.id_usuario;

```

Select 8 – Ranking de Sistemas Mais Utilizados

Consulta estatística utilizada para identificar os sistemas com maior quantidade de fichas cadastradas.

```

SELECT
    s.nm_sistema AS "Sistema de RPG",
    COUNT(f.id_ficha) AS "Quantidade de Fichas"
FROM tb_sistema s
LEFT JOIN tb_ficha f
ON s.id_sistema = f.fk_id_sistema
GROUP BY s.id_sistema, s.nm_sistema
ORDER BY COUNT(f.id_ficha) DESC;

```

4.5. Wireframe das Telas

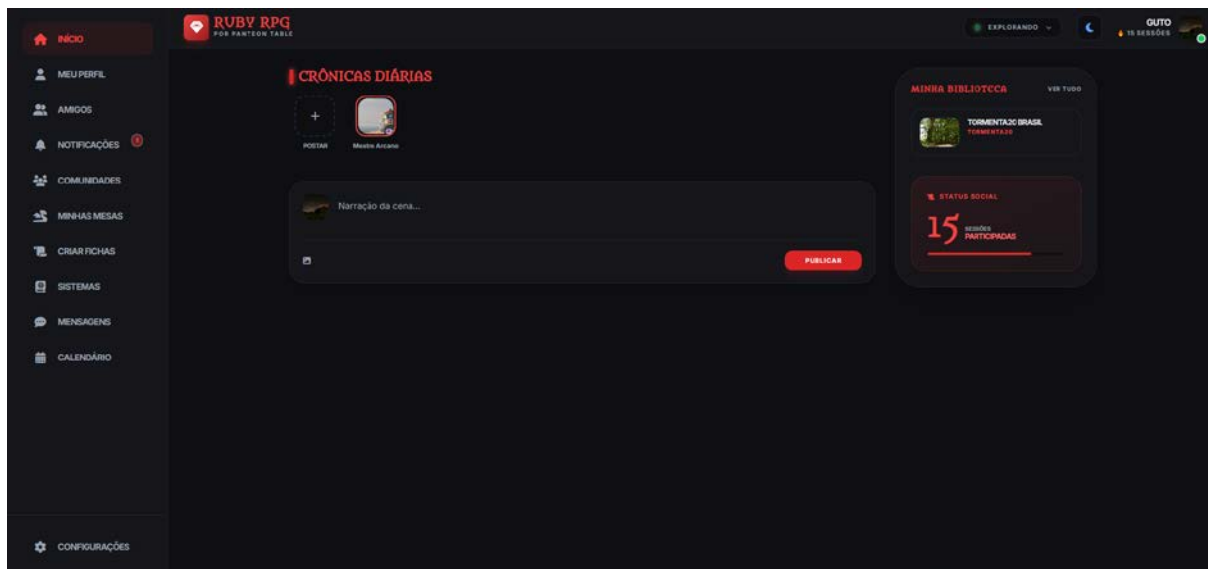


Figura 4 - Tela inicial (modo escuro)
Fonte: Wireframe das Telas

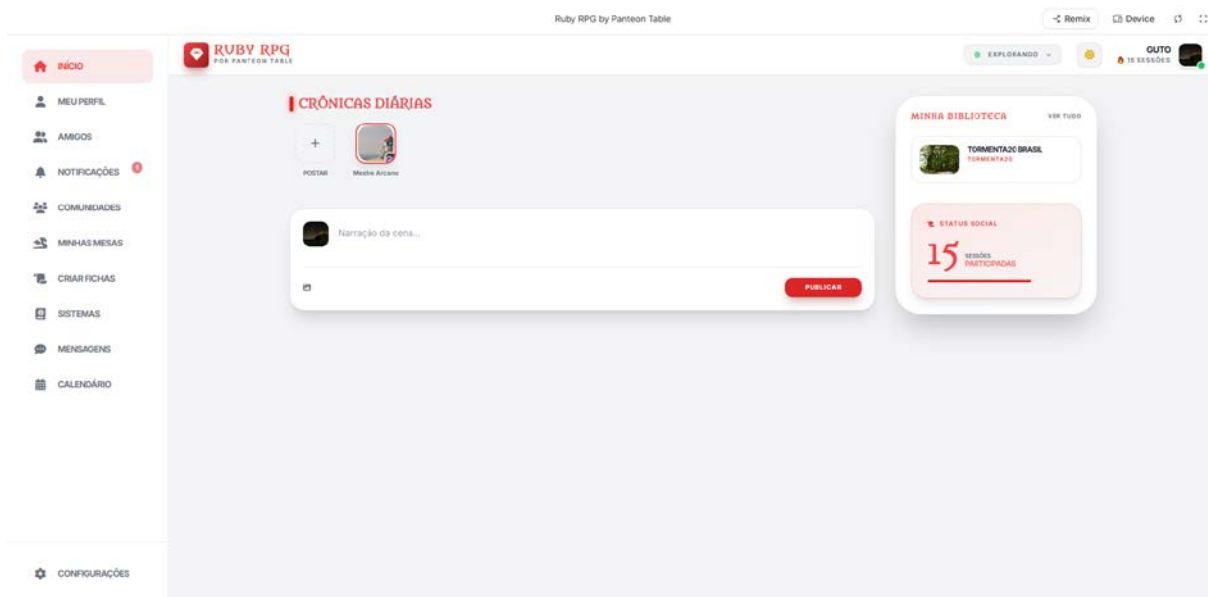


Figura 5 - Tela inicial (modo claro)
Fonte: Wireframe das telas

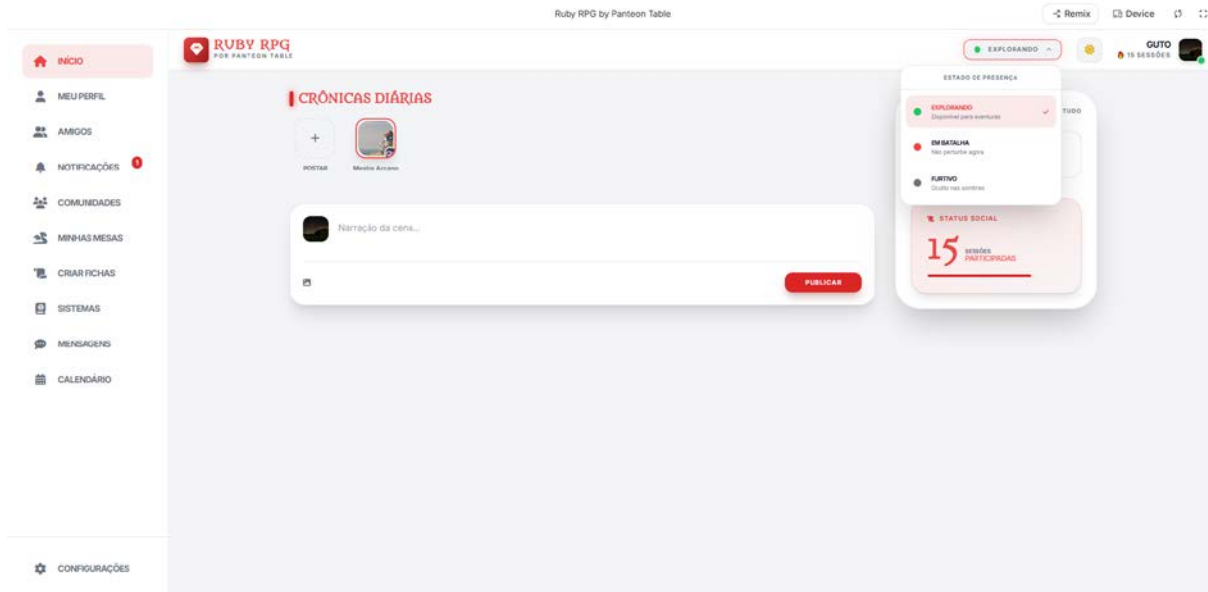


Figura 6 - Tela inicial (status de perfil)

Fonte: Wireframe das telas

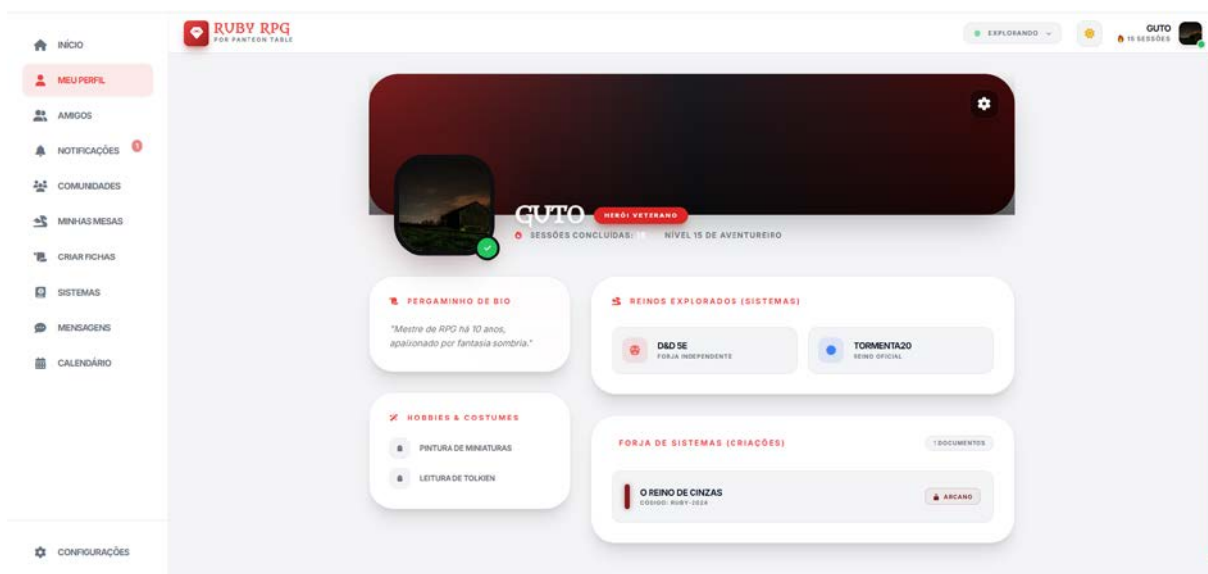


Figura 7 - Perfil do usuário

Fonte: Wireframe das telas

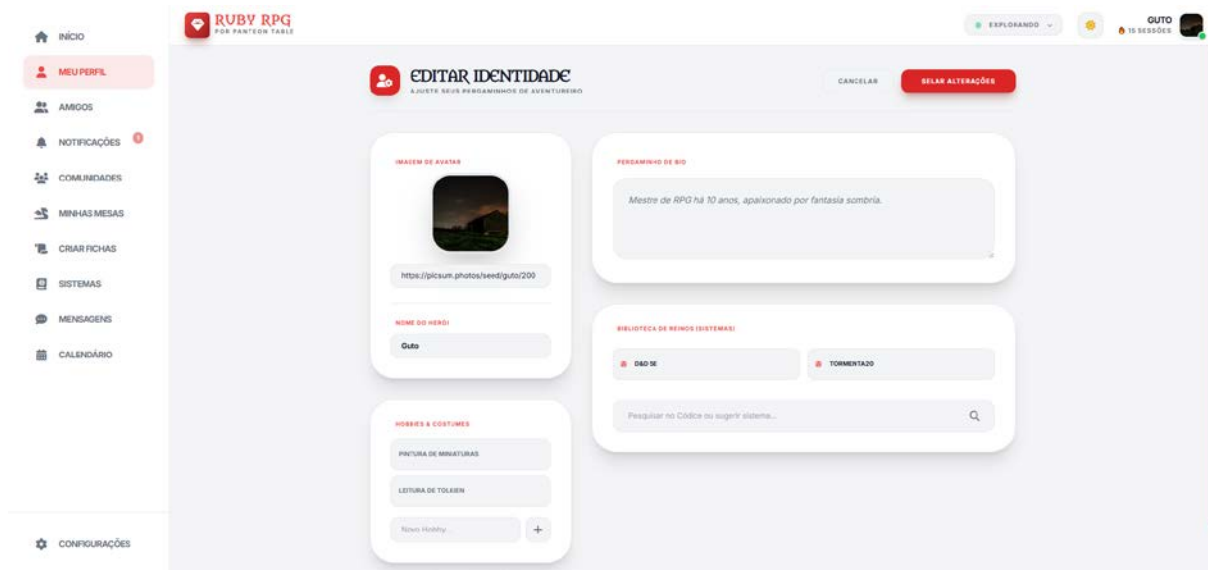


Figura 8 - Tela de configuração de perfil

Fonte: Wireframe das telas

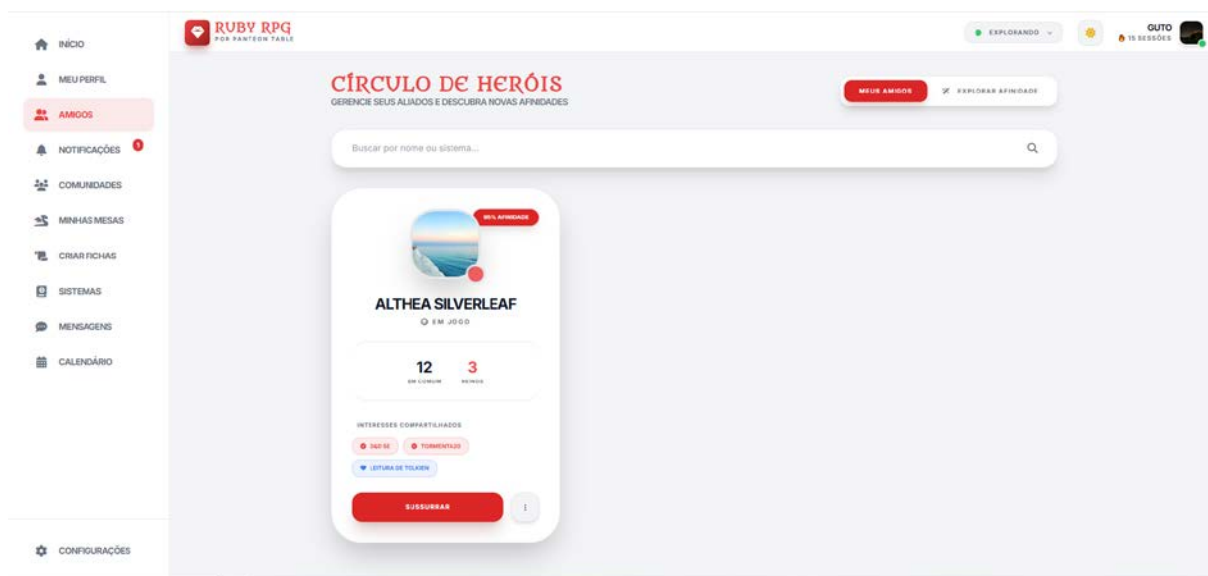


Figura 9 - Amigos (meus amigos)

Fonte: Wireframe das telas

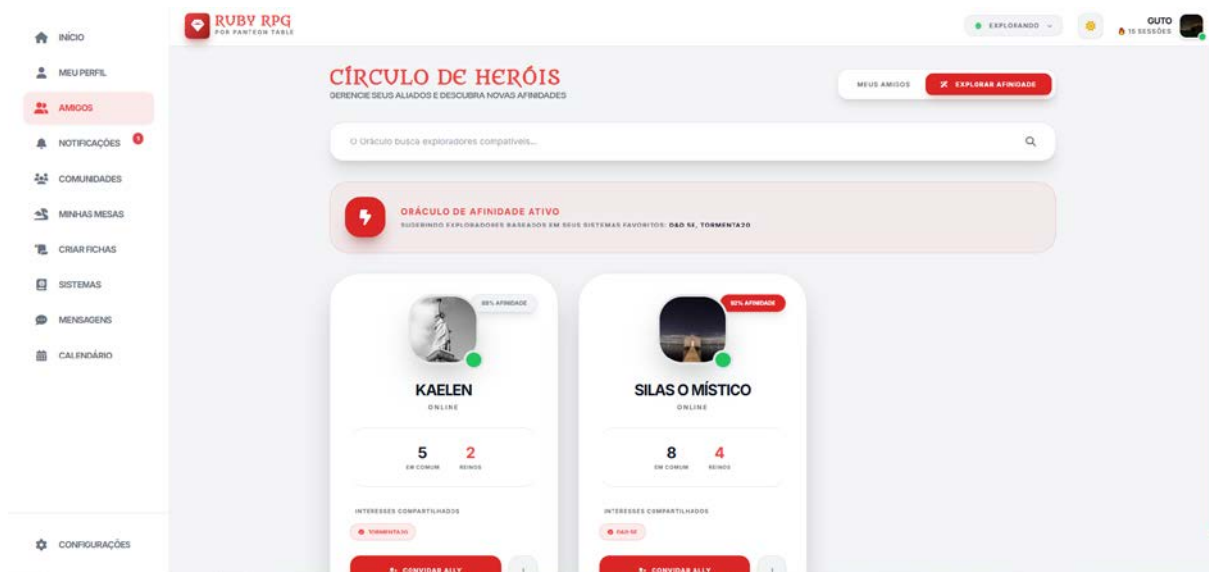


Figura 10 - Tela de migos (Explorar afinidade)

Fonte: Wireframe das telas

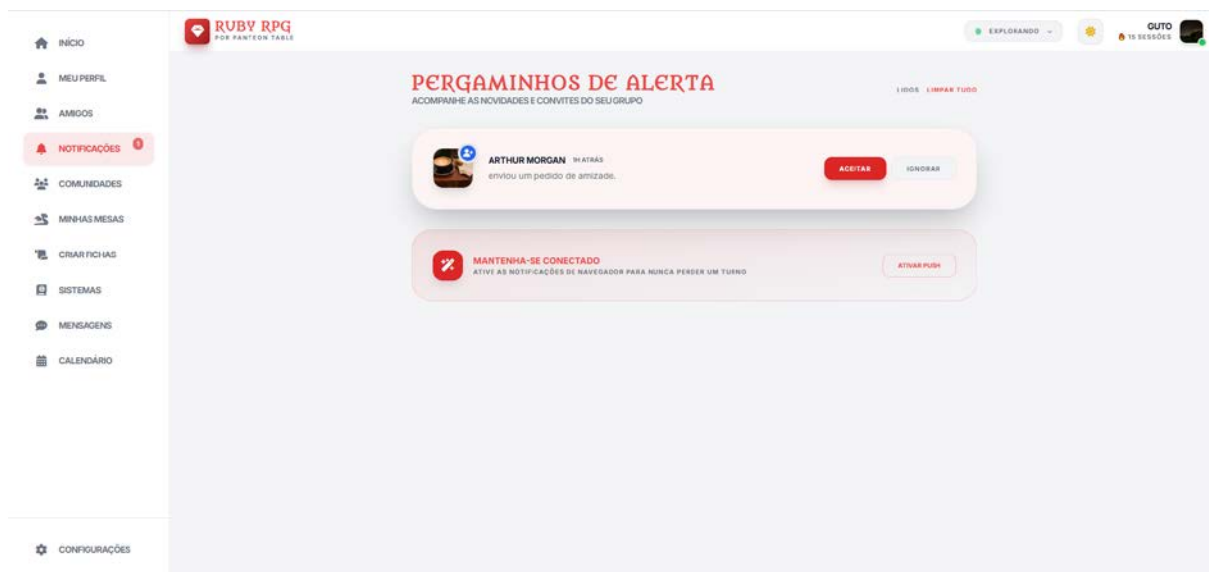


Figura 11 - Notificações

Fonte: Wireframe das telas

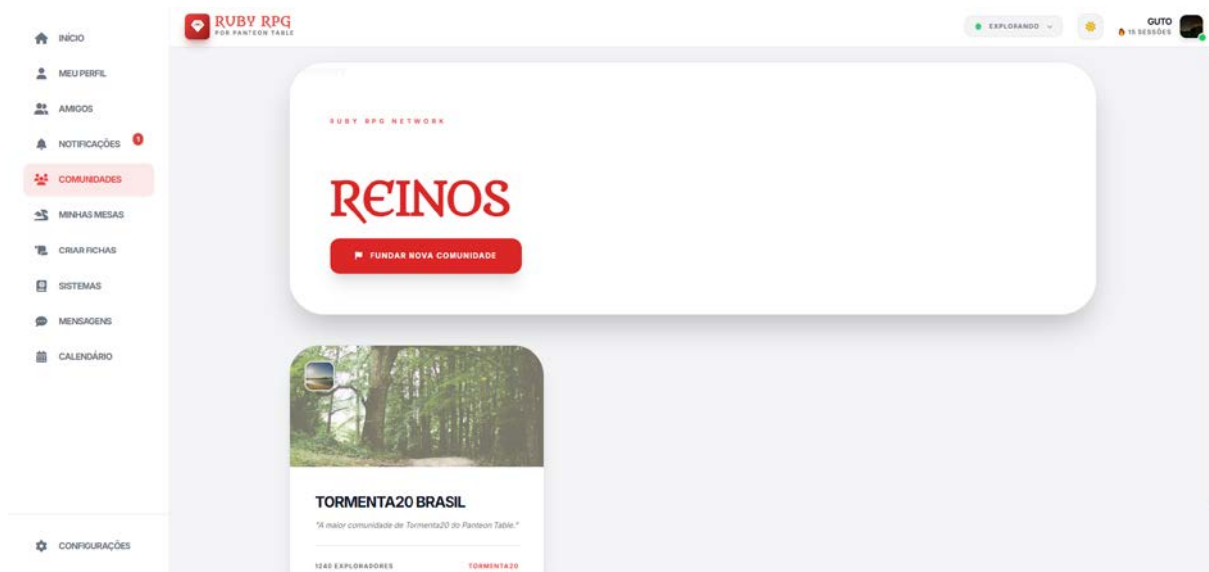


Figura 12 - Tela de comunidades

Fonte: Wireframe das telas



Figura 13 - Tela de uma comunidade (Salão de mesas)

Fonte: Wireframe das telas

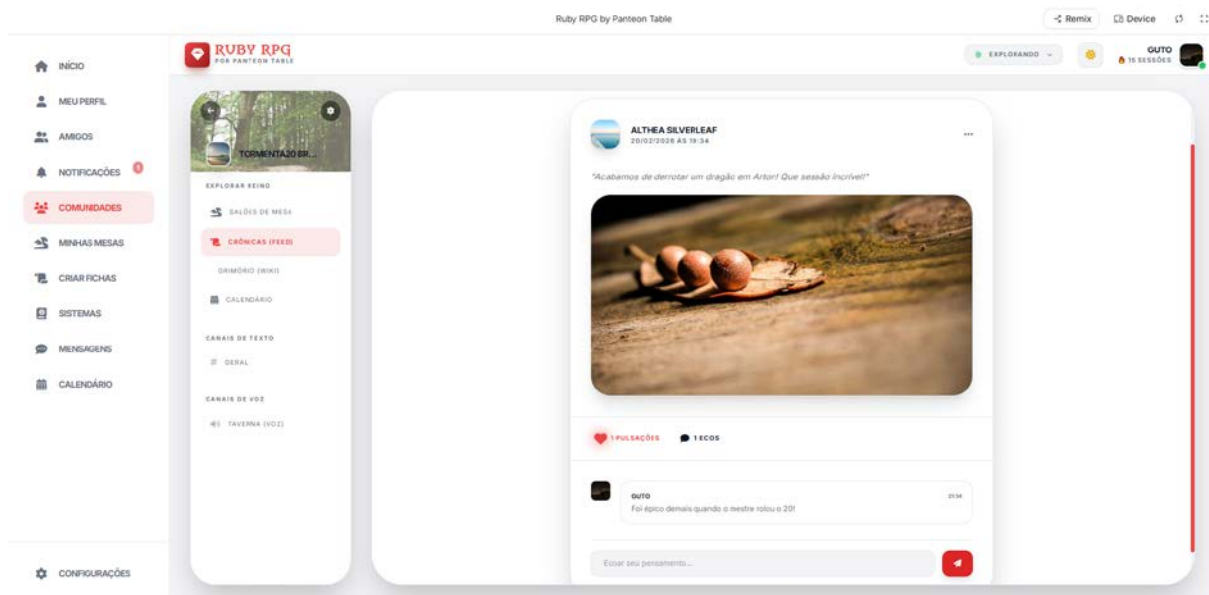


Figura 14 - Tela de crônicas (feed da comunidade)

Fonte: Wireframe das telas

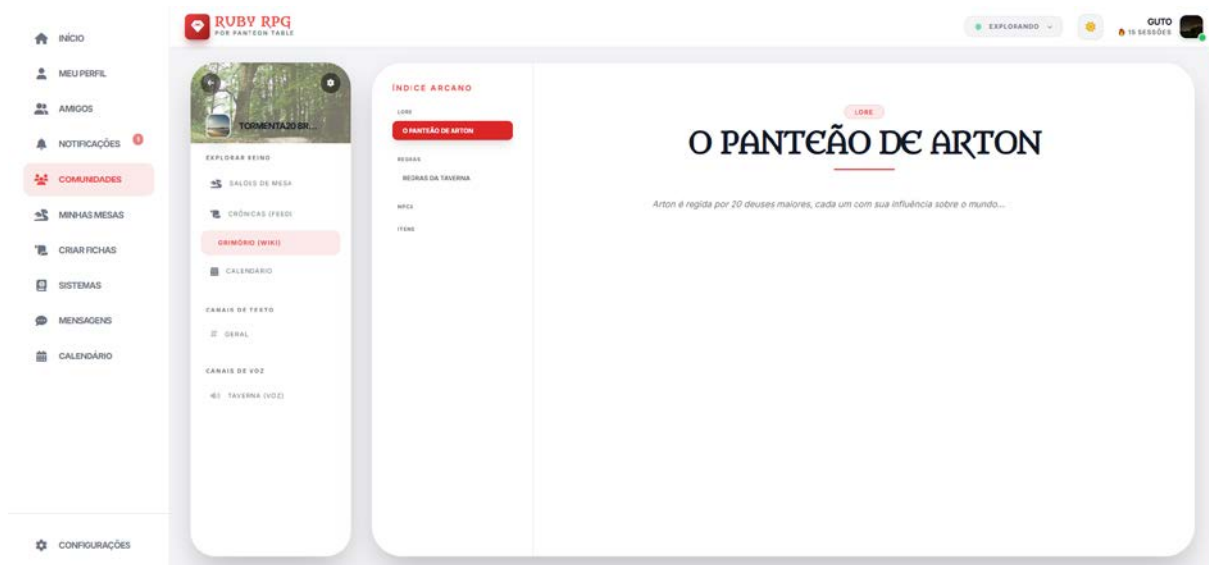


Figura 15 - Tela da Wiki da comunidade (fórum)

Fonte: Wireframe das telas

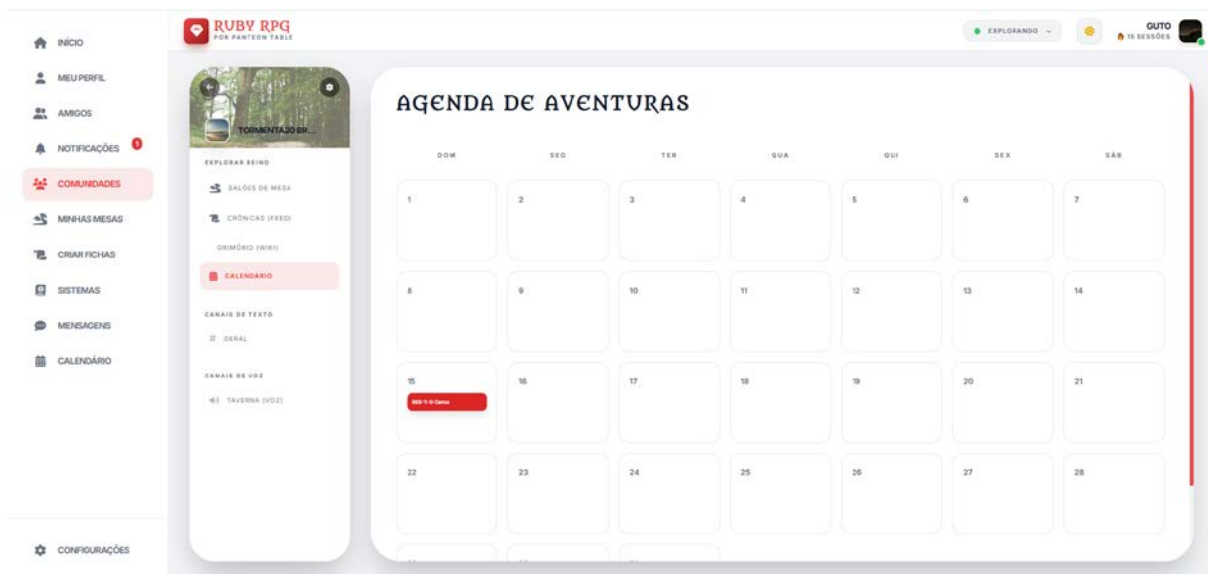


Figura 16 - Tela do calendário da comunidade

Fonte: Wireframe das telas

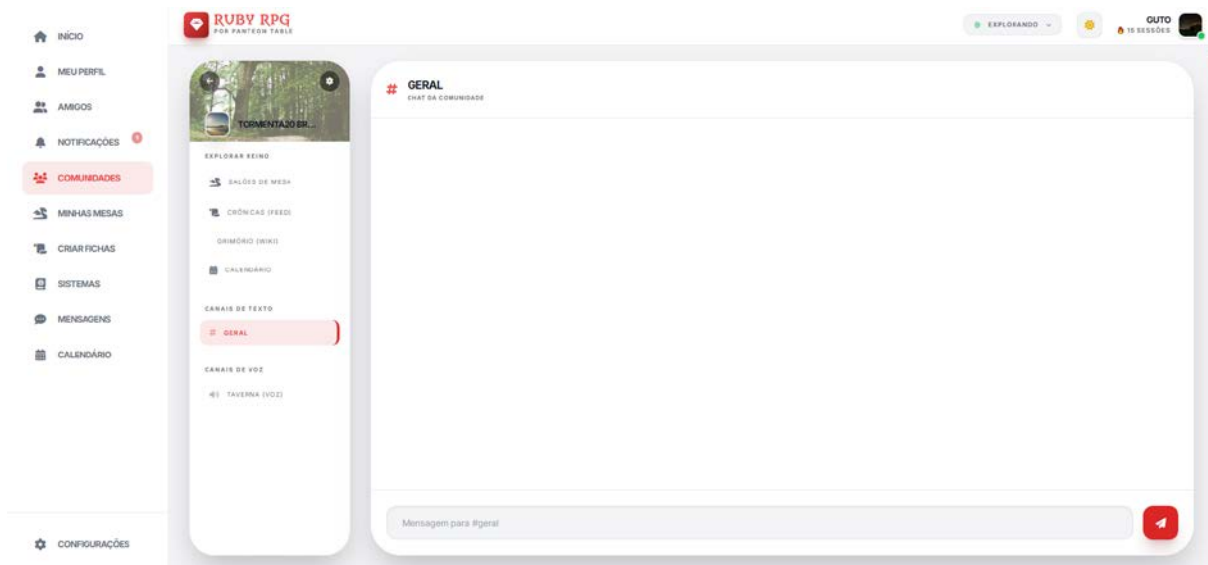


Figura 17 - Tela do canais de texto da comunidade

Fonte: Wireframe das telas

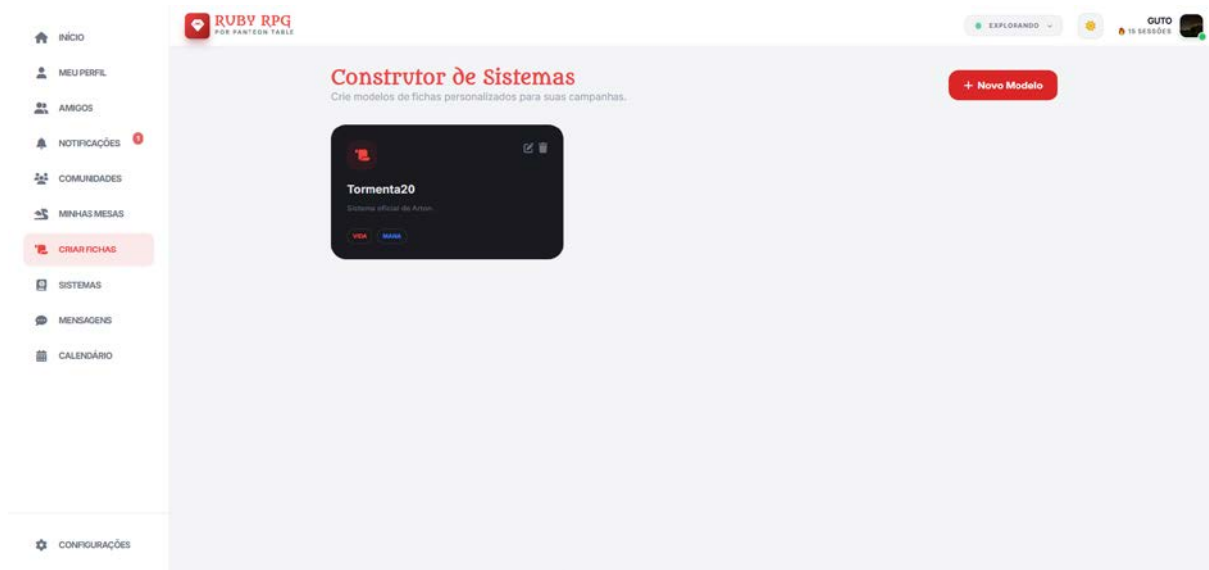


Figura 18 - Tela de criador de fichas para sistemas

Fonte: Wireframe das telas

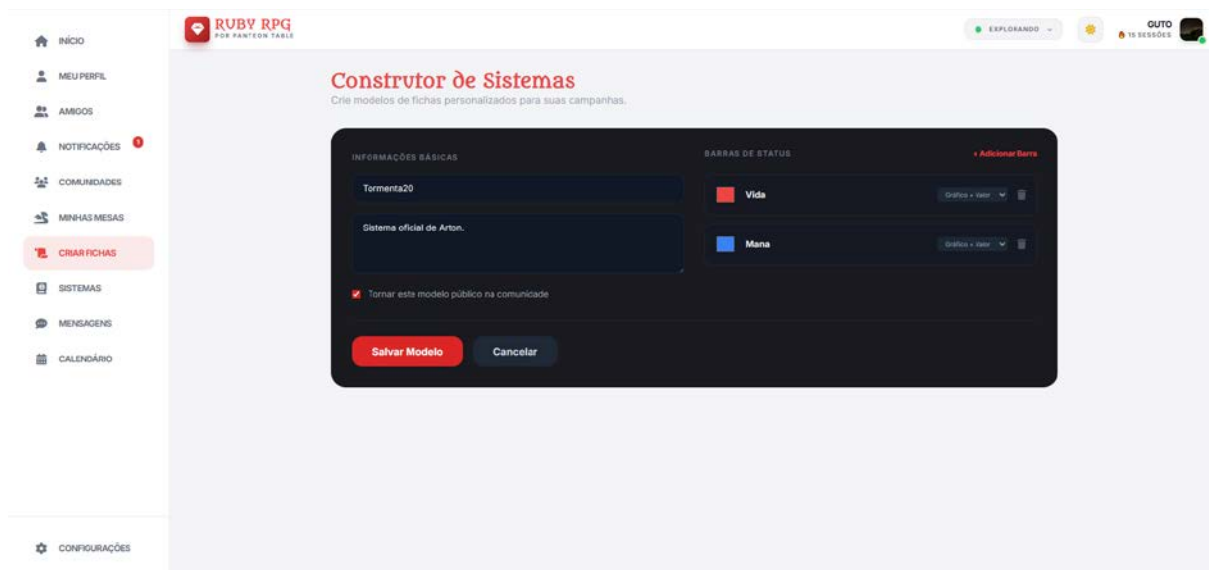


Figura 19 - Tela de editor de ficha para sistemas

Fonte: Wireframe das telas

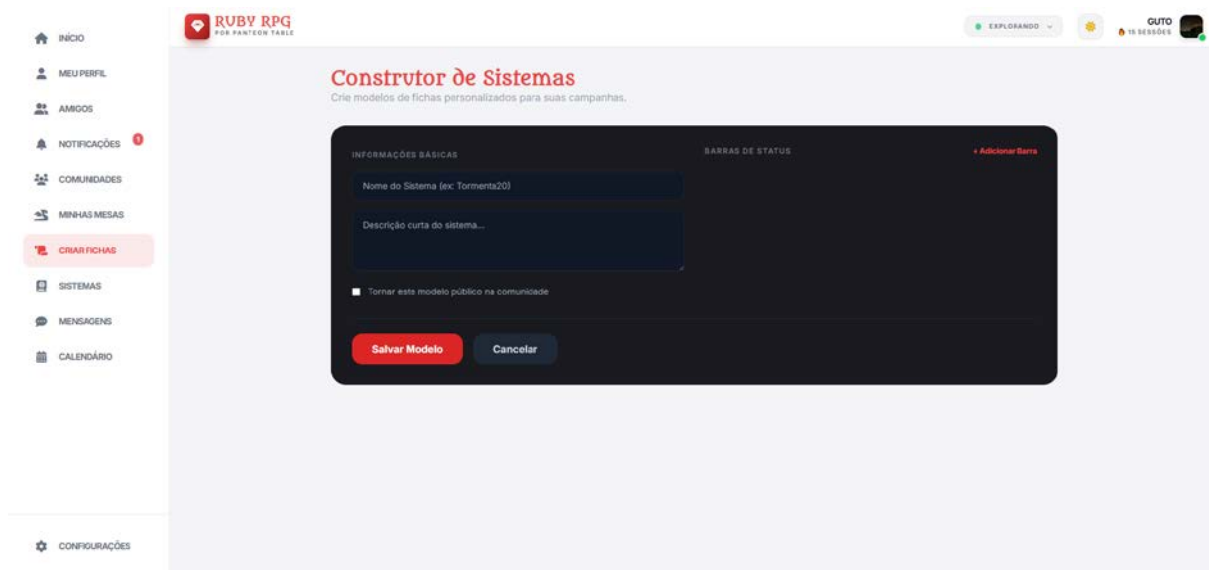


Figura 20 - Tela de criação de fichas para sistemas

Fonte: Wireframe das telas

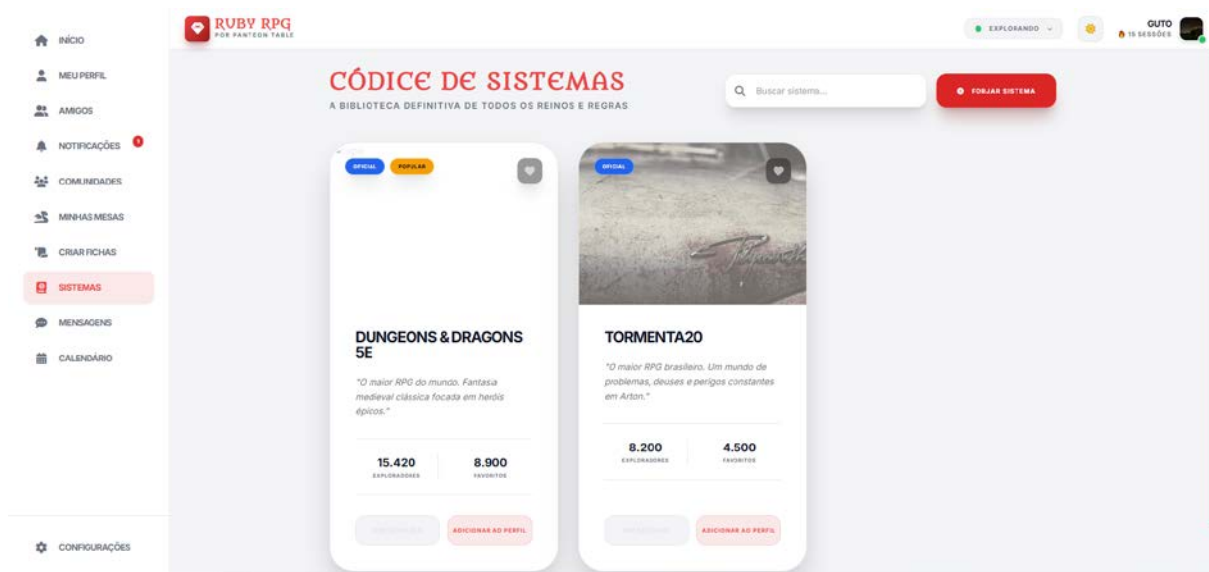


Figura 21 - Tela de sistemas de RPG já existentes

Fonte: Wireframe das telas

Figura 22 - Tela de criação de sistemas de RPG's

Fonte: Wireframe das telas

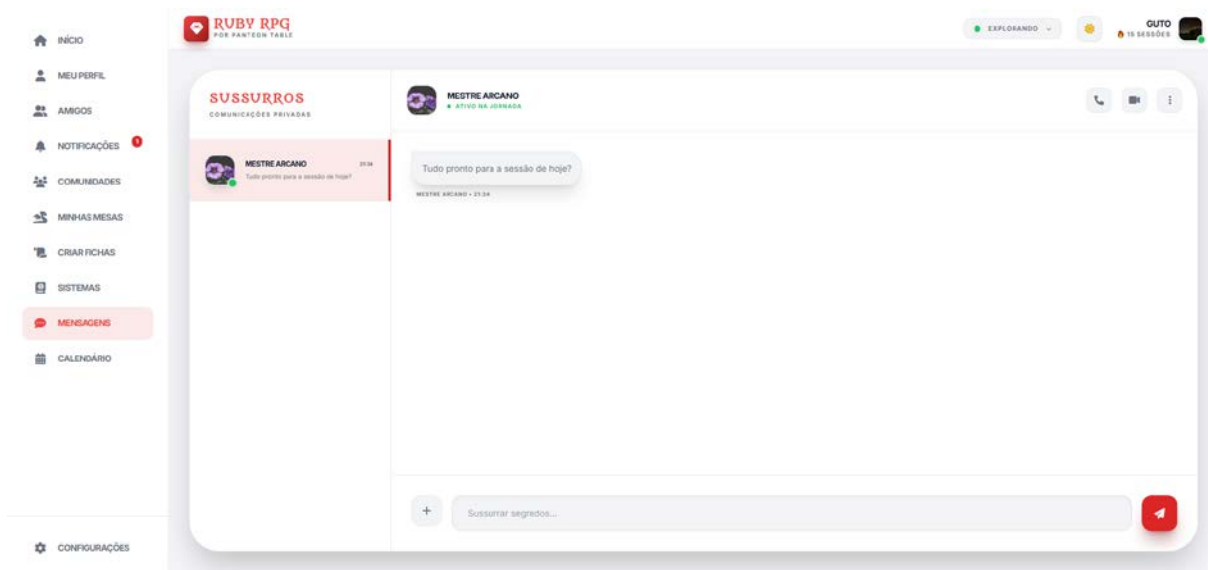


Figura 23 - Tela de mensagens

Fonte: Wireframe das telas

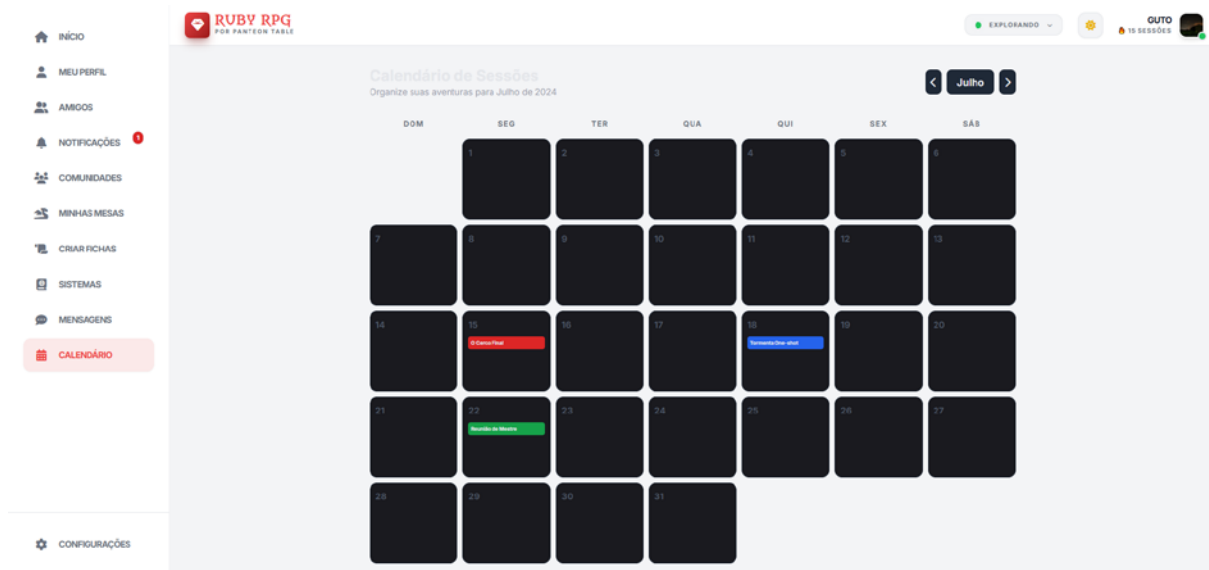


Figura 24 - Tela de calendário pessoal

Fonte: Wireframe das telas

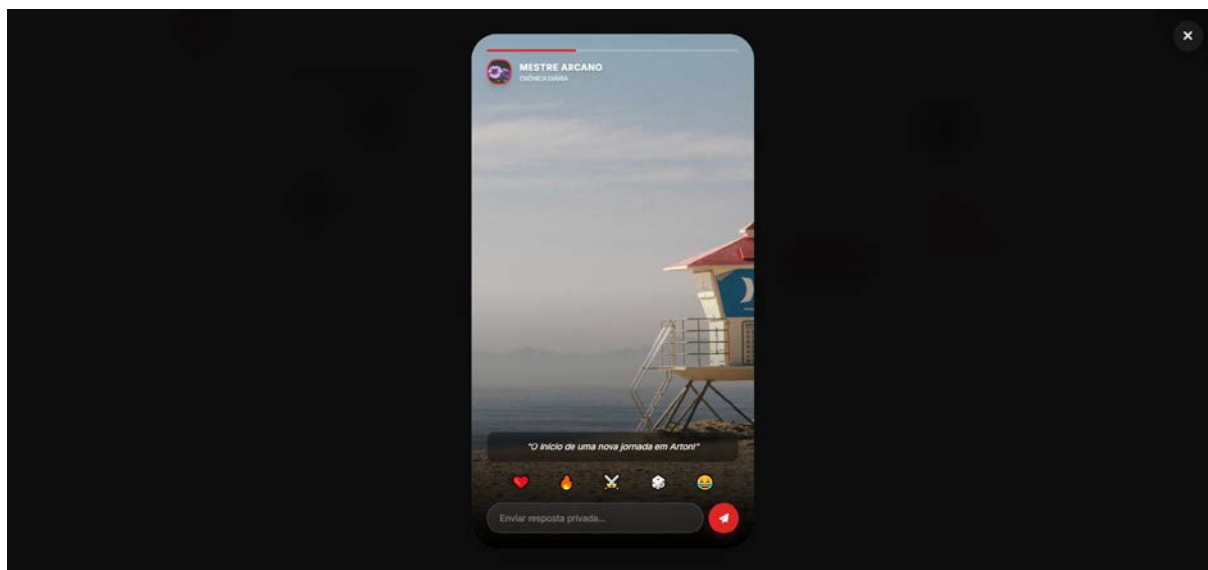


Figura 25 - Tela de storys de perfil

Fonte: Wireframe das telas

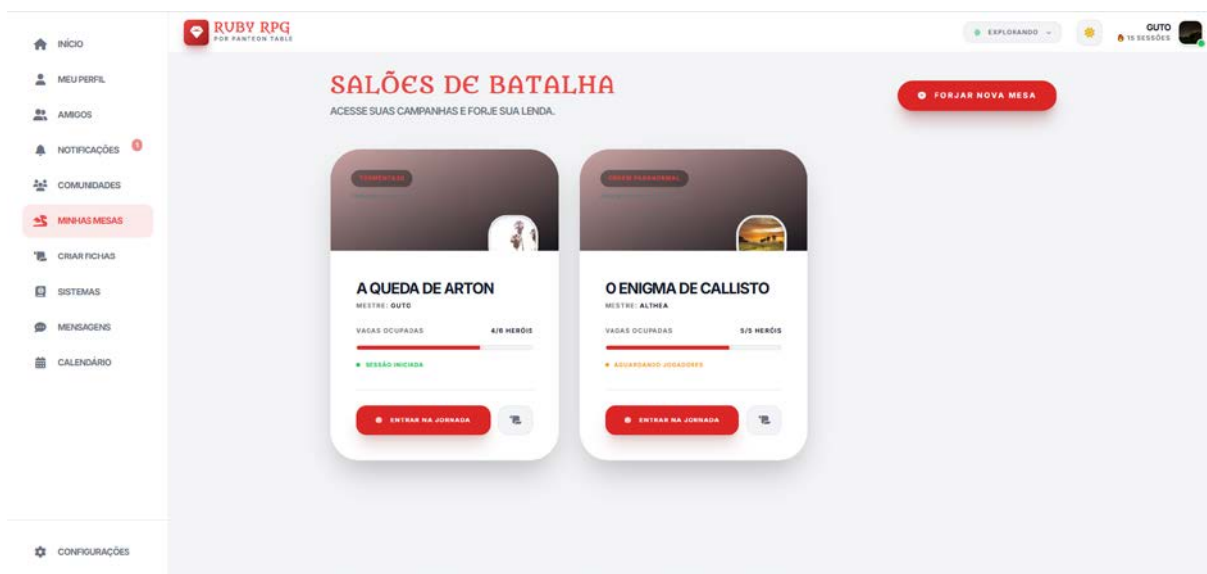


Figura 26 - Tela de mesas

Fonte: Wireframe das telas

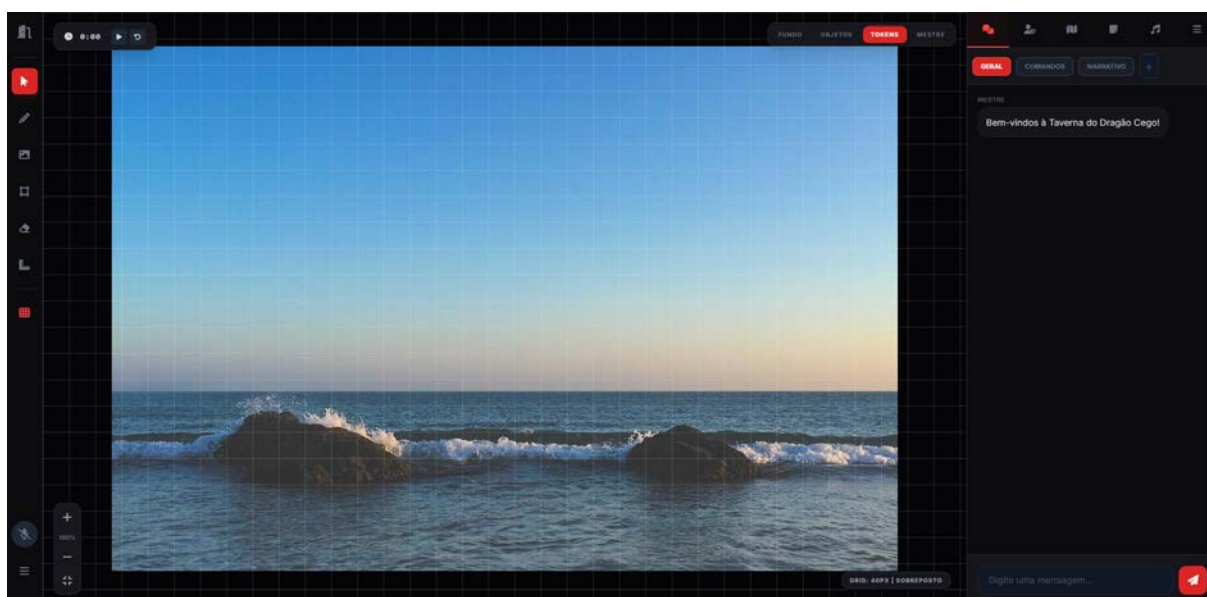


Figura 27 - Tela inicial da mesa (com o chat da mesa aberto)

Fonte: Wireframe das telas

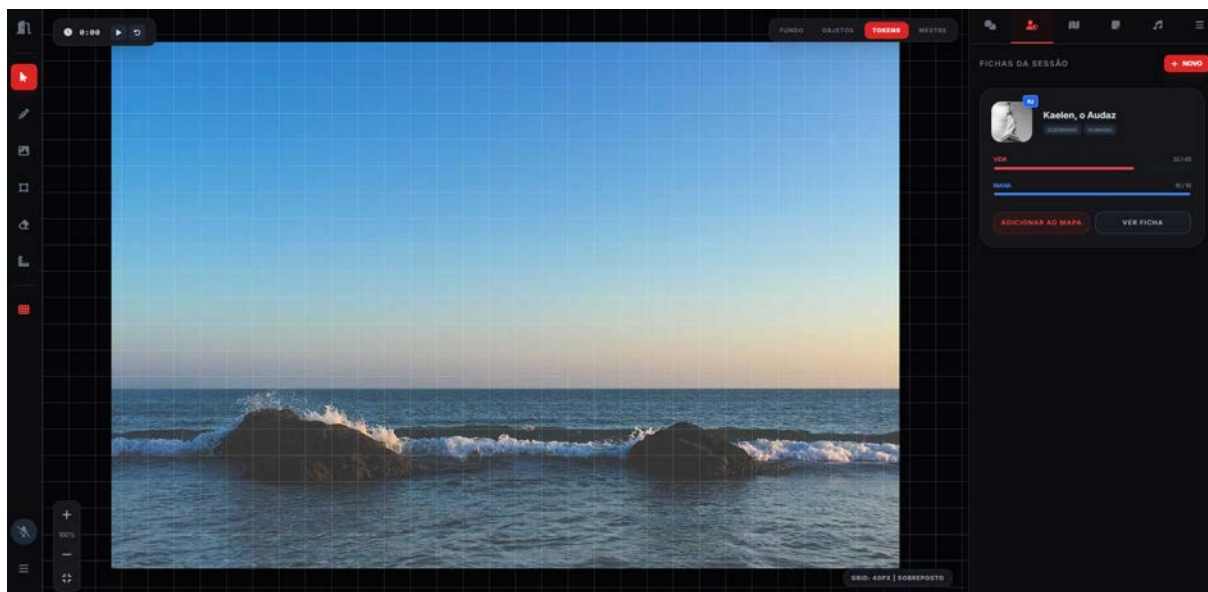


Figura 28 - Tela da mesa (fichas da mesa)

Fonte: Wireframe das telas

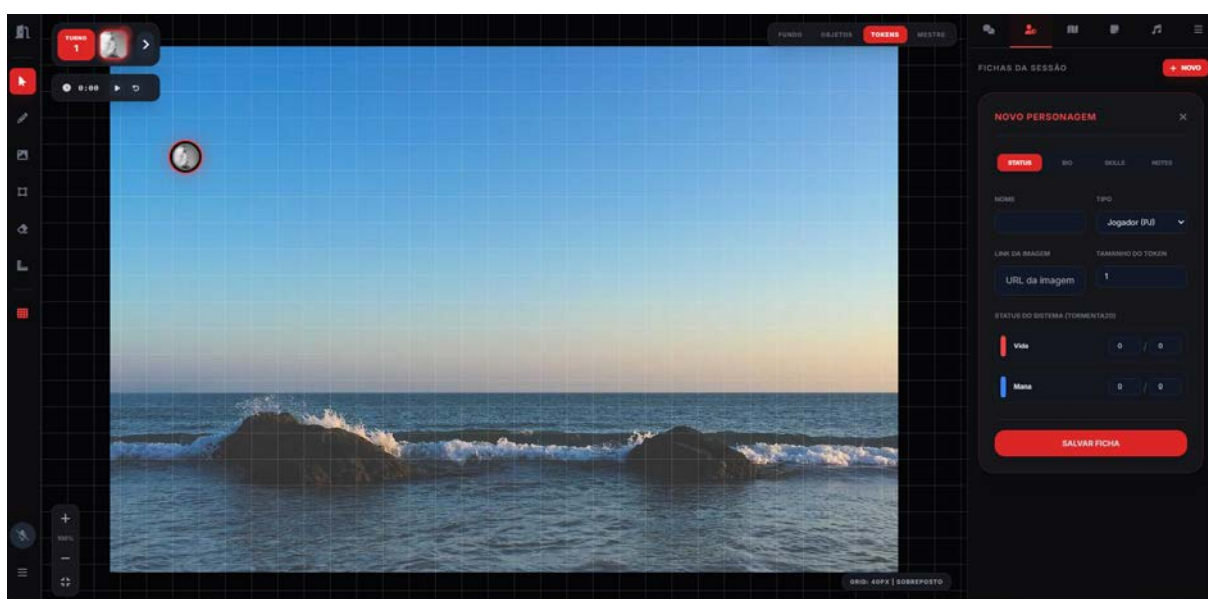


Figura 29 - Tela de mesa (criação de fichas/status)

Fonte: Wireframe das telas

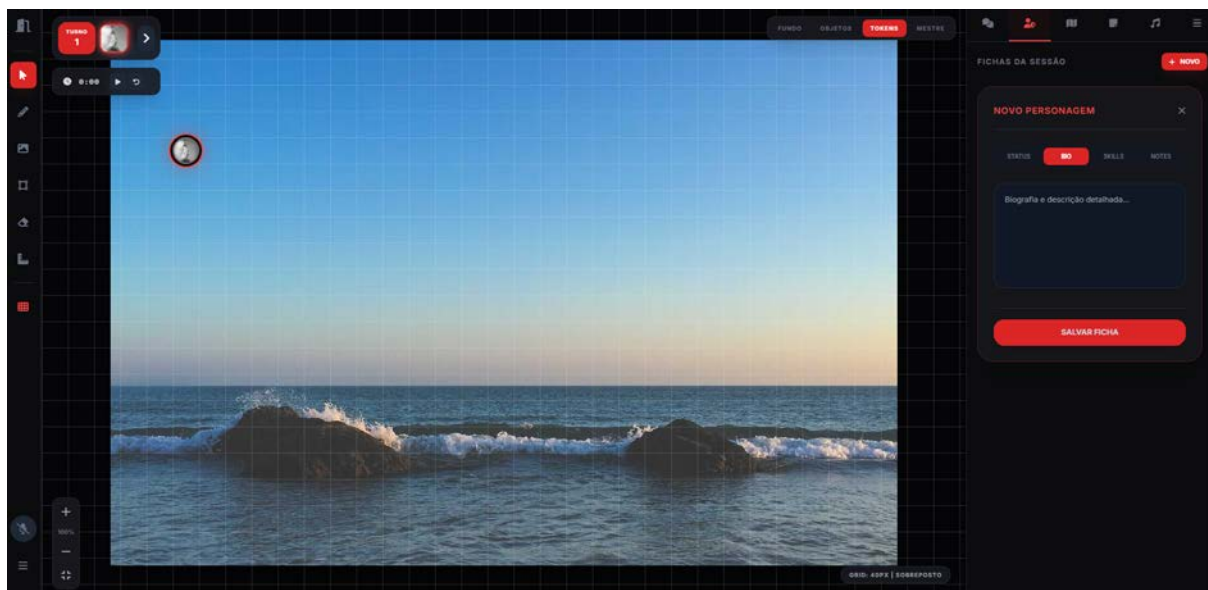


Figura 30 - Tela de mesa (criação de fichas/biografia)

Fonte: Wireframe das telas

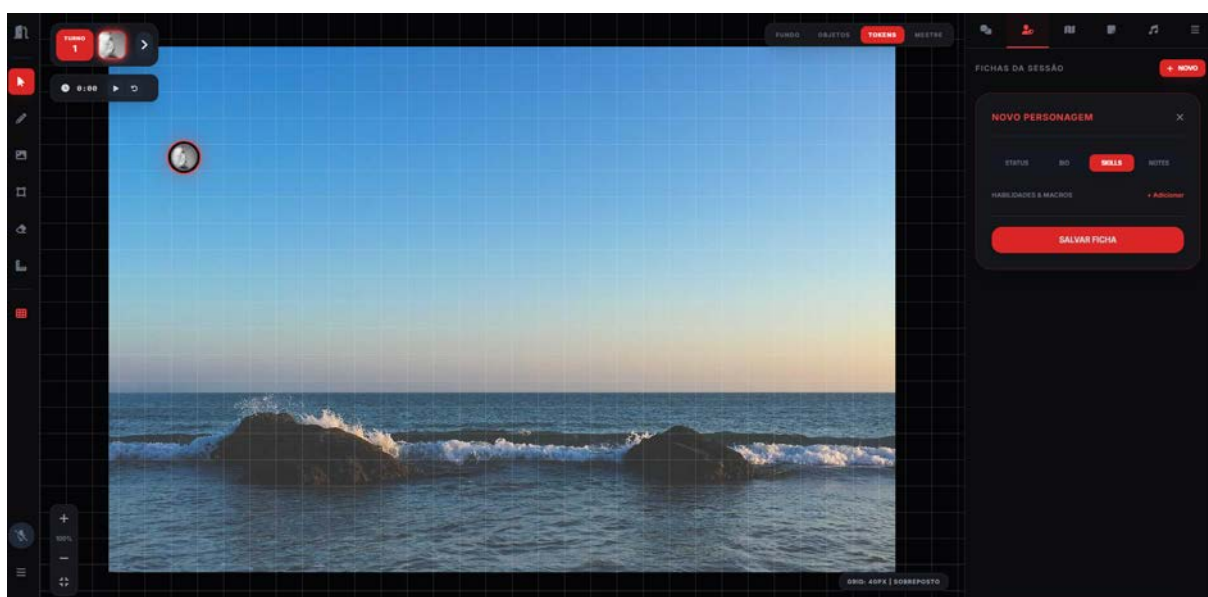


Figura 31 - Tela de mesa (criação de fichas/skills)

Fonte: Wireframe das telas

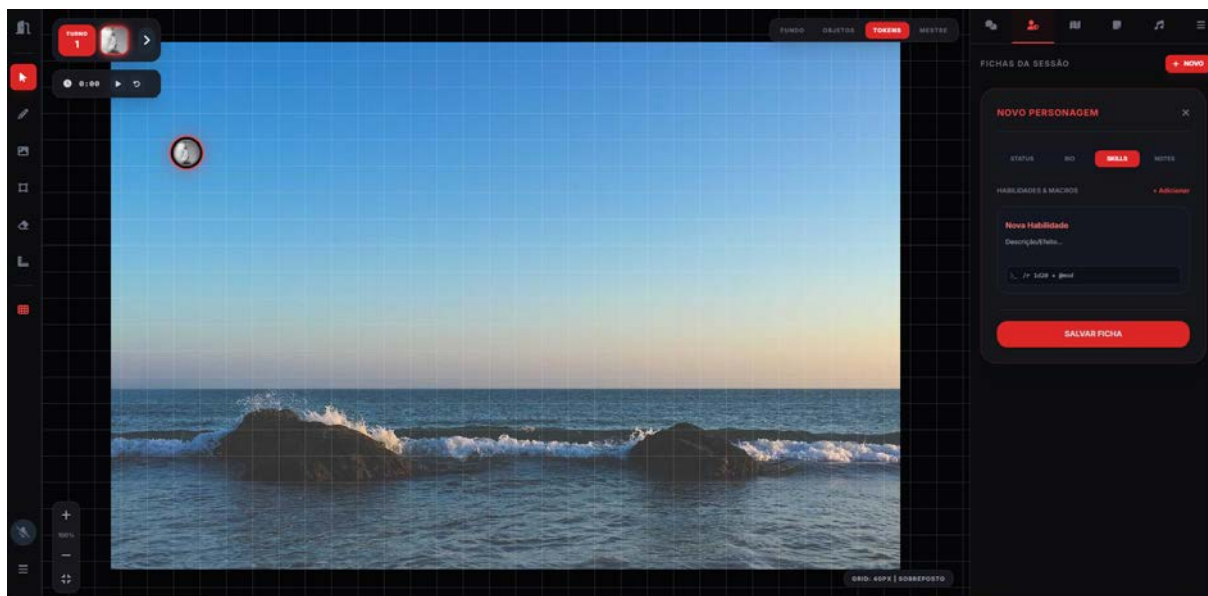


Figura 32 - Tela de mesa (criação de fichas/skills/criação)

Fonte: Wireframe das telas

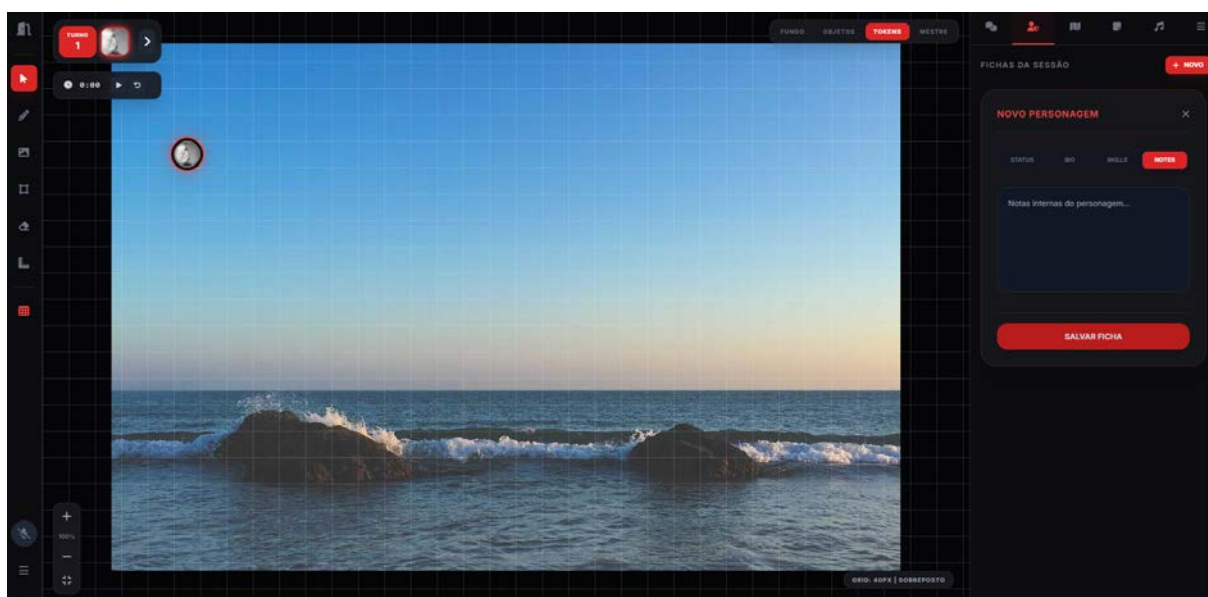


Figura 33 - Tela de mesa (criação de fichas/anotações)

Fonte: Wireframe das telas

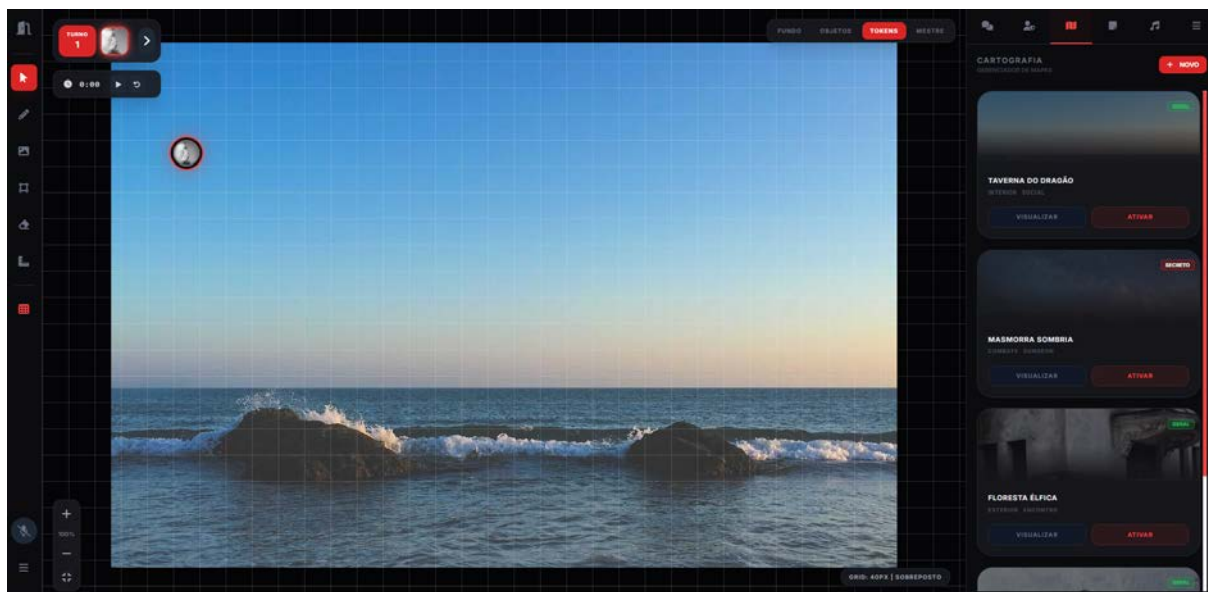


Figura 34 - Tela de mesa (mapas)

Fonte: Wireframe das telas

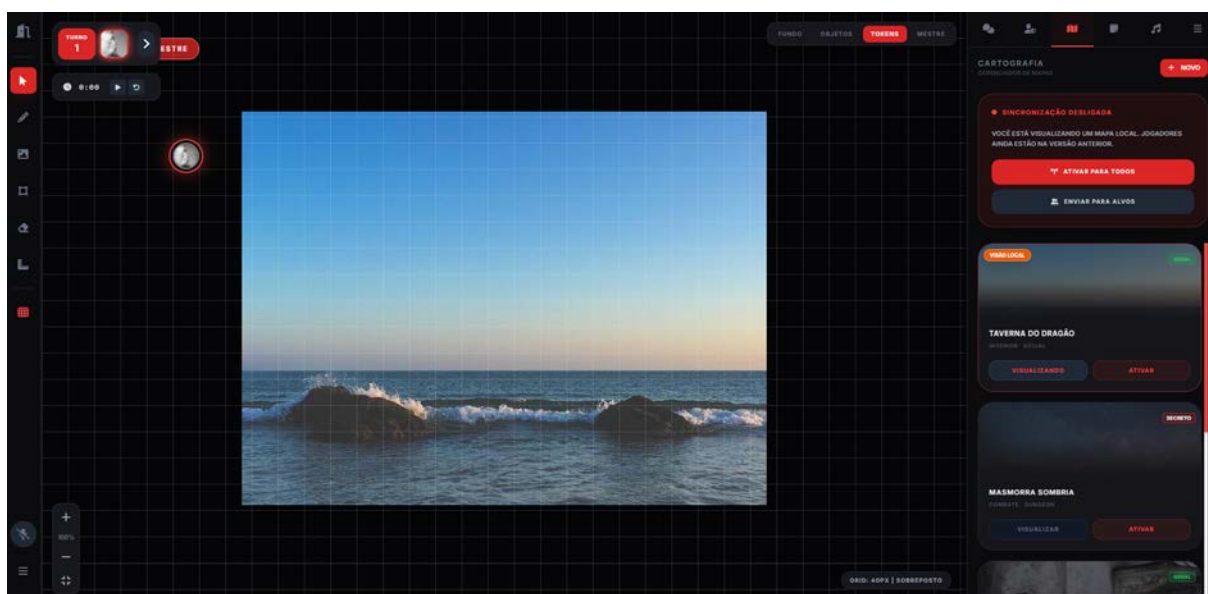


Figura 35 - Tela de mesa (mapas/visualização de mapas)

Fonte: Wireframe das telas

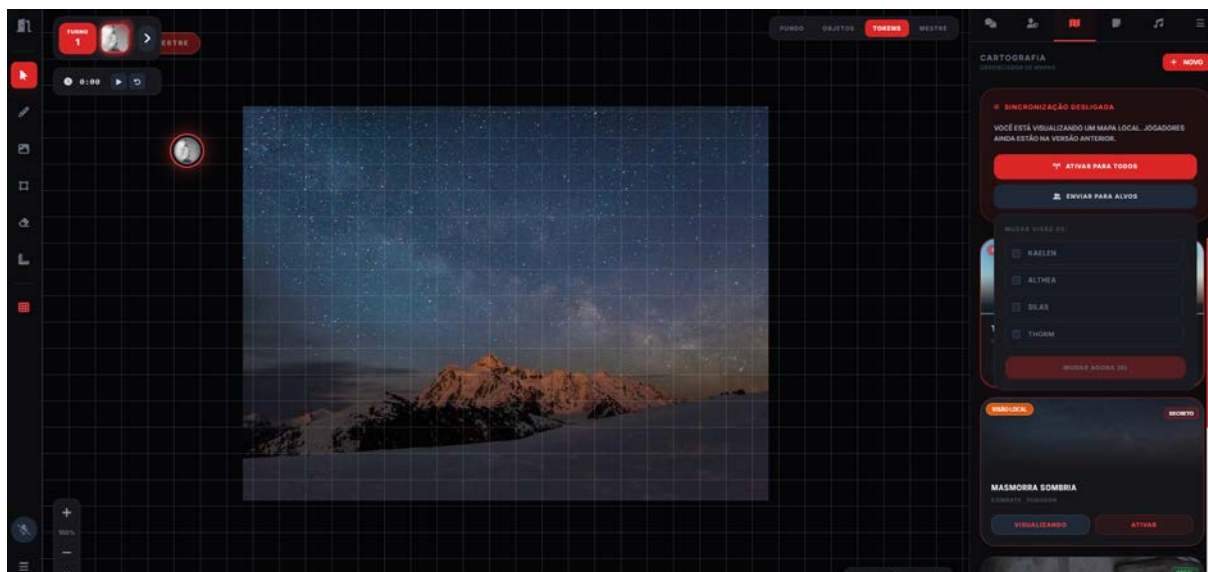


Figura 36 - Tela de mesa (mapas/visualização de mapas/privacidade)

Fonte: Wireframe das telas

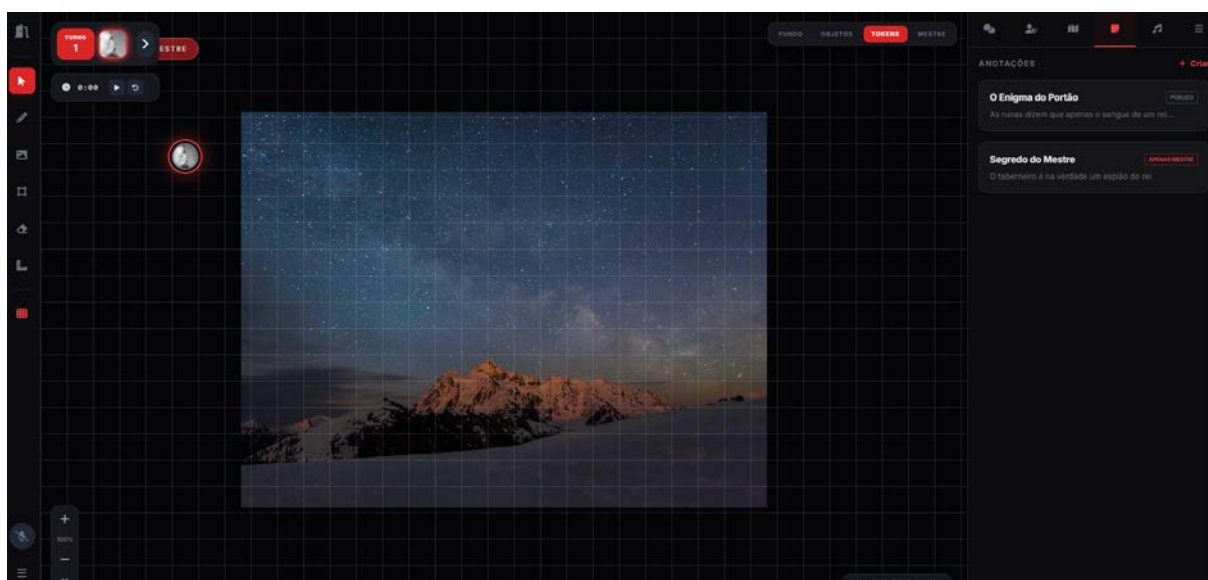


Figura 37 - Tela de mesa (anotações pessoais)

Fonte: Wireframe das telas

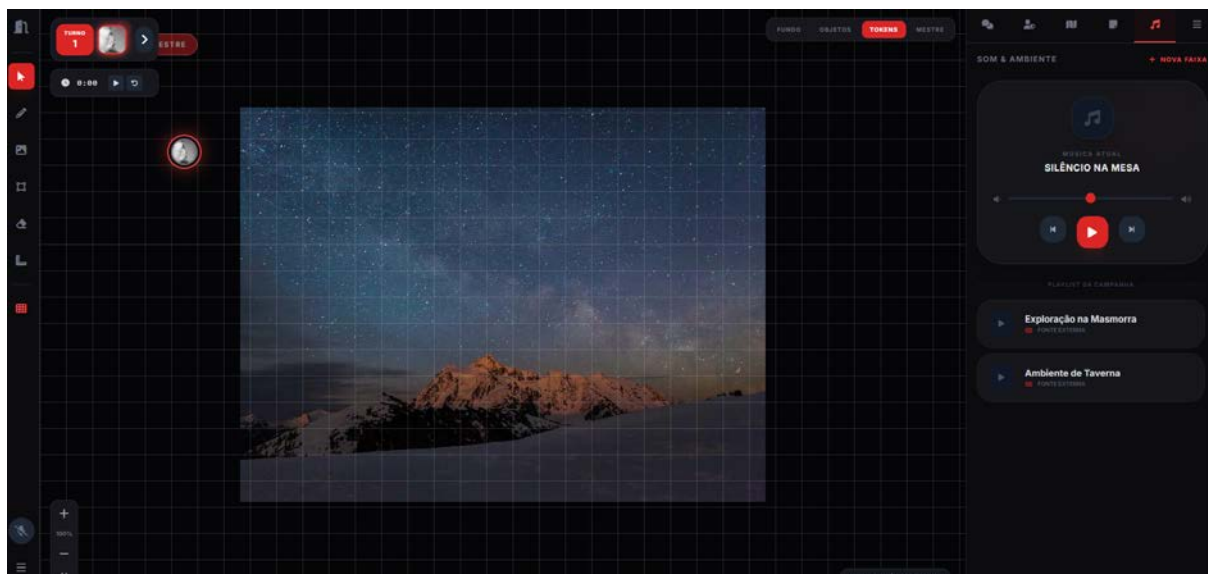


Figura 38 - Tela de mesa (tria sonora)

Fonte: Wireframe das telas

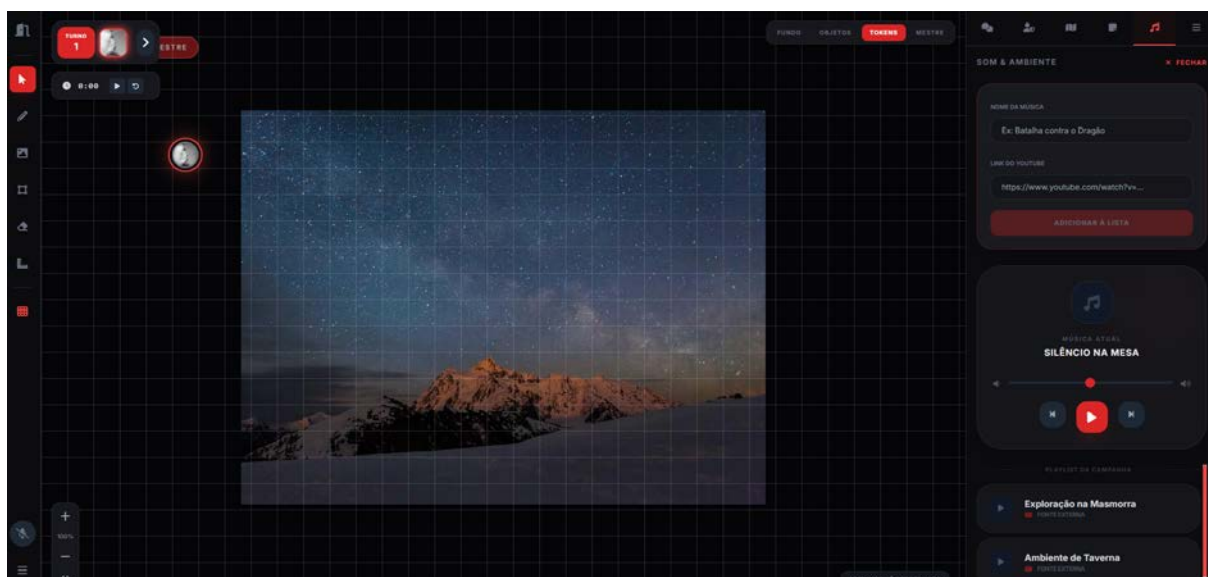


Figura 39 - Tela de mesa (tria sonora/criar faixa)

Fonte: Wireframe das telas

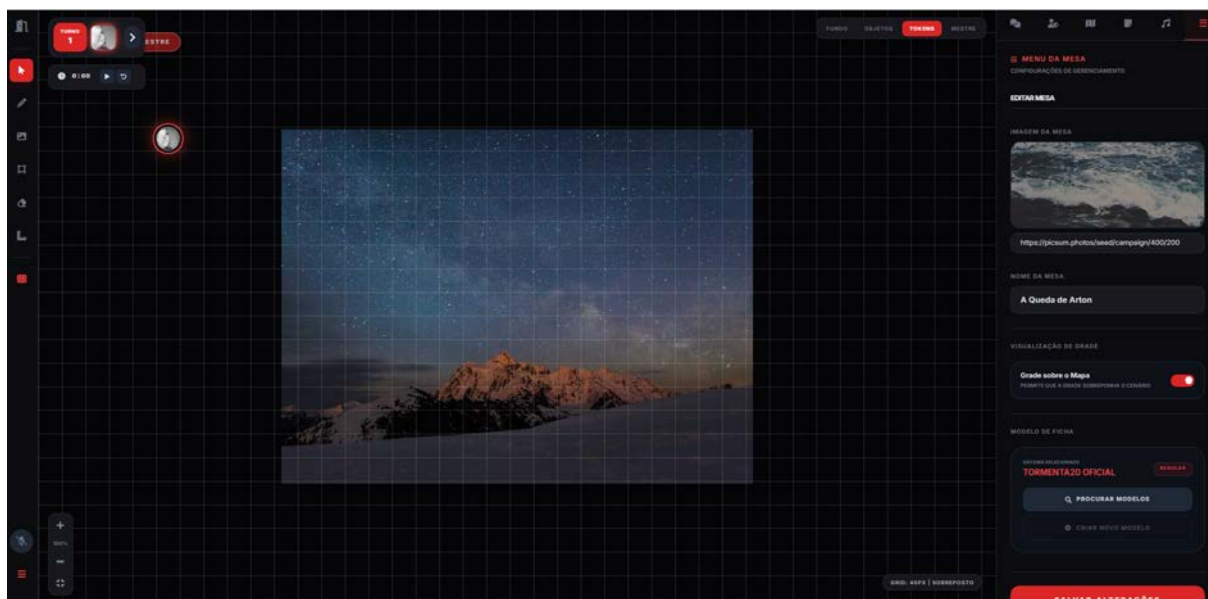


Figura 40 - Tela de mesa (Configuração da mesa)

Fonte: Wireframe das telas

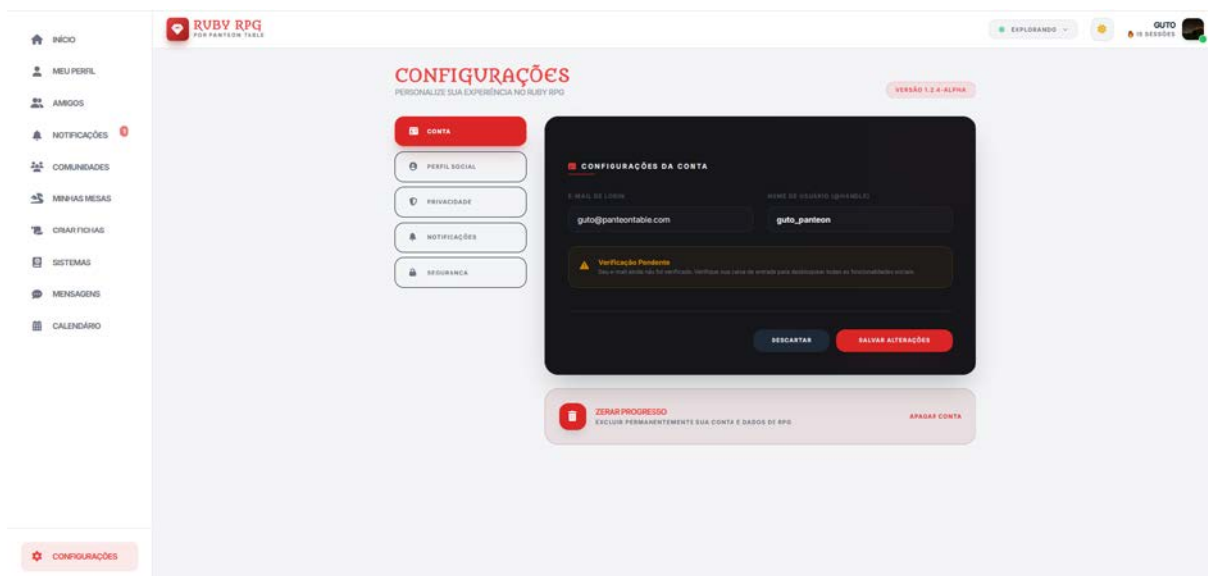


Figura 41 - Tela de configuração de conta

Fonte: Wireframe das telas

4.6 Prints das Telas

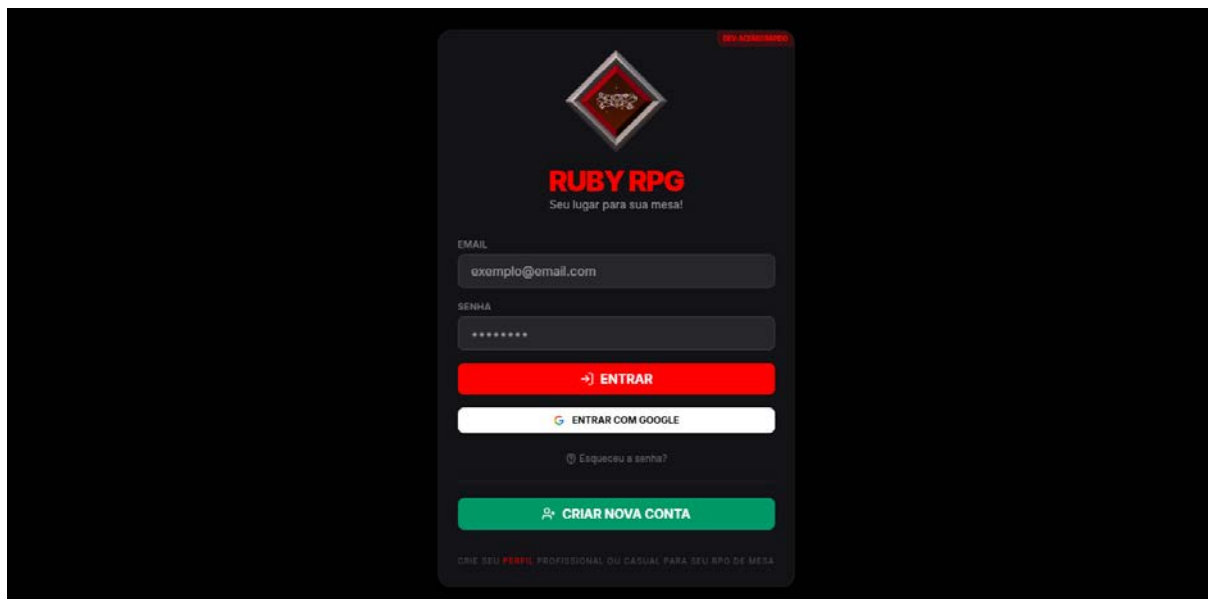


Figura 42 - Login

Fonte: Criado pela equipe

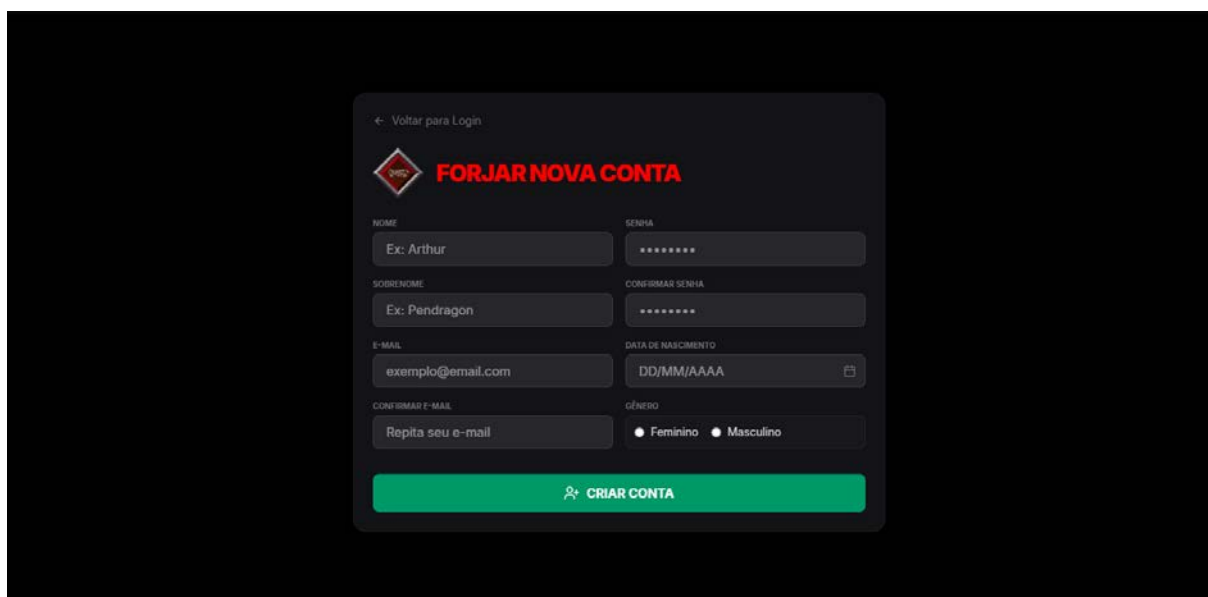


Figura 43 - Cadastro

Fonte: Criado pela equipe

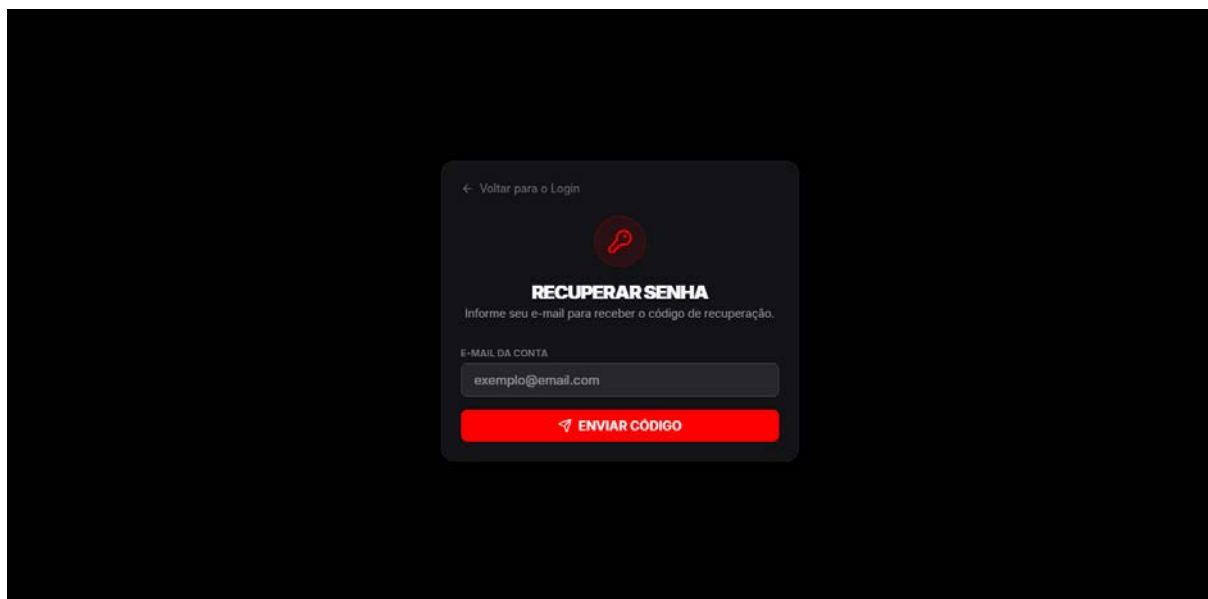


Figura 44 – Recuperação de senha

Fonte: Criado pela equipe

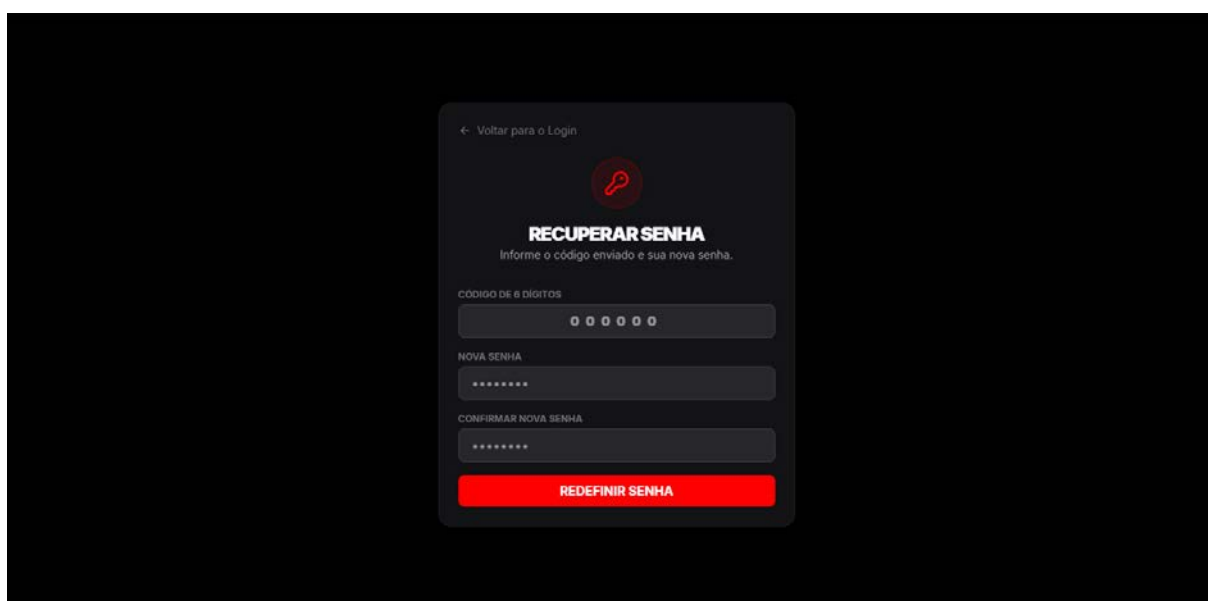


Figura 45 – Recuperação de conta (redefinição de senha)

Fonte: Criado pela equipe

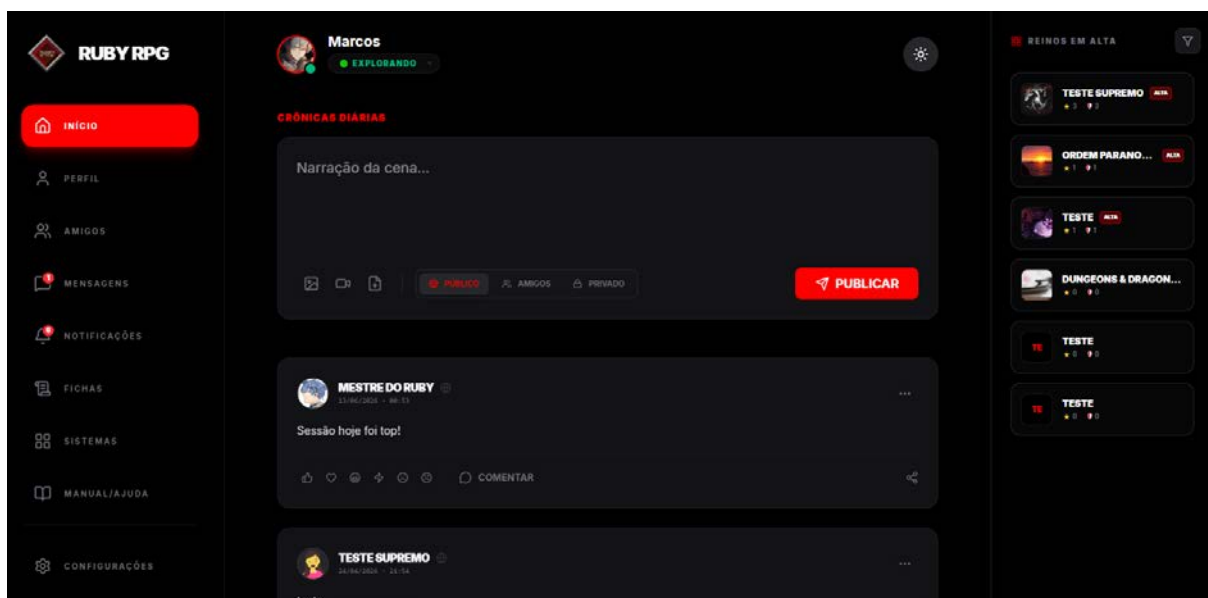


Figura 46 – Tela inicial

Fonte: Print das Telas

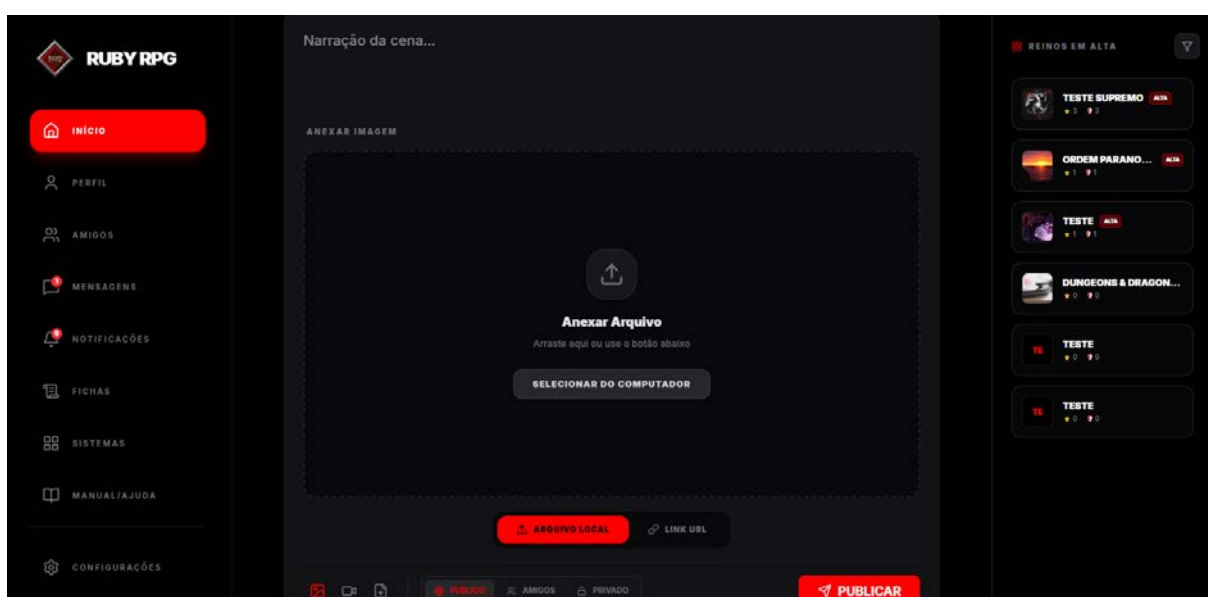


Figura 47 – Tela inicial (criação de post/upload)

Fonte: Print das Telas

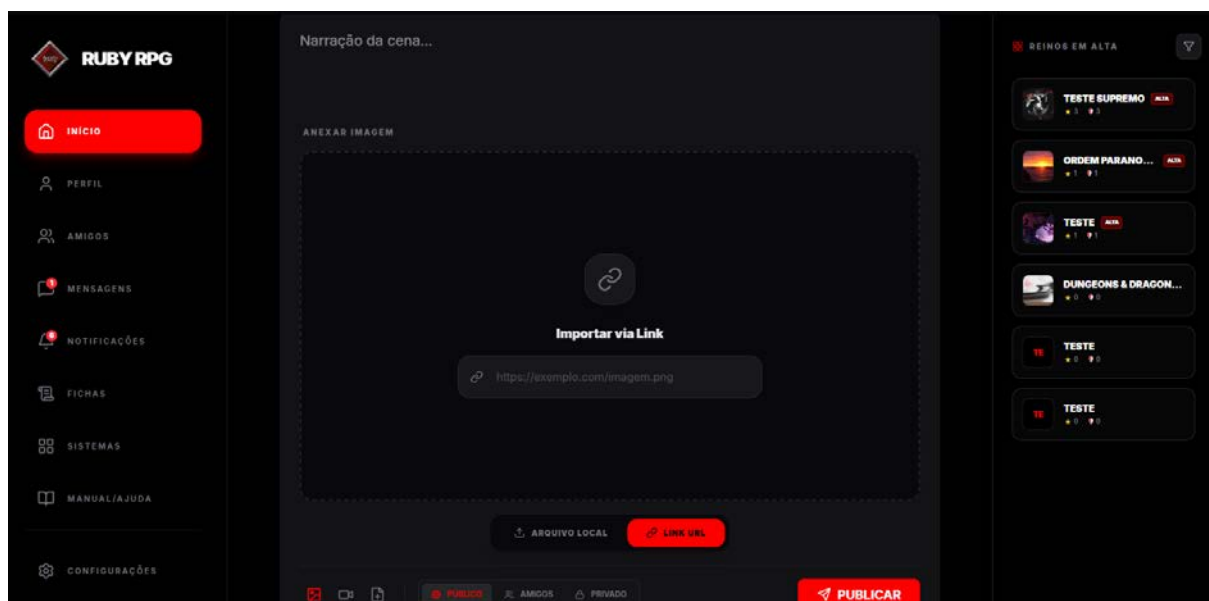


Figura 48 – Tela inicial (criação de post/link)

Fonte: Criado pela equipe

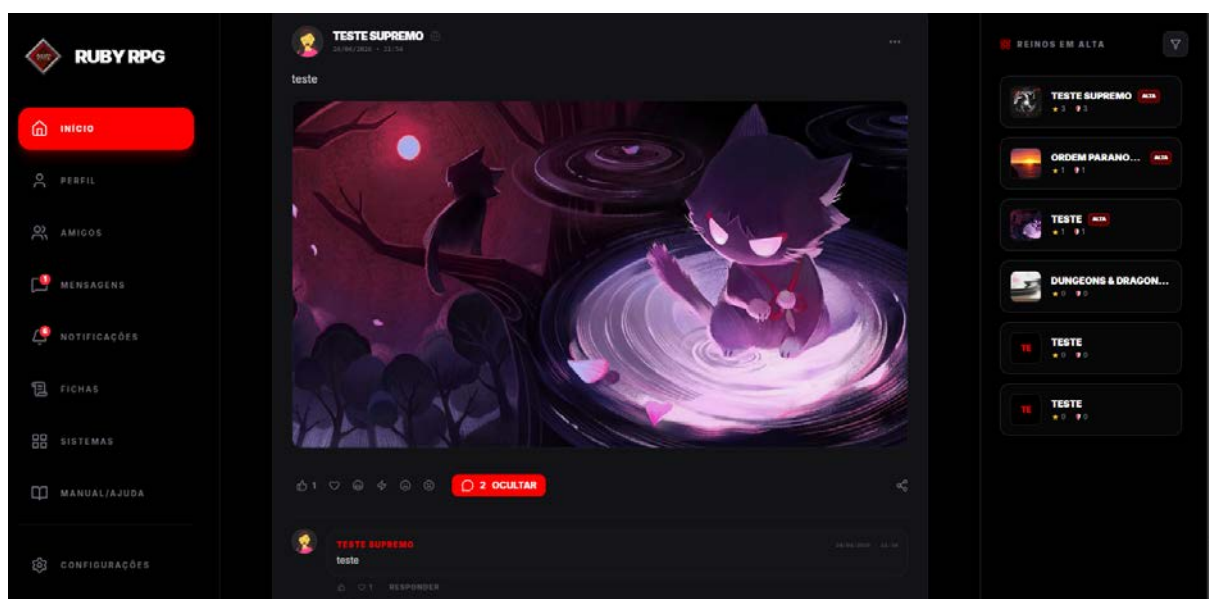


Figura 49 – Post e comentários

Fonte: Criado pela equipe

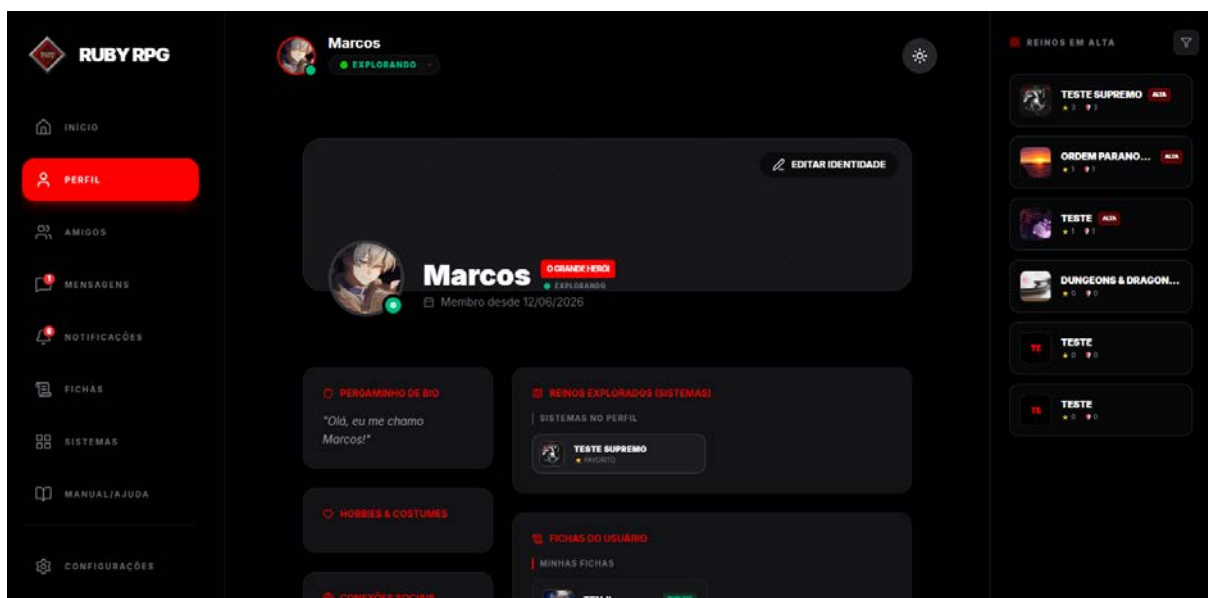


Figura 50 – Perfil de usuário

Fonte: Criado pela equipe

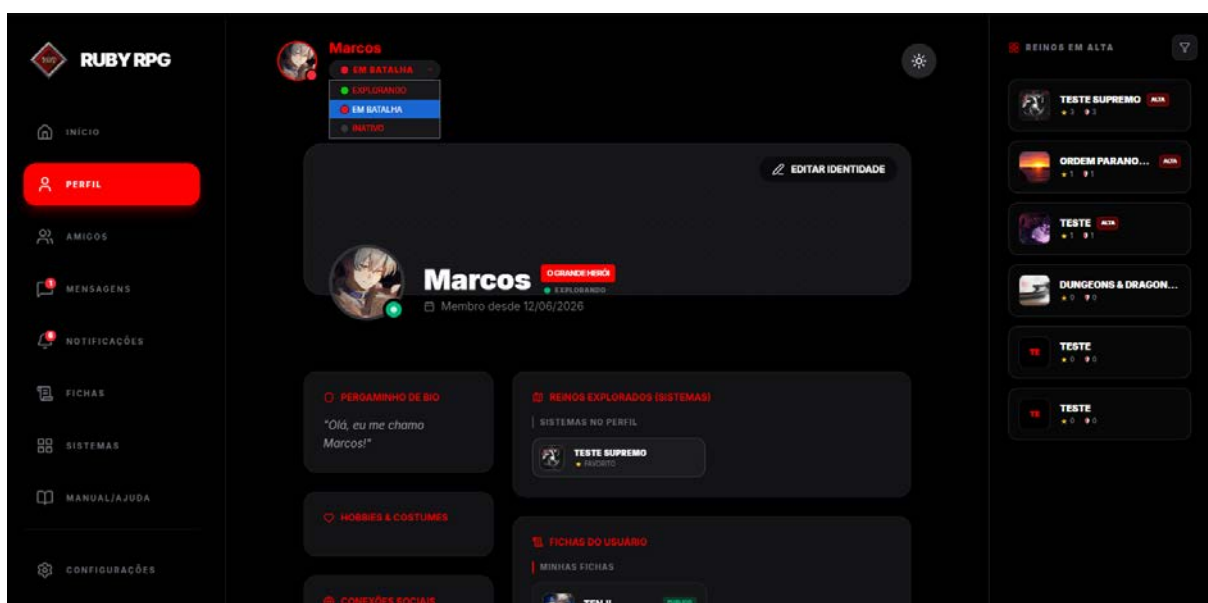


Figura 51 – Status de perfil

Fonte: Criado pela equipe

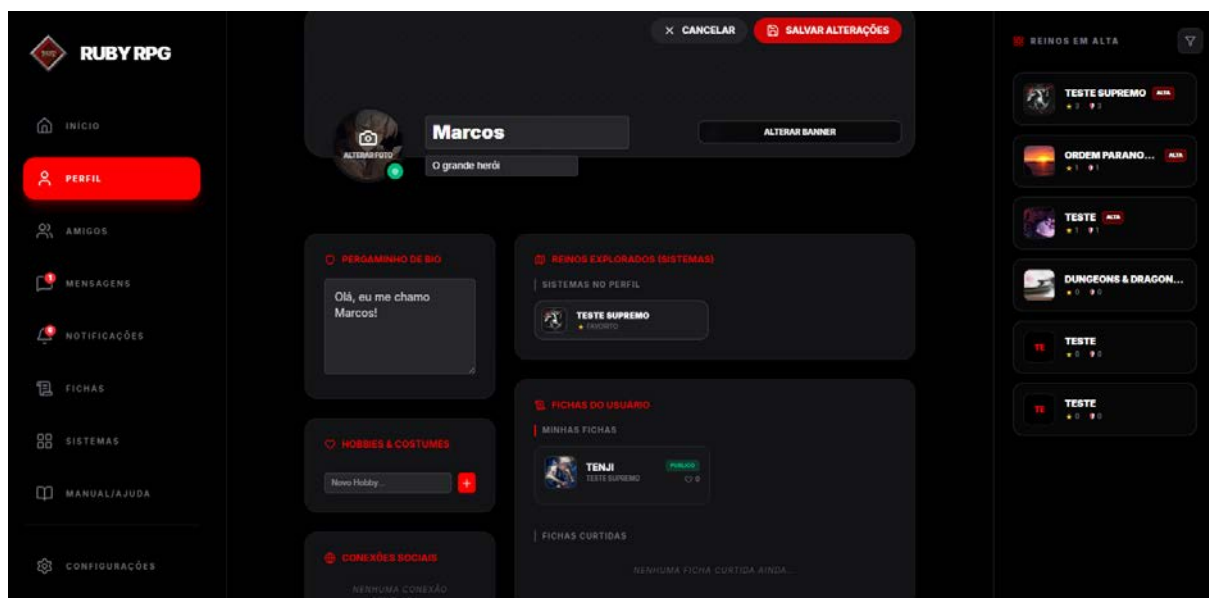


Figura 52 – Edição de perfil

Fonte: Criado pela equipe

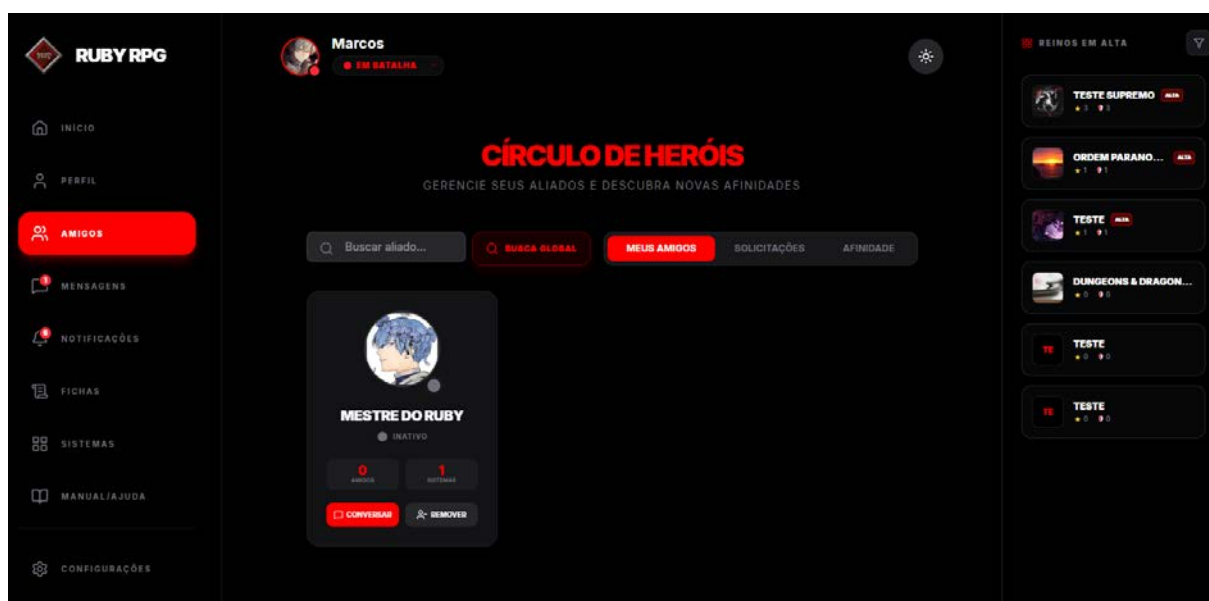


Figura 53 – Amigos do usuário

Fonte: Criado pela equipe

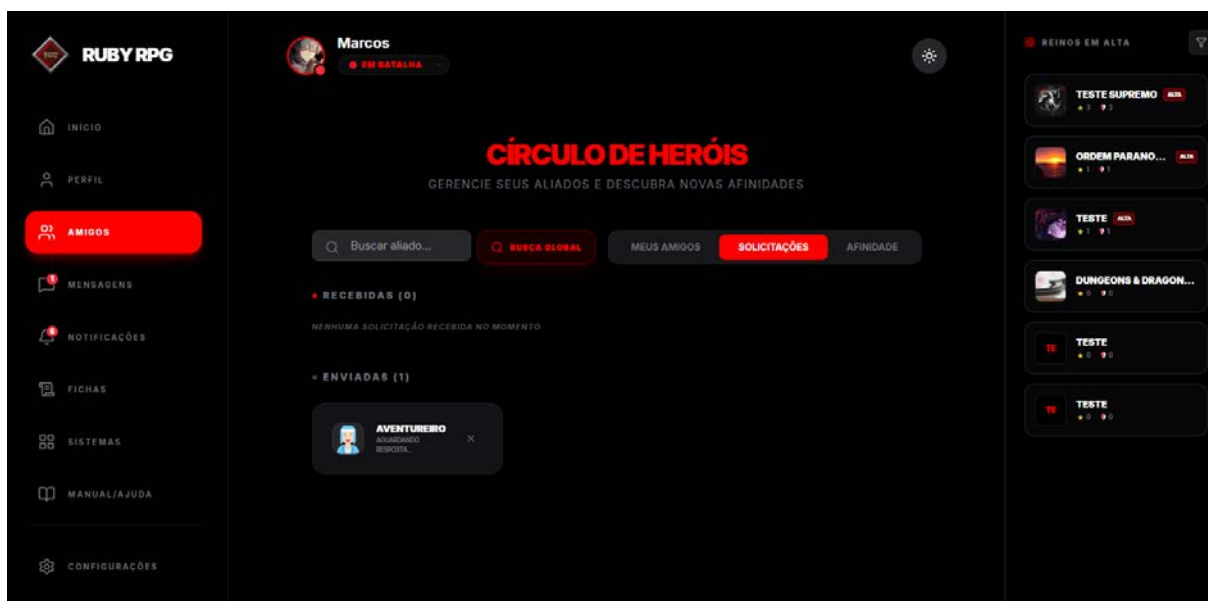


Figura 54 – Solicitações de amizade

Fonte: Criado pela equipe

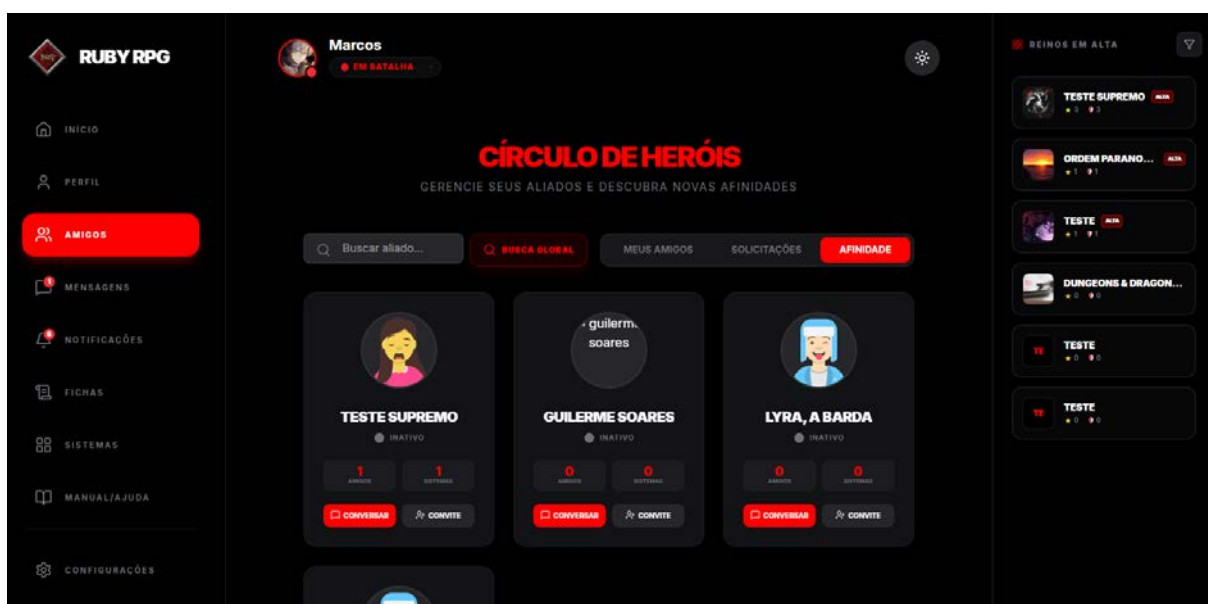


Figura 55 – Usuários

Fonte: Criado pela equipe

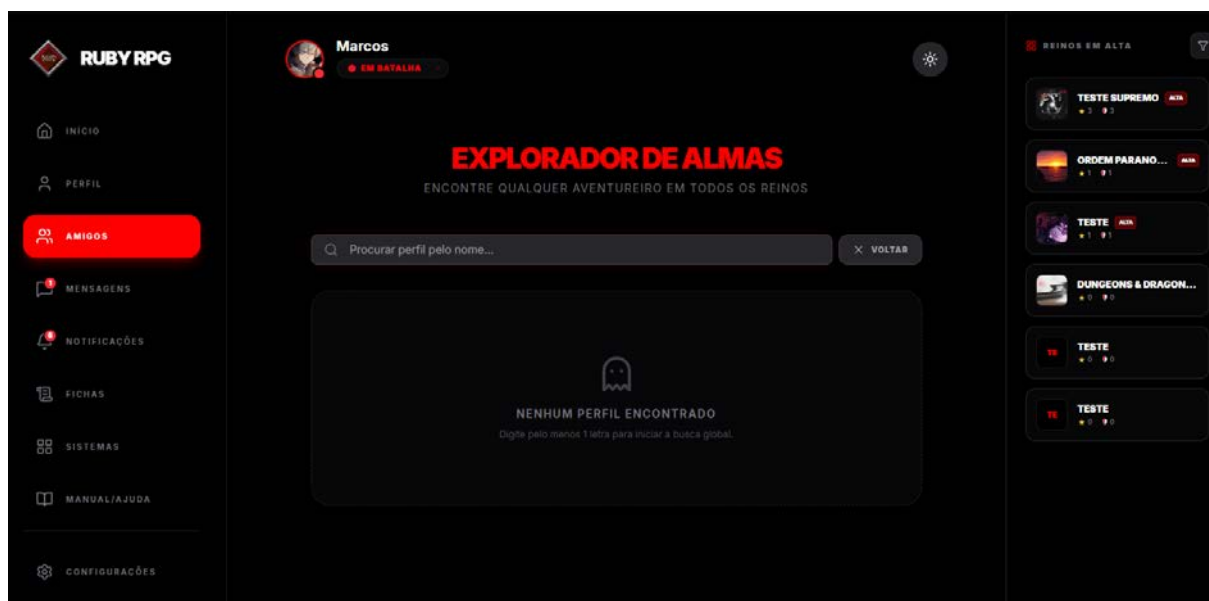


Figura 56 – Pesquisa de usuários

Fonte: Criado pela equipe

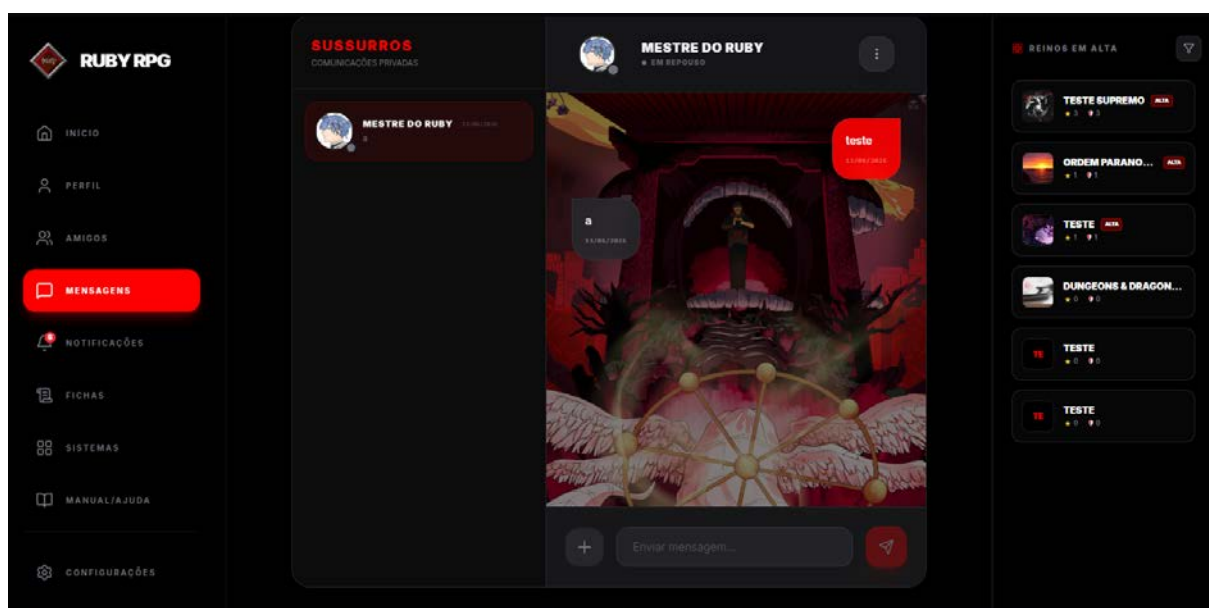


Figura 57 – Chat de conversa

Fonte: Criado pela equipe

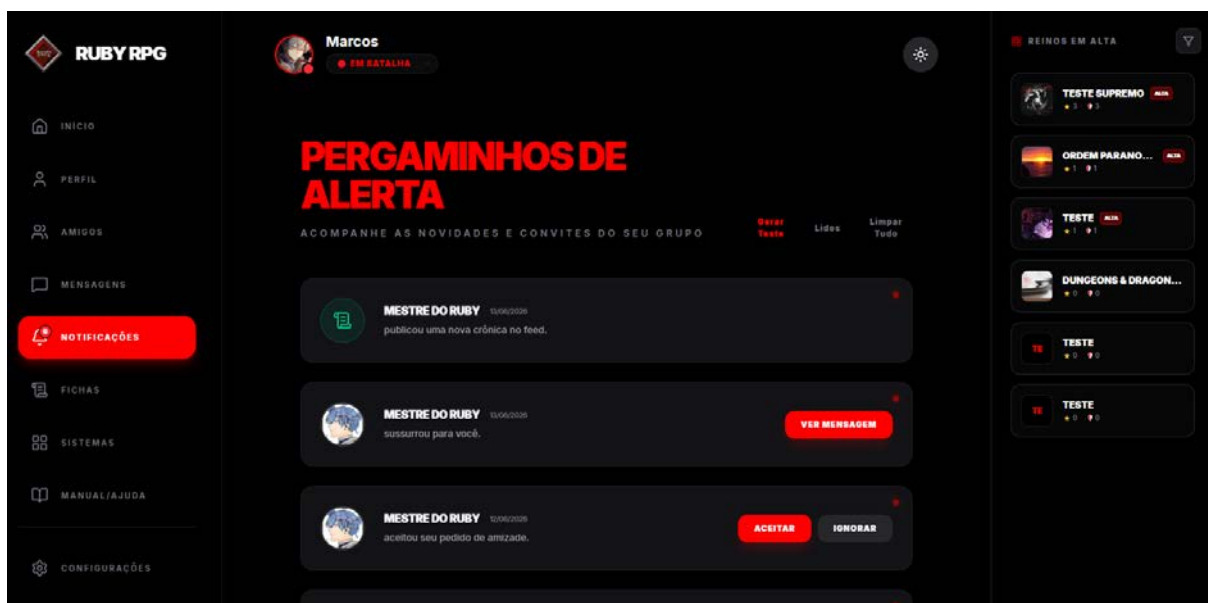


Figura 58 - Notificações

Fonte: Criado pela equipe

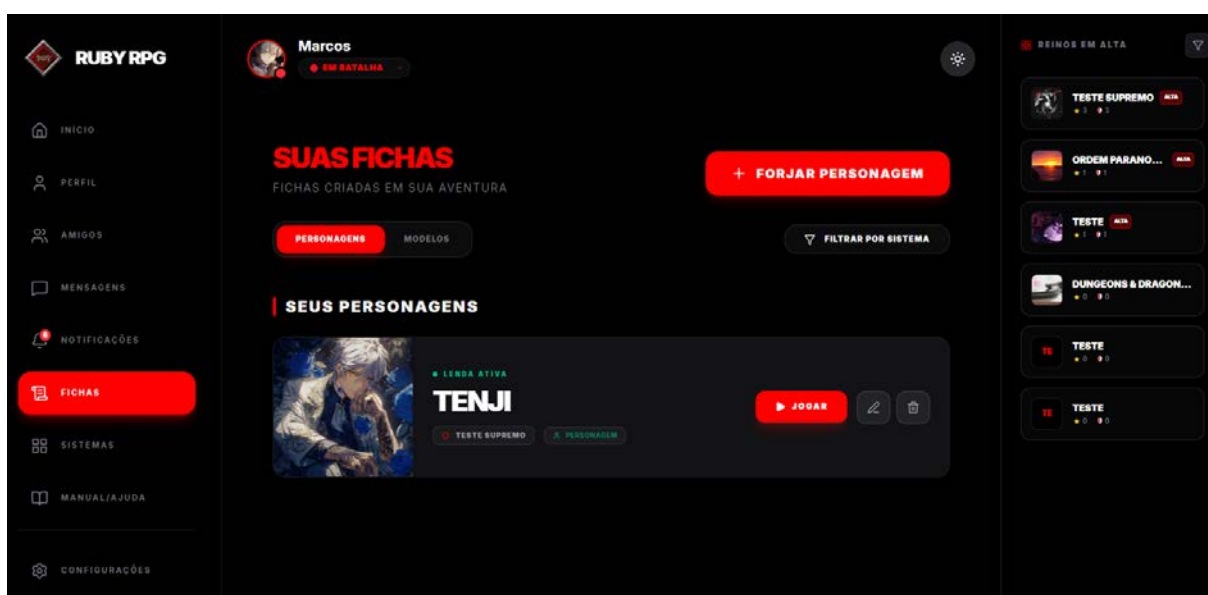


Figura 59 – Ficha de personagem

Fonte: Criado pela equipe

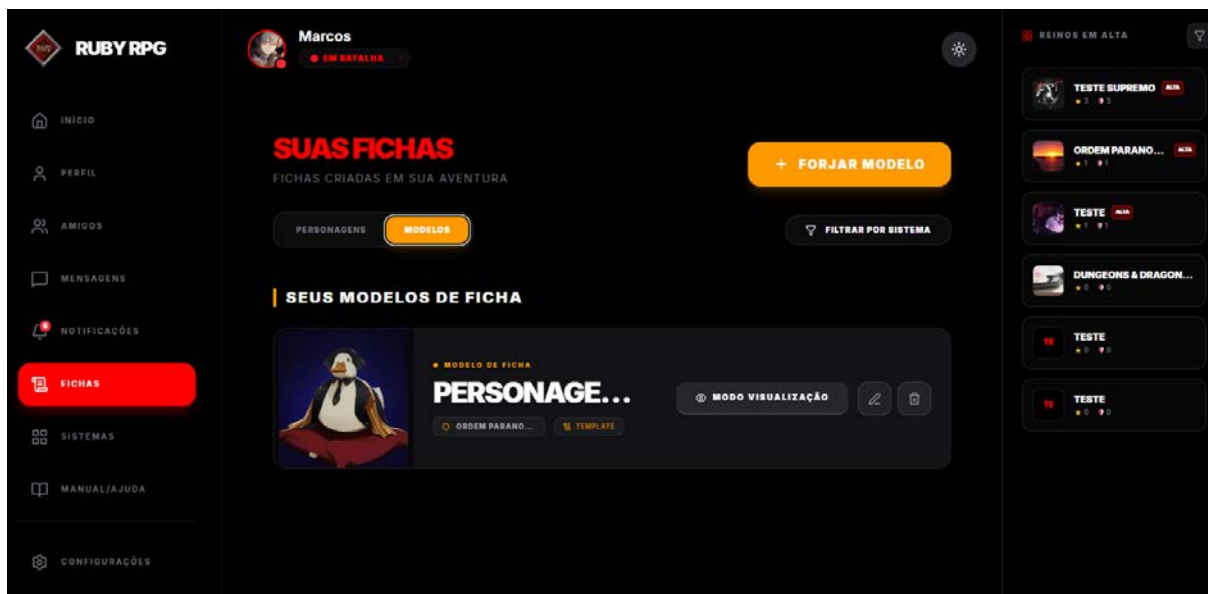


Figura 60 – Modelo de ficha

Fonte: Criado pela equipe

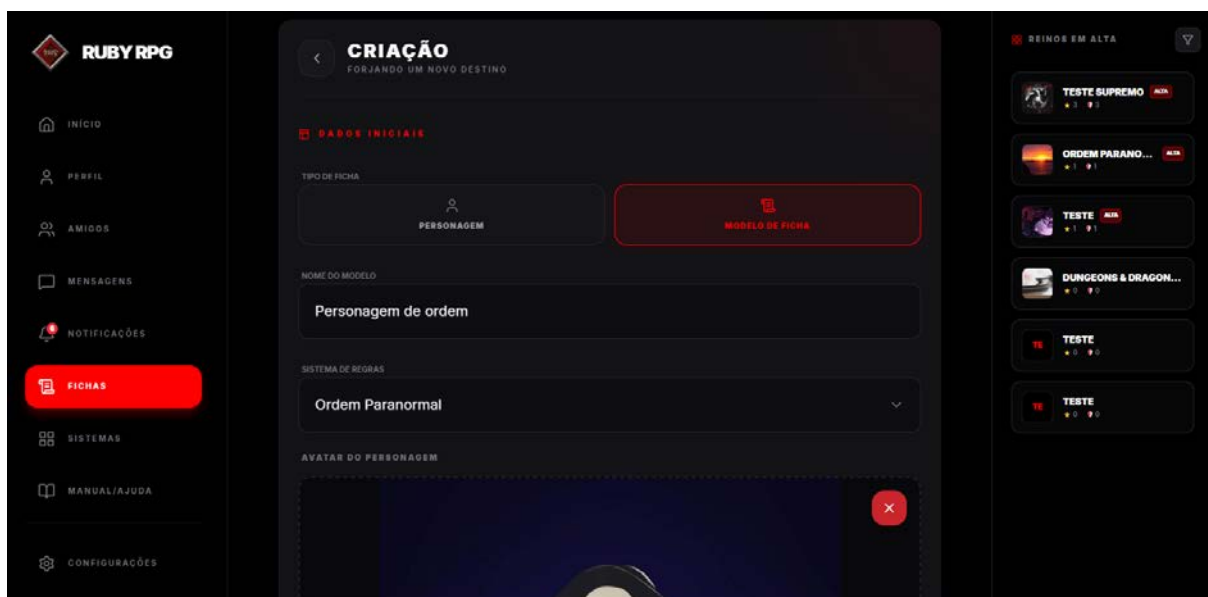


Figura 61 – Criação de ficha

Fonte: Criado pela equipe

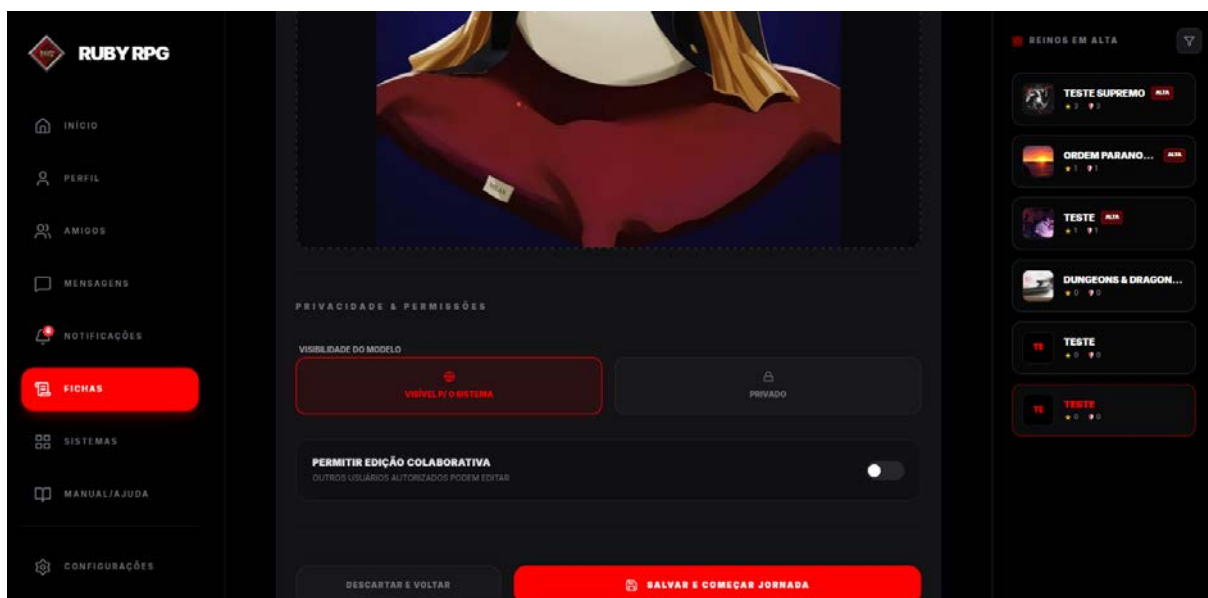


Figura 62 – Criação de ficha

Fonte: Criado pela equipe

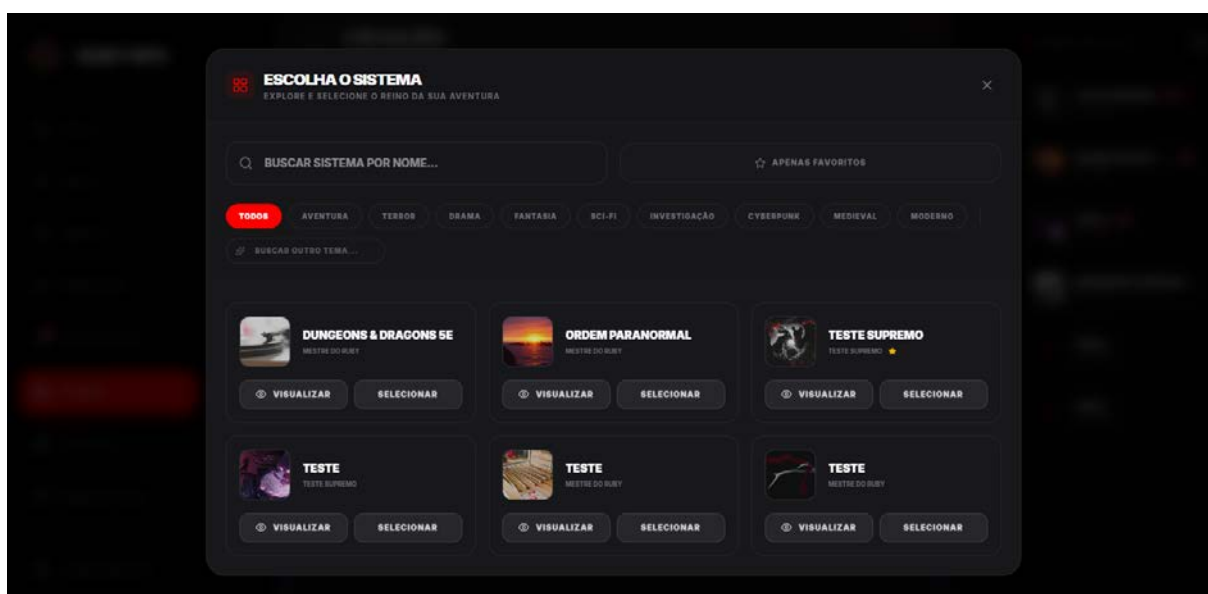


Figura 63 – Criação de ficha (seleção de sistema)

Fonte: Criado pela equipe

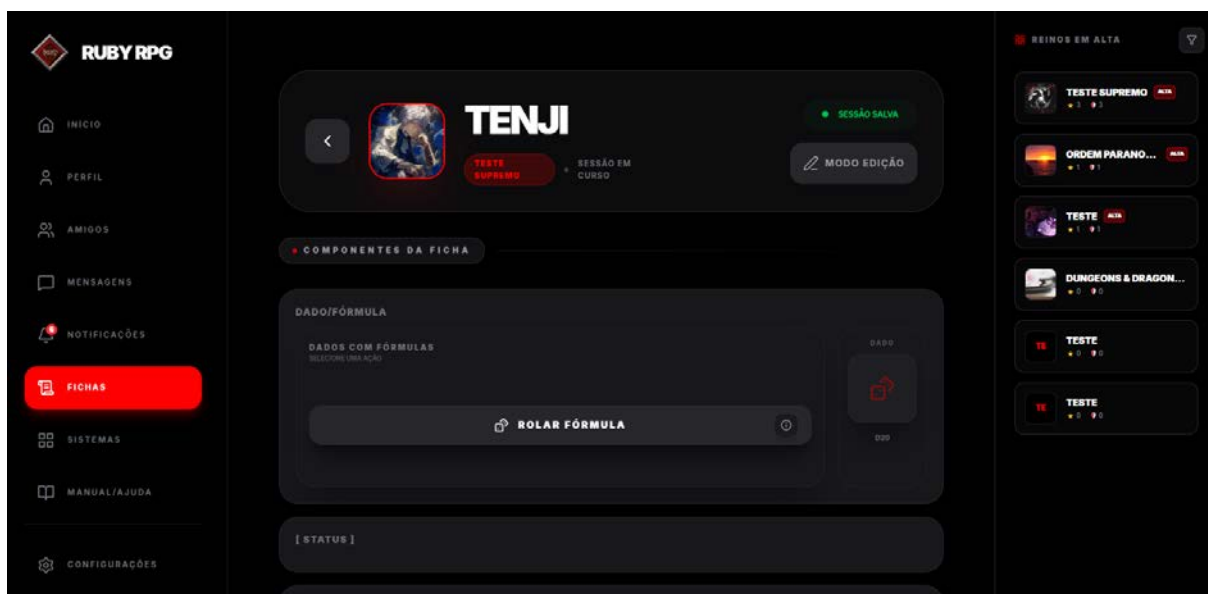


Figura 64 – Ficha de personagem

Fonte: Criado pela equipe

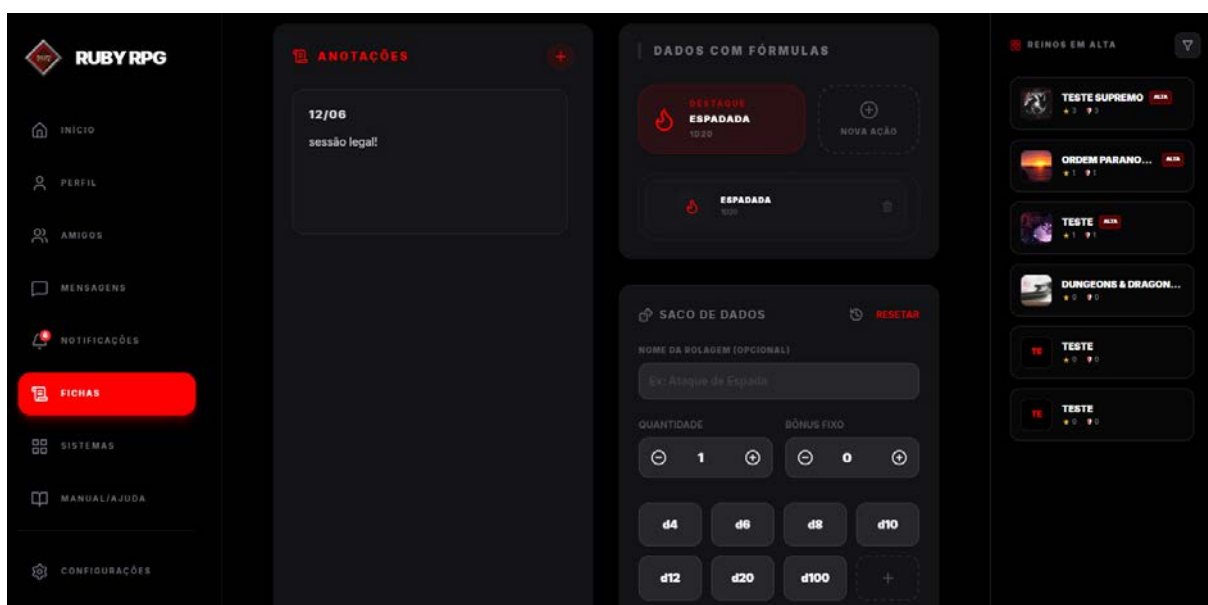


Figura 65 – Ficha de personagem

Fonte: Criado pela equipe

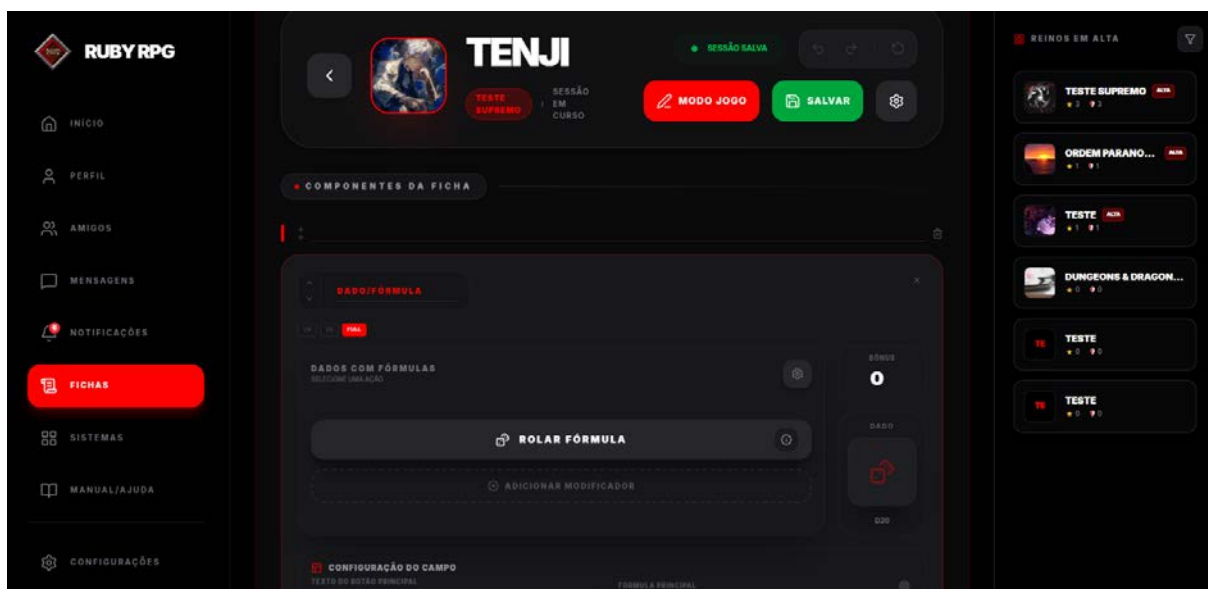


Figura 66 – Edição de ficha de personagem

Fonte: Criado pela equipe

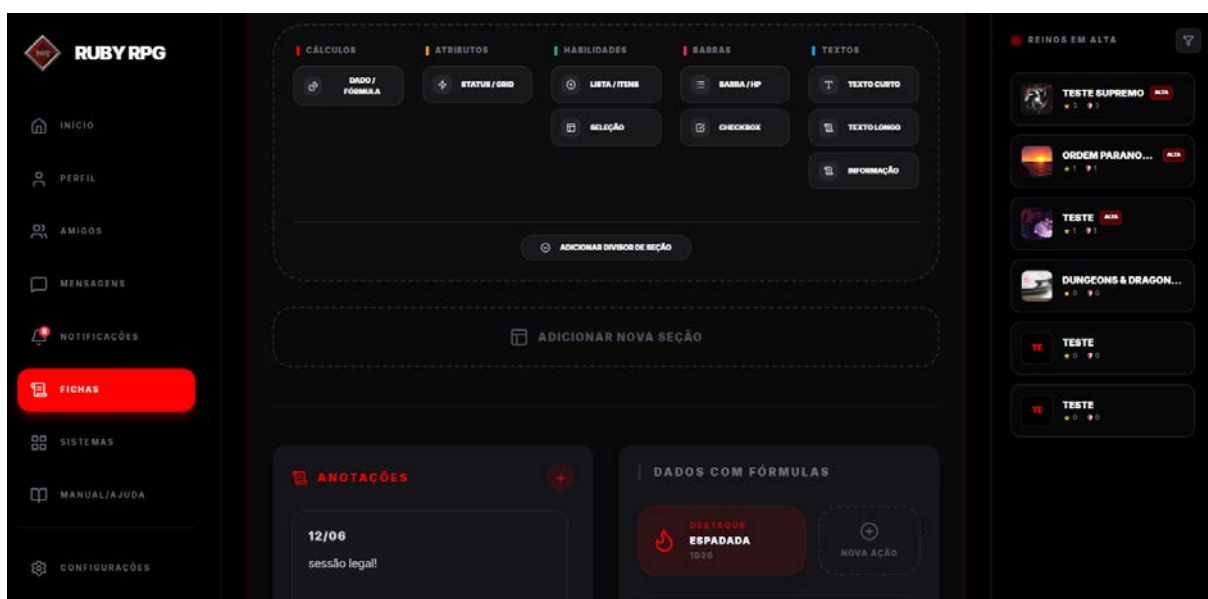


Figura 67 – Edição de ficha de personagem (adição de funcionalidades)

Fonte: Criado pela equipe

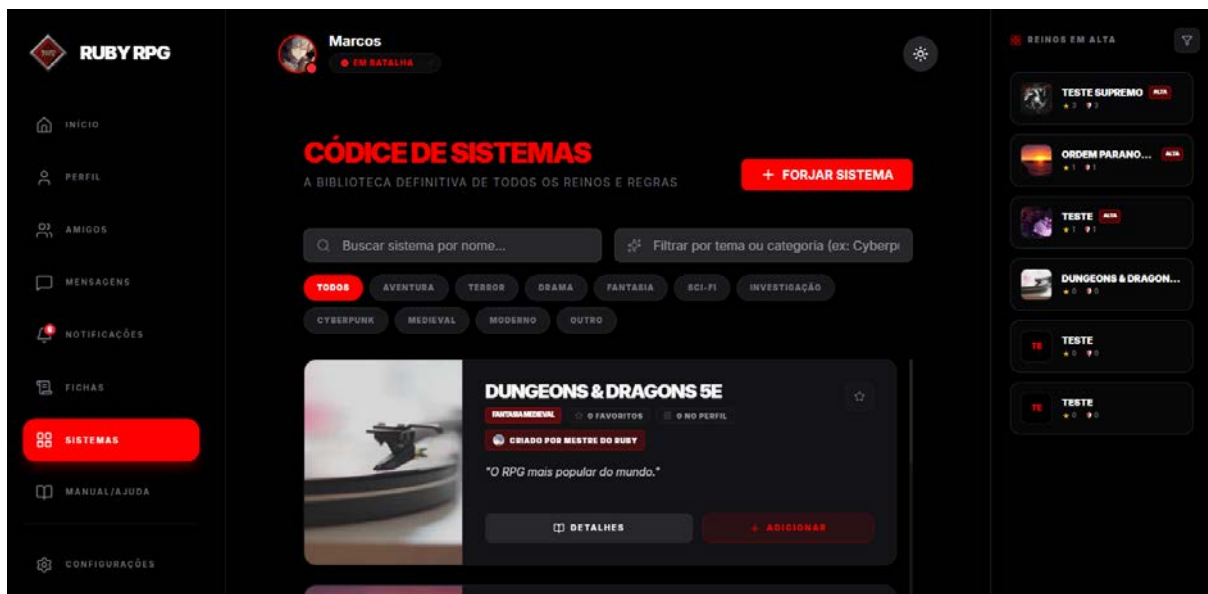


Figura 68 - Sistemas

Fonte: Criado pela equipe

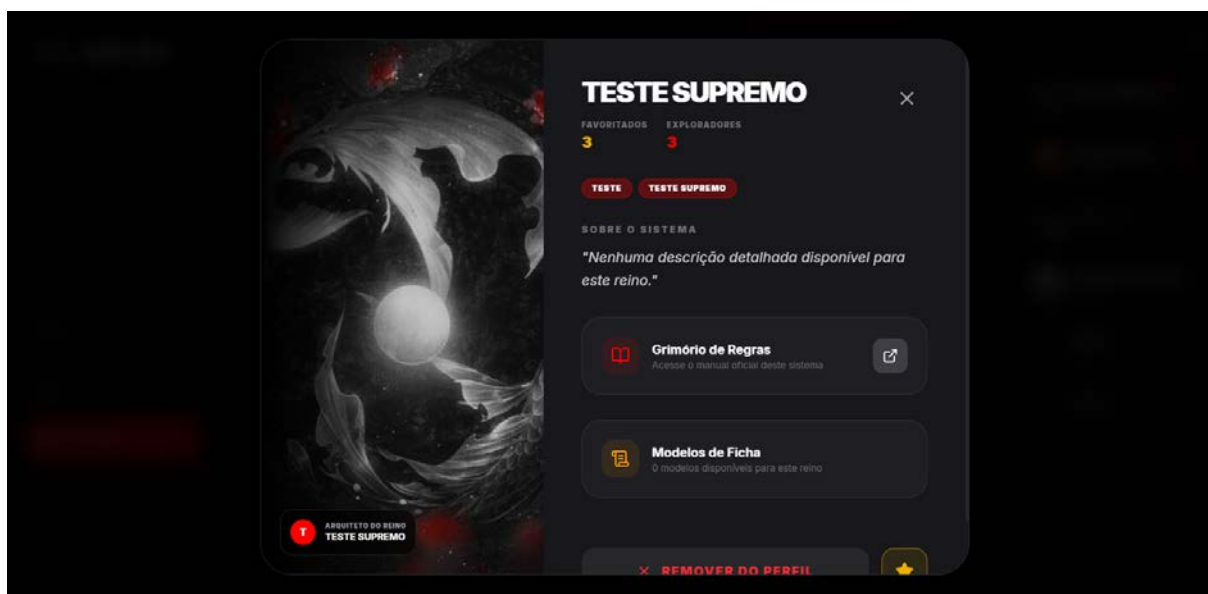


Figura 69 – Perfil de sistema

Fonte: Criado pela equipe

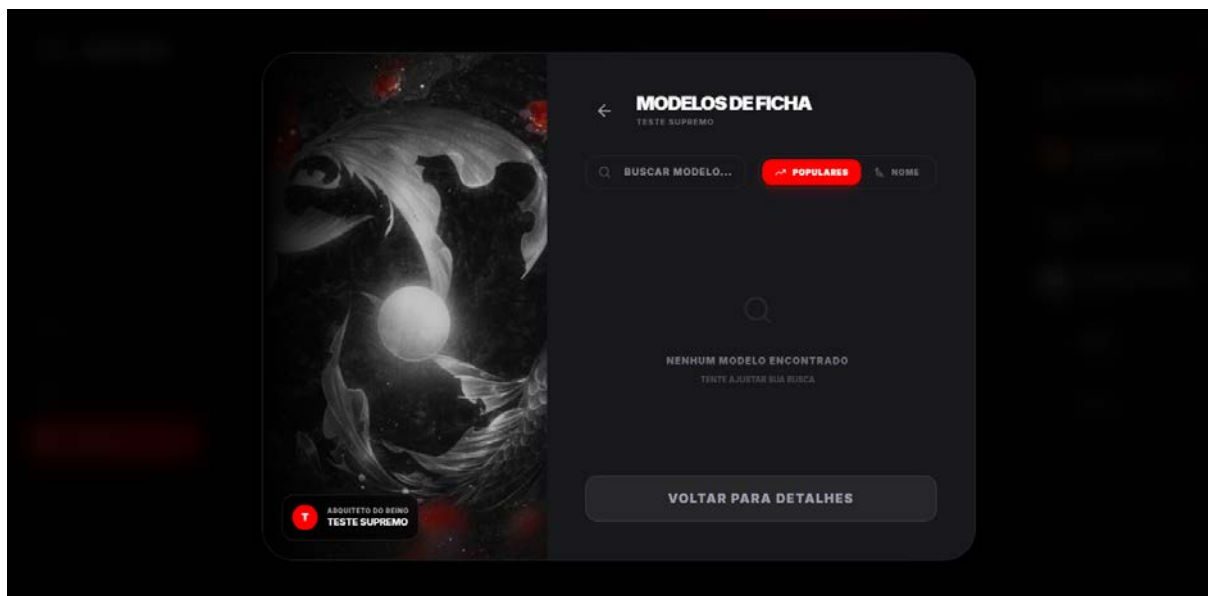


Figura 70 – Modelos de fichas de sistemas

Fonte: Criado pela equipe

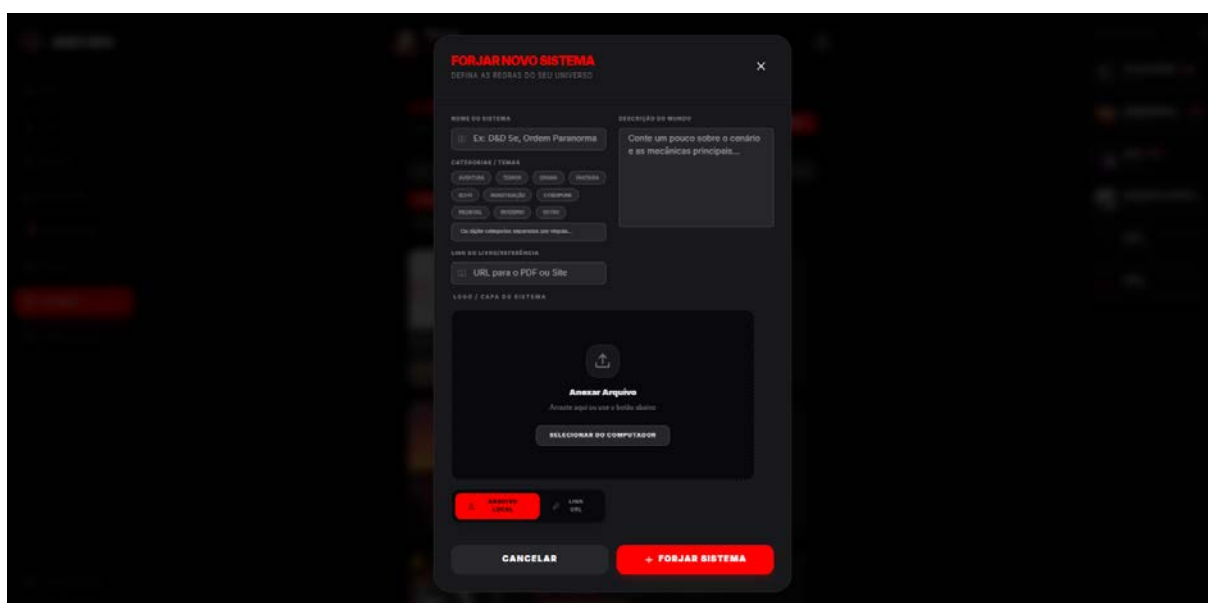


Figura 71 – Criação de sistema

Fonte: Criado pela equipe

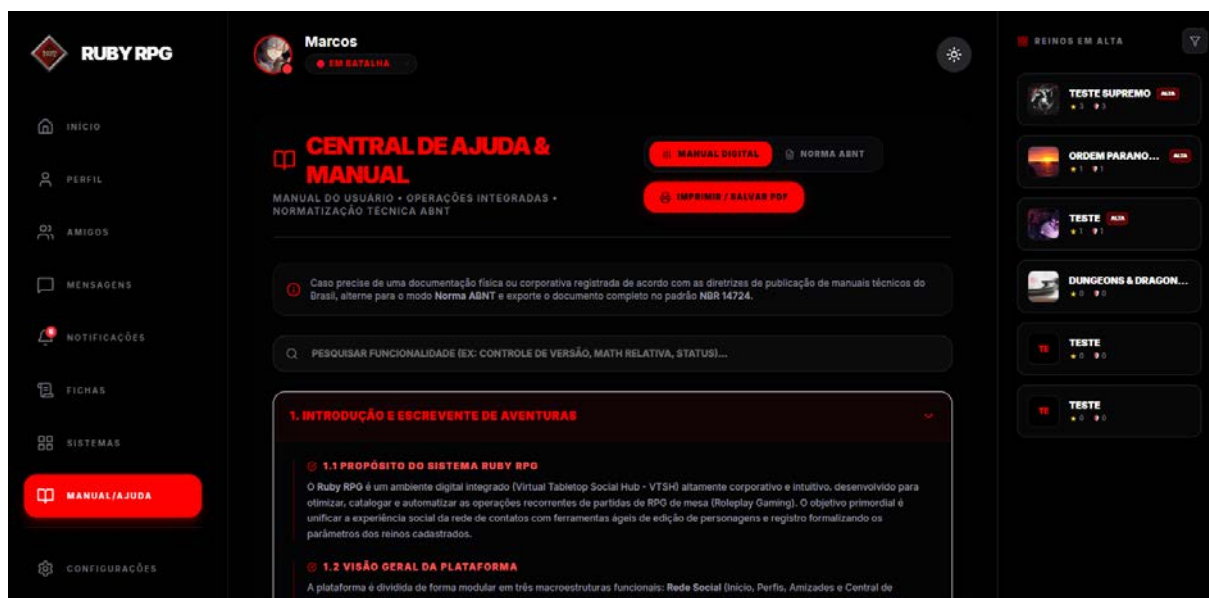


Figura 72 – Manual de usuário

Fonte: Criado pela equipe

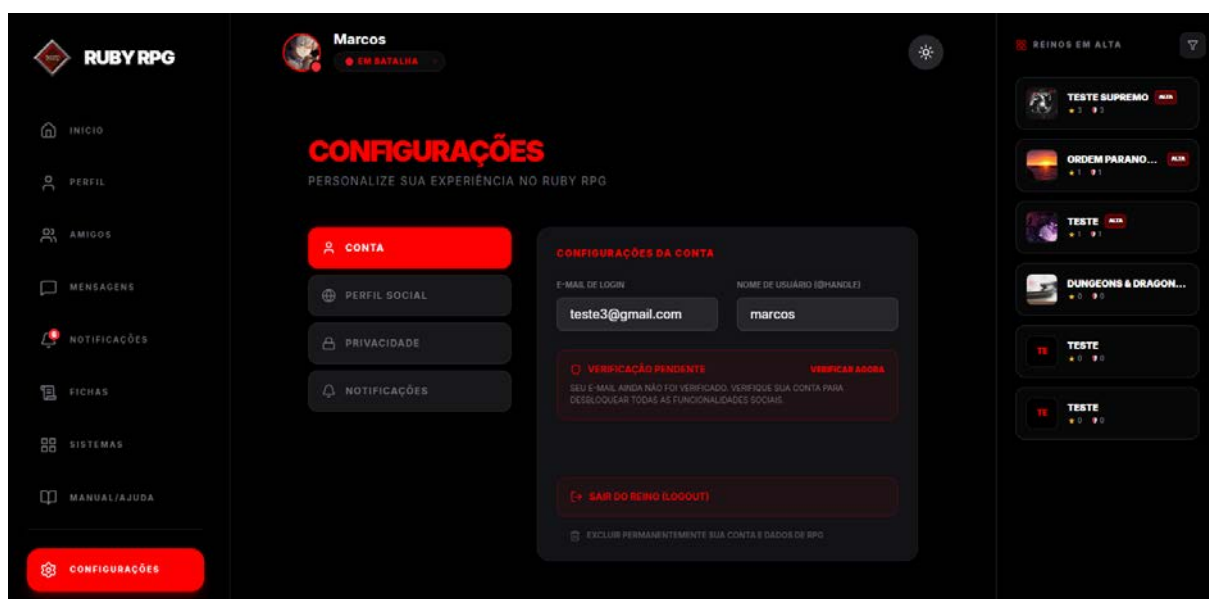


Figura 73 – Configuração de conta geral

Fonte: Criado pela equipe

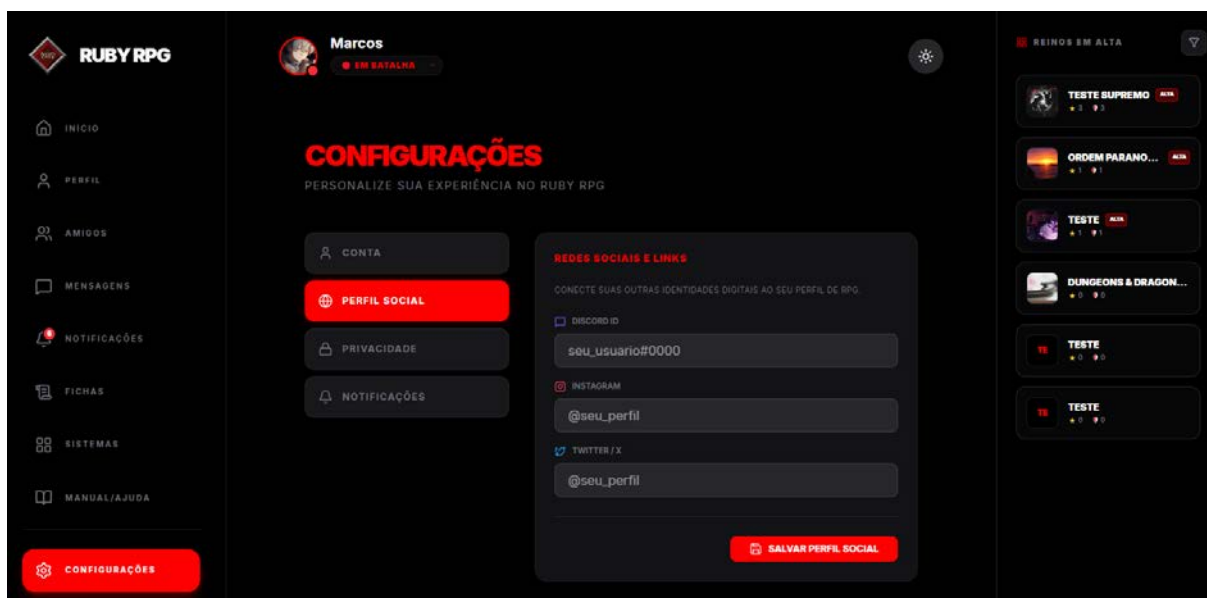


Figura 74 – Configuração de conta (redes sociais)

Fonte: Criado pela equipe

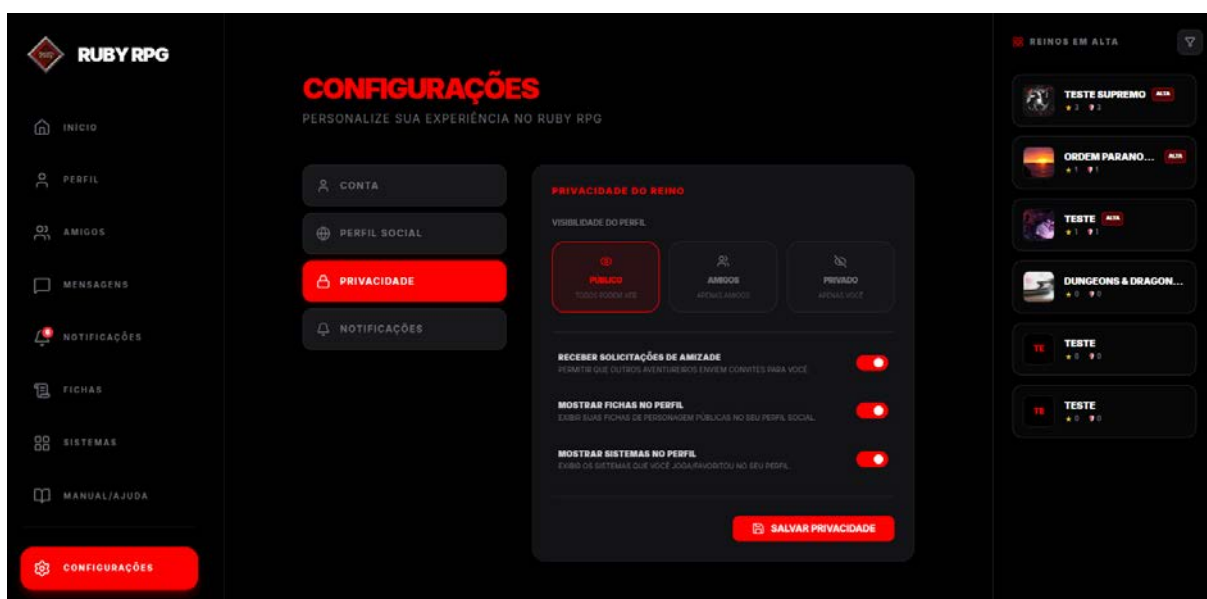


Figura 75 – Configuração de conta (visibilidade/privacidade)

Fonte: Criado pela equipe



Figura 76 – Configuração de conta (notificações)

Fonte: Criado pela equipe

4.7 Trecho do Código Fonte

```
// ♦ LÓGICA DE DADOS

const rolarDado = (lados: number, bonus: number = 0, label: string = 'Dado',
qtd: number = 1, formulaCustom?: string, tempMod: number = 0) => {

  let formula = formulaCustom || `${qtd}d${lados}${bonus !== 0 ? (bonus > 0 ?
`+${bonus}` : bonus) : ''}`;

  // Adicionar modificador temporário se existir
  if (tempMod !== 0) {
    formula = `(${formula}) ${tempMod > 0 ? '+' : ''}${tempMod}`;
  }

  // 1. Resolver variáveis da ficha (ex: "Força", "Força:1")
  let formulaProcessada = formula;
  const varRegex = /"([^\"]+)"/g;

  formulaProcessada = formulaProcessada.replace(varRegex, (match,
varLabel) => {
    let valorEncontrado: any = null;
    const parts = varLabel.split(':');
```

```

const rowOrFieldLabel = parts[0].trim().toLowerCase();
const colHeaderOrIndex = parts[1]??.trim().toLowerCase();

for (const secao of fichaEstrutura.sections) {
  // 1. Busca em campos diretos (se não houver coluna especificada)
  if (!colHeaderOrIndex) {
    const campo = secao.fields.find(f => f.label.toLowerCase() ===
rowOrFieldLabel);
    if (campo) {
      if (campo.type === 'status_value' || campo.type === 'status_mod') {
        valorEncontrado = calcularStatus(campo.statusColumns![0],
undefined, campo);
        break;
      } else if (campo.type !== 'status') {
        valorEncontrado = campo.value;
        break;
      }
    }
  }
}

// 2. Busca dentro de campos de Status
for (const f of secao.fields) {
  if (f.type === 'status' && Array.isArray(f.value)) {
    const row = f.value.find((r: StatusRow) => r.label.toLowerCase() ===
rowOrFieldLabel);
    if (row) {
      let targetCol = null;
      if (colHeaderOrIndex) {
        // Tenta buscar por índice (1-based)
        const colIndex = parseInt(colHeaderOrIndex);
        if (!isNaN(colIndex) && f.statusColumns) {
          targetCol = f.statusColumns[colIndex - 1];
        }
      }
    }
  }
}

```

```

        // Se não encontrou por índice, tenta por header
        if (!targetCol) {
            targetCol = f.statusColumns?.find(c => c.header.toLowerCase() ===
colHeaderOrIndex);
        }
        } else {
            // Padrão: primeira coluna de tipo 'value'
            targetCol = f.statusColumns?.find(c => c.type === 'value');
        }

        if (targetCol) {
            valorEncontrado = calcularStatus(targetCol, row, f);
            break;
        }
    }
}
}
if (valorEncontrado !== null) break;
}

if (valorEncontrado !== null) {
    // Tratar vírgula como ponto decimal e remover sinal de "+" redundante
    let valorLimpo = String(valorEncontrado).replace(',', '.').replace(/^\+/, "");

    // Se for apenas um número, não colocamos parênteses para não quebrar
a notação de dados (ex: 1d"TRICK" -> 1d20)
    if (/^-?\d+(\.\d+)?$/i.test(valorLimpo)) {
        return valorLimpo;
    }
    return `(${valorLimpo})`;
}

return `(0)`; // Retorna 0 se não encontrar para não quebrar a fórmula
});

```

// 2. Substituir colchetes remanescentes por parênteses (caso o usuário use para agrupamento)

```
formulaProcessada = formulaProcessada.replace(/\[/g, '(').replace(/\]/g, ');');
```

// 3. Regex para encontrar dados (ex: 1d20, 2d6, d100, 4dF)

```
const diceRegex = /(\d+)?d(\d+|F)/gi;
```

```
const individuais: number[] = [];
```

```
const rollSingleFormula = (f: string) => {
```

```
  let fRoll = f.replace(diceRegex, (match, qtdStr, ladosStr) => {
```

```
    const q = qtdStr ? parseInt(qtdStr) : 1;
```

```
    let total = 0;
```

```
    if (ladosStr.toUpperCase() === 'F') {
```

```
      // Fudge Dice: [+1, +1, -1, -1, 0, 0]
```

```
      for (let i = 0; i < q; i++) {
```

```
        const r = Math.floor(Math.random() * 3) - 1; // -1, 0, 1
```

```
        total += r;
```

```
        individuais.push(r);
```

```
      }
```

```
    } else {
```

```
      const l = parseInt(ladosStr);
```

```
      for (let i = 0; i < q; i++) {
```

```
        const r = Math.floor(Math.random() * l) + 1;
```

```
        total += r;
```

```
        individuais.push(r);
```

```
      }
```

```
    }
```

```
    return total.toString();
```

```
  });
```

```
  try {
```

```
    // Limpa a fórmula final de caracteres estranhos antes de avaliar
```

```
    const safeRoll = fRoll.replace(/[^\d-9.\s+\/\(\)]/g, "").trim();
```

```
    const cleanRoll = safeRoll.replace(/[+\/-*/\(\)]$/g, "");
```

```

    return new Function(`return (${cleanRoll || '0'})` )();
  } catch (e) {
    console.error("Erro ao avaliar fórmula:", e);
    return 0;
  }
};

```

```

// 4. Lógica de Rolagem Múltipla (#)
if (formulaProcessada.includes('#')) {
  const parts = formulaProcessada.split('#');
  const count = parseInt(parts[0]) || 1;
  const subFormula = parts[1];
  const results: number[] = [];
  for (let i = 0; i < count; i++) {
    results.push(rollSingleFormula(subFormula));
  }
  setResultadoDado({
    natural: 0,
    bonus: tempMod + bonus,
    total: results,
    formula,
    label,
    individuais,
    isMultiplo: true,
    timestamp: Date.now()
  });
  adicionarLog({
    label,
    lados: formulaCustom ? 'Fórmula' : lados,
    qtd: formulaCustom ? 1 : qtd,
    bonus: tempMod + bonus,
    formula,
    total: results,
    individuais,

```

```

        isMultiplo: true
    });
} else {
    const totalFinal = rollSingleFormula(formulaProcessada);
    setResultadoDado({
        natural: totalFinal - (tempMod + bonus),
        bonus: tempMod + bonus,
        total: totalFinal,
        formula,
        label,
        individuais,
        isMultiplo: false,
        timestamp: Date.now()
    });
    adicionarLog({
        label,
        lados: formulaCustom ? 'Fórmula' : lados,
        qtd: formulaCustom ? 1 : qtd,
        bonus: tempMod + bonus,
        formula,
        total: totalFinal,
        individuais,
        isMultiplo: false
    });
}

setTimeout(() => setResultadoDado(null), 10000);
};
const [nomePersonagem, setNomePersonagem] = useState("");
const [imagemPersonagem, setImagemPersonagem] = useState("");
const [sistemaId, setSistemaId] = useState("");
const [tpPrivacidade, setTpPrivacidade] = useState<'publico' | 'amigos' | 'privado'>('privado');
const [stPermitirEdicao, setStPermitirEdicao] = useState(false);

```

```

const [editoresSelecionados, setEditoresSelecionados] =
useState<number[]>([]);
const [visualizadoresSelecionados, setVisualizadoresSelecionados] =
useState<number[]>([]);
const [amigos, setAmigos] = useState<any[]>([]);
const [userConfig, setUserConfig] = useState<any>(null);
const privacyLevels: Record<string, number> = { 'publico': 0, 'amigos': 1,
'privado': 2 };

const [showEditorSelector, setShowEditorSelector] = useState(false);
const [configMod, setConfigMod] = useState<{ secaoId: string, campId:
string, colId: string, rowId?: string } | null>(null);
const [showColumnManager, setShowColumnManager] = useState<{
secaoId: string, campId: string } | null>(null);
const [activeConfigTab, setActiveConfigTab] = useState<'aumento' |
'formula'>('aumento');

useEffect(() => {
  if (configMod) {
    setActiveConfigTab('aumento');
  }
}, [configMod]);
const [confirmDeleteId, setConfirmDeleteId] = useState<string | null>(null);
const [showFormulaManual, setShowFormulaManual] = useState(false);

const calcularStatus = (col: StatusColumnConfig, row?: StatusRow, campo?:
FichaField) => {
  // Se for um campo standalone (status_value ou status_mod)
  if (campo && (campo.type === 'status_value' || campo.type ===
'status_mod')) {
    const config = campo.statusColumns?.[0];
    if (!config) return campo.value;

    const dependsOn = config.dependsOn;

```

```

const ratioValue = config.ratioValue;
const ratioMod = config.ratioMod;

if (!dependsOn || !ratioValue || !ratioMod) return campo.value;

// Encontrar o campo de dependência na ficha inteira
let valorBase = 0;
fichaEstrutura.sections.forEach(s => {
  s.fields.forEach(f => {
    if (f.id === dependsOn) {
      valorBase = parseFloat(String(f.value).replace(',', '.')) || 0;
    }
  });
});

const resultado = Math.floor(valorBase / ratioValue) * ratioMod;
return config.type === 'mod' ? (resultado >= 0 ? `+${resultado}` :
`${resultado}`) : resultado;
}

// Se for parte de uma tabela de status
if (!row || !campo) return 0;
const cell = row.cells[col.id];
const useOverride = cell?.overrideConfig ?? false;

// Lógica de Fórmula (Prioritária se ativa)
if (useOverride && cell?.formula) {
  try {
    let formula = cell.formula;
    // Regex para referências tipo "label:index" ou "label"
    const regex = /"([^\"]+)"/g;

    formula = formula.replace(regex, (match, content) => {
      const parts = content.split(':');

```

```

const label = parts[0].trim().toLowerCase();
const indexStr = parts[1]?.trim();

const targetRow = (campo.value as StatusRow[]).find(r =>
  r.label.toLowerCase().trim() === label ||
  r.label.toLowerCase().normalize("NFD").replace(/[\u0300-\u036f]/g,
    "").trim() === label.normalize("NFD").replace(/[\u0300-\u036f]/g, "").trim()
  );

if (!targetRow) return '0';

let targetColRef = null;
if (indexStr) {
  const targetColIndex = parseInt(indexStr) - 1;
  targetColRef = campo?.statusColumns?.[targetColIndex];
} else {
  // Padrão: primeira coluna de tipo 'value'
  targetColRef = campo?.statusColumns?.find(c => c.type === 'value');
}

const targetCellId = targetColRef?.id;

if (!targetCellId || !targetRow.cells[targetCellId]) return '0';

const targetCell = targetRow.cells[targetCellId];
const targetCol = { id: targetCellId, type: targetCell.type || 'value', header:
targetCell.header || " } as StatusColumnConfig;

// Chama calcularStatus recursivamente para obter o valor final (manual
ou calculado)
const targetCellValue = calcularStatus(targetCol, targetRow, campo);

// Limpa o valor para conversão numérica
const cleanValue = String(targetCellValue)

```

```
.replace(/[^\d.,-]/g, "") // Remove tudo que não for número, ponto, vírgula
ou sinal de menos
```

```
.replace(',', '.');
```

```
const numValue = parseFloat(cleanValue);
return isNaN(numValue) ? '0' : String(numValue);
});
```

```
// Avalia a fórmula de forma segura
```

```
// Remove caracteres potencialmente perigosos, permitindo apenas
matemática básica e colchetes para agrupamento
```

```
const safeFormula = formula.replace(/^[^0-9.\s+\-*/()\[\]]/g, "").replace(/\[/g,
'(').replace(/\]/g, ')').trim();
```

```
// Remove operadores pendentes no final para evitar erro de sintaxe
```

```
const cleanFormula = safeFormula.replace(/[+\-*/](?![\s,])/g, "");
```

```
const result = new Function(`return (${cleanFormula || '0'})` )();
```

```
// Garante que o resultado seja um número válido
```

```
const finalResult = isNaN(result) ? 0 : result;
```

```
return col.type === 'mod' ? (finalResult >= 0 ? `+${finalResult}` :
`${finalResult}`) : finalResult;
```

```
} catch (e) {
```

```
  console.error("Erro na fórmula:", e);
```

```
  return "ERR";
```

```
}
```

```
}
```

```
const dependsOn = useOverride ? (cell?.dependsOn ?? "") : col.dependsOn;
```

```
const ratioValue = useOverride ? (cell?.ratioValue ?? 1) : col.ratioValue;
```

```
const ratioMod = useOverride ? (cell?.ratioMod ?? 0) : col.ratioMod;
```

```
if (!dependsOn || !ratioValue || !ratioMod) return cell?.value || (col.type ===
'mod' ? '+0' : 0);
```

```

const valorBase =
parseFloat(String(row.cells[dependsOn]?.value).replace(',', '.')) || 0;
const resultado = Math.floor(valorBase / ratioValue) * ratioMod;

if (col.type === 'mod') {
  return resultado >= 0 ? `+${resultado}` : `-${resultado}`;
}
return resultado;
};

const [showQuickActionModal, setShowQuickActionModal] = useState(false);
const [showCustomDieModal, setShowCustomDieModal] = useState(false);
const [newAction, setNewAction] = useState<Partial<QuickAction>>({ name:
", dice: '1d20', modifier: 0, icon: 'Sword' });
const [newDie, setNewDie] = useState({ name: "", sides: 20 });

```

5. MANUAL DO USUÁRIO

1. INTRODUÇÃO E ESCRIVENTE DE AVENTURAS

O Ruby RPG configura-se como um núcleo tecnológico integrado focado no atendimento pleno de jogadores, mestres e entusiastas de RPG de mesa. Através de um projeto web escalável e responsivo, o sistema provê facilidades de gerenciamento que antes demandavam soluções desconexas.

1.1 Propósito do Sistema Ruby RPG

O escopo deste produto abrange a catalogação estruturada de regras, automação de testes de sorte por meio de rolares randômicos de alta integridade, e a constituição de feeds com engajamento comunitário de forma dinâmica. O software unifica fichas complexas com chats em tempo real.

1.2 Visão Geral da Plataforma

A plataforma foi desenhada no modelo de Single Page Application (SPA), garantindo tempo de resposta mínimo a cada transição de interface. Módulos dedicados à segurança garantem que cada usuário possua total poder de guarda e visibilidade sobre as suas informações geradas na rede.

2. ARQUITETURA DO PORTAL, CADASTRO E CONTROLE DE ACESSO

O sistema operacional requisita um rigor de controle de privacidade com o objetivo de guardar com integridade os arquivos e dados gerados. Um sistema de chave de sessões robusto no Express do backend cuida de blindar o tráfego de dados sensíveis corporativamente.

2.1 Processo de Registro e Autenticação Criptográfica

Toda senha cadastrada pelo usuário na interface passa por um gerador de sal (salt) e algoritmo de rotação segura antes de ser gravada fisicamente. Isso assegura que, mesmo em tentativas extremas de invasão externa, os segredos dos RPGistas fiquem permanentemente protegidos.

2.2 Recuperação de Acesso

A alteração de senhas obedece a regras rígidas de integridade. O usuário recebe confirmações por rotas autenticadas da API, permitindo realizar mudanças nos dados apenas mediante a apresentação de hashes válidas e chaves em formato HTTPS protegido.

3. A REDE SOCIAL E FEED OPERACIONAL (PÁGINA INICIAL)

A página inicial acolhe a totalidade das interações sociais, simulando uma corporação ativa de contação de histórias onde cada feito nas campanhas pode ser eternizado em texto público, amigos associados ou relatórios anexados.

3.1 Criação de Publicações Multi-mídia

O interpretador de postagens aceita marcações estendidas. Os uploads de fotos, banners e links de vídeos externos contam com controle de compressão automático no middleware bodyParser com teto de payload configurado em 50MB, otimizando o carregamento da página global de feeds.

3.2 Níveis de Privacidade de Postagens

Filtros indexados no banco relacional do Ruby RPG garantem o mascaramento correto das publicações na linha do tempo. No instante em que um post é sinalizado como "Somente Amigos", o motor de exibição consulta o diretório de conexões do emissor para bloquear qualquer leak visual de dados para não autorizados.

4. DIRETÓRIO GLOBAL DE REINOS (SISTEMAS)

Os sistemas formam o baralho de parâmetros de leis físicas de cada universo fictício. Através deste módulo, criadores publicam sistemas proprietários ou mestres localizam regras célebres do mercado.

4.1 Navegação de Sistemas e Classificações

A interface de exploração apresenta grades ricas de rankings. Os reinos ativos computam favoritados em tempo real, gerando escores de tração que ajudam na recomendação de novas regras aos jogadores ativos.

4.2 Cadastro de Novos Reinos e Ajuste de Imagem sem Tarja Preta

O portal combate o efeito de cortes automáticos grosseiros de layout. Cada imagem submetida para capas e heróis passa por redimensionamento assistido que impede barras pretas no fundo. Ele aceita o ajuste da escala em zoom livre encostando precisamente no bocal limite do frame do sistema.

5. GERENCIADOR DINÂMICO DE FICHAS DE PERSONAGENS

O editor de fichas é o epicentro operacional da plataforma, combinando flexibilidade total com recursos de bancos corporativos de persistência continuada e controle reativo.

5.1 Criação Baseada em Template ou Criação Livre

Modelos cadastrados nos sistemas salvam o tempo precioso dos jogadores. Se o mestre estruturou regras para o reino, a nova ficha herda imediatamente blocos de

estatísticas integradas; do contrário, o jogador pode modelar suas seções conforme achar conveniente.

5.2 Estrutura Modular: Seções, Linhas e Colunas

O aplicativo utiliza grids fluidos para criar layouts. Dentro de cada seção, o operador insere novos campos numéricos, de texto para histórias com rolagem interna, listas ou botões de gatilhos automáticos rápidos, suportando rearranjos estruturais sem quebra de fluxo.

5.3 Atributos, Matemática Relativa (+5, -3) e Fórmulas

Um dos maiores destaques de usabilidade da ficha é a aceitação de fórmulas relativas. No momento de combates rápidos ou perdas e ganhos de vida, o operador apenas introduz o modificador direto. Por exemplo, inserir -12 ou +5 na entrada numérica recalcula o montante atual utilizando o valor original como base histórica, evitando lapsos de contabilidade.

5.4 Rolador Integrado de Dados

O software possui um parser nativo de notação de dados padrão (ex: 2d20+4). Ao acionar qualquer estatística mapeada como rolável, o terminal lateral reporta isoladamente cada sorteio de dado combinado às somas de modificadores ativos de maneira auditável.

5.5 Logs Históricos de Desfazer/Refazer (Undo/Redo) e Auto-Save

Todas as fichas incluem proteção contra acidentes críticos. Cada alteração de estrutura ou valores alimenta uma pilha de histórico em memória de 30 posições. Botões visuais proeminentes permitem o retorno a qualquer estado retroativo com um clique. Toda alteração é persistida em background em transações automáticas não agressivas no banco de dados.

4. CONCLUSÃO

O RPG de mesa consolidou-se ao longo das últimas décadas como uma importante forma de entretenimento colaborativo, promovendo interação social, criatividade e construção coletiva de narrativas. Com o avanço das tecnologias digitais, a prática do RPG passou por transformações significativas, incorporando ferramentas online que ampliaram sua acessibilidade e permitiram a realização de campanhas entre participantes localizados em diferentes regiões geográficas.

Durante o desenvolvimento deste trabalho, foi possível analisar a trajetória histórica do RPG de mesa, sua relevância social e cultural, bem como sua adaptação ao ambiente digital por meio das redes sociais de nicho e das plataformas Virtual Tabletop (VTT). A pesquisa bibliográfica demonstrou que, apesar da evolução tecnológica observada nos últimos anos, ainda existe uma fragmentação significativa entre as ferramentas utilizadas pelos jogadores e mestres para organizar e participar de campanhas online.

A partir dessa problemática, foi proposta a criação da plataforma Ruby RPG | Panteon Table, concebida como uma rede social especializada voltada à comunidade de RPG de mesa. A proposta busca integrar funcionalidades de interação social, gerenciamento de campanhas e recursos específicos para RPG em um único ambiente digital, reduzindo a necessidade de utilização simultânea de múltiplas plataformas.

No âmbito do desenvolvimento do projeto, foram aplicados conhecimentos adquiridos ao longo do curso técnico, abrangendo áreas como desenvolvimento web, banco de dados, engenharia de software, modelagem de sistemas e experiência do usuário. A utilização de tecnologias modernas, como TypeScript, React, Node.js, Express e MySQL, permitiu a construção de uma base sólida para a implementação das funcionalidades propostas.

Entretanto, devido às limitações de tempo, recursos e equipe disponíveis para a realização do Trabalho de Conclusão de Curso, optou-se por reduzir o escopo inicialmente planejado. Dessa forma, foram priorizadas as funcionalidades essenciais para validação da proposta, especialmente os recursos relacionados à rede social e ao gerenciamento de fichas de personagens, estabelecendo uma estrutura capaz de servir como base para futuras expansões.

Conclui-se, portanto, que a proposta apresentada possui potencial para contribuir com a comunidade de RPG de mesa ao oferecer uma solução especializada para um nicho de mercado ainda pouco explorado. Além disso, o projeto evidencia a importância da aplicação dos conhecimentos técnicos adquiridos durante a formação profissional na criação de soluções tecnológicas capazes de atender demandas reais da sociedade.

Como perspectiva futura, espera-se ampliar gradualmente as funcionalidades da plataforma, incorporando recursos avançados de Virtual Tabletop, sistemas de comunicação em tempo real, ferramentas de automação para mestres e integrações com inteligência artificial, aproximando cada vez mais a Ruby RPG da visão originalmente concebida para o projeto.

5. REFERÊNCIA BIBLIOGRÁFICA

APRYSE. Get started with WebViewer using Vite. Disponível em: <https://docs.apryse.com/web/guides/get-started/vite>. Acesso em: 15/04/2026.

ÁRAUJO, Raony, RAMALHO, Gaber. Narrativa e Jogos Digitais: Lições do RPG de Mesa. Pernambuco: Universidade Federal de Pernambuco. 4 páginas. Disponível em: <https://www.cin.ufpe.br/~sbgames/proceedings/aprovados/23764.pdf>. Acesso: 27/11/2025.

BARROS, Isabelle, SANTOS, Carlos. Apostila de Introdução ao CSS. Niterói: Universidade Federal Fluminense (UFF) – Grupo PET-Tele, 2008. Disponível: <https://www.telecom.uff.br/pet/petws/downloads/tutoriais/css/css2k80912.pdf>. Acesso: 20/03/2026.

BROWN, Ethan. Programação web com Node e Express, beneficiando-se da stack JavaScript. São Paulo: Novatec Editora Ltda, 2020. 451 páginas. Disponível em: <https://www.kufunda.net/publicdocs/Programa%C3%A7%C3%A3o%20web%20com%20Node%20e%20Express-Ethan%20Brown.pdf>. Acesso em: 15/04/2026

CALDEIRA, Carlos. Introdução ao HTML (HyperText Markup Language). Évora: Universidade de Évora – Departamento de Informática, 2015. 45 páginas. Disponível: <https://dspace.uevora.pt/rdpc/bitstream/10174/13240/1/Introdu%C3%A7%C3%A3o%20ao%20HTML.pdf>. Acesso: 20/03/2026.

CARMO, Airine, RICON, Maya, XEXÉO, Geraldo. O RPG e o uso de Ferramentas on-line um estudo exploratório. Rio de Janeiro: SBC – Proceedings of SBGames, 2018. 4 páginas. Disponível: <https://www.sbgames.org/sbgames2018/files/papers/CulturaShort/188376.pdf>. Acesso: 27/11/2025.

CHIAPPA, João, RIBEIRO, Humberto. RPG de Mesa na era digital: Ferramentas e plataformas para jogar on-line. Uberaba: Universidade de Uberaba (Uniube), 2023. 5 páginas. Disponível: <https://repositorio.uniube.br/bitstream/123456789/2678/1/RPG%20de%20mesa%20n>

[a%20era%20digital%20ferramentas%20e%20plataformas%20para%20jogar%20on-line.pdf](#). Acesso: 27/11/2025.

FILHO, Cláudio, ALBUQUERQUE, Rafael. Notas sobre a história dos RPGs (RolePlaying Games) de mesa brasileiros. Londrina: Projética, 2021. 15 páginas. Disponível: <https://ojs.uel.br/revistas/uel/index.php/projetica/article/download/39900/30838/230071>. Acesso: 27/11/2025.

GOLDBERG, Josh. Aprendendo TypeScript, melhore suas habilidades de desenvolvimento usando JavaScript Type-Safe. São Paulo: Novatec Editora Ltda, 2022. 354 páginas. Disponível em: <https://pt.scribd.com/document/778280691/Aprendendo-TypeScript-Josh-Goldberg-Z-Library>. Acesso em: 15/04/2026.

GRILLO, Filipe, FORTES, Renata. Aprendendo JavaScript. São Carlos: ReP (Repositório da Produção USP). 47 páginas. Disponível: https://repositorio.usp.br/directbitstream/4cd7f9b7-7144-40f4-bfd0-7a1d9a6bd748/nd_72.pdf. Acesso: 20/03/2026

JUNIOR, Edson. Sistema gerenciador de planejamento de aula on-line: SGPA On-line. Ponta grossa: Universidade Tecnológica Federal do Paraná, 2018. Disponível em: https://repositorio.utfpr.edu.br/jspui/bitstream/1/16812/1/PG_COADS_2018_1_03.pdf. Acesso em: 15/04/2026.

MINNICK, Chris. *Professional React v19.0.0 course*. 2024. Disponível em: <https://pt.scribd.com/document/860588973/Professional-React-v19-0-0-course>. Acesso em: 15/04/2025.

NUNES, Sergio, MOURA, Ricardo. Programação em Banco de Dados. Londrina: S.A, 2018. 218 páginas. Disponível em: https://cm-kls-content.s3.amazonaws.com/201802/INTERATIVAS_2_0/PROGRAMACAO_EM_BANCO_DE_DADOS/U1/LIVRO_UNICO.pdf. Acesso em: 15/04/2026.

PEREIRA, Caio. Node.js, aplicações web real-time com Node.js. São Paulo: Casa do Código, 2014. 114 páginas. Disponível:

[https://www.kufunda.net/publicdocs/Node.js%20-%20Aplica%C3%A7%C3%B5es%20web%20real-time%20com%20Node.js%20\(Caio%20Ribeiro%20Pereira\).pdf](https://www.kufunda.net/publicdocs/Node.js%20-%20Aplica%C3%A7%C3%B5es%20web%20real-time%20com%20Node.js%20(Caio%20Ribeiro%20Pereira).pdf). Acesso em: 15/04/2026.

WITWER, Michael. Dungeons and Dragons: o império da imaginação: A história de Gary Gygax, Editora LeYa: São Paulo, 2016. Disponível em: http://img.travessa.com.br/capitulo/LEYA/DUNGEONS_DRAGONS_O_IMPERIO_DA_IMAGINACAO-9788544104194.pdf. Acesso: 27/11/2025.