

Universidade de São Paulo
Instituto de Ciências Matemáticas e da Computação
Departamento de Ciências da Computação

**Design acessível de um jogo para apoiar pessoas
com dificuldade intelectual na realização de
atividades de rotina**

Pedro Afonso Fazio Michalichem

São Carlos - SP

**Design acessível de um jogo para apoiar pessoas
com dificuldade intelectual na realização de
atividades de rotina**

Pedro Afonso Fazio Michalichem

Orientadora: Kamila Rios da Hora Rodrigues

Trabalho de Conclusão de Curso
apresentado ao Curso de Graduação
de Sistemas de Informação da Universidade
de São Paulo como requisito
para a obtenção do título de Bacharel em Sistemas de Informação. Orientadora:
Prof. Kamila Rios da Hora Rodrigues

Área de Concentração: Engenharia
de Software e Interação de Humano-
Computador

USP - São Carlos
11/06/2023

DEDICATÓRIA

Dedico esse projeto primeiramente a minha família, pelo amor incondicional, apoio e incentivo ao longo de toda a minha jornada acadêmica. Sem vocês, nada disso seria possível. Aos meus professores, pela orientação precisa, paciência e valiosas contribuições para o desenvolvimento deste trabalho. Aos meus amigos que sempre estiveram ao meu lado, me apoiando e encorajando em todas as etapas desta jornada. À Universidade de São Paulo, pelo ambiente propício à pesquisa e pelos recursos disponibilizados, que foram fundamentais para a realização deste trabalho. Por fim, a todos que, de alguma forma, contribuíram para a conclusão deste projeto de conclusão de curso, meu sincero agradecimento.

AGRADECIMENTOS

Os agradecimentos principais são direcionados a professora Kamila Rios da Hora Rodrigues que me incentivou a pesquisar e desenvolver uma aplicação para auxiliar pessoas com dificuldades, e também por ter me apoiado no projeto de ensino a uso de smartphones aos idosos, sendo estes estudos precursores e motivadores ao desenvolvimento desta monografia. Sem esse apoio e acompanhamento este projeto não teria sido possível.

RESUMO

FAZIO, P. A. M. **Design acessível de um jogo para apoiar pessoas com dificuldade intelectual na realização de atividades de rotina.** 2023. 55f. Monografia (Graduação - Instituição de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP

Esta monografia descreve o desenvolvimento de um jogo para auxiliar pessoas com dificuldade intelectual a lidar com suas atividades rotineiras. O jogo de rotina permite que o jogador possa cumprir tarefas estipuladas por ele mesmo, e ao cumprir essas tarefas, o jogador recebe moedas virtuais, que poderão ser gastas comprando roupas para o personagem virtual. O uso de elementos de gamificação auxiliam a manter o jogador motivado a cumprir suas tarefas e manter hábitos saudáveis em sua rotina. O *framework React.js* foi empregado para o desenvolvimento *Front End*, e *Json-server* para o desenvolvimento *Back End*. O protótipo de alta fidelidade previamente projetado foi criado utilizando a ferramenta Figma. Recursos de acessibilidade foram empregados para tornar o jogo mais adequado para diferentes perfis de usuários.

Palavras-chave: Gamificação, Jogos, Design, Dificuldade Intelectual, Acessibilidade.

ABSTRACT

FAZIO, P. A. M. **Design acessível de um jogo para apoiar pessoas com dificuldade intelectual na realização de atividades de rotina.** 2023. 55f. Monografia (Graduação - Instituição de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP

This monograph describes the development of a game to help people with intellectual difficulties to deal with their routine activities. The routine game allows the player to fulfill tasks stipulated by himself/herself, and by fulfilling these tasks, the player receives virtual coins, which can be spent buying clothes for the virtual character. The use of gamification elements helps keep the player motivated to fulfill their tasks and maintain healthy habits in their routine. *framework React.js* was employed for *Front End* development, and *Json-server* for *Back End* development. The previously designed high-fidelity prototype was created using the Figma tool. Accessibility features were employed to make the game more suitable for different user profiles.

Palavras-chave: Gamification, Games, Design, Intellectual Difficulty, Accessibility.

Sumário

1	Introdução	13
1.1	Objetivo Geral	14
1.2	Objetivos Específicos	14
1.3	Organização do Documento	14
2	Referencial Teórico	15
2.1	Contextualização dos jogos	15
2.2	Associação dos jogos ao termo gamificação	17
2.3	Acessibilidade no geral e na tecnologia	19
2.4	Jogos similares encontrados	20
2.4.1	Habitica	20
2.4.2	Uno edição com mais Acessibilidade	24
2.5	Diretrizes de Acessibilidade WCAG	25
2.5.1	Diretrizes WCAG utilizadas no projeto	26
2.6	Pessoas com dificuldade intelectual	27
2.6.1	Jogos para pessoas com dificuldade intelectual	28
2.7	Considerações Finais	29
3	Metodologias Empregadas	29
3.1	Etapa de Design	30
3.1.1	Considerações na criação do <i>Design</i>	30
3.1.2	Ferramentas utilizadas	31
3.2	Etapa de desenvolvimento	31
3.2.1	Desenvolvimento Incremental	31
3.2.2	Clean Code	32
3.2.3	Ferramentas Utilizadas	32
3.3	Etapa de Avaliação	33
4	Jogo Acessível	33
4.1	Etapa de coleta de requisitos	33
4.2	Prototipação de alta fidelidade	34
4.2.1	Criação do <i>mock up</i> no figma	34
4.2.2	Prototipação - Páginas Principais	35
4.2.3	Páginas de jogabilidade	37
4.2.4	Páginas de Produtos	43
4.2.5	Área de usuários	46
4.3	Implementação do jogo: Front-End	50
4.3.1	Descrição da aplicação	50

4.4	Setores da aplicação	52
4.4.1	Login	52
4.4.2	Rotina	53
4.4.3	Loja	53
4.4.4	Perfil	53
4.4.5	Inventário	53
4.4.6	Sobre	53
4.5	Implementação do jogo: Back-End	54
4.5.1	Descrição da aplicação	54
4.5.2	API Rest	55
4.6	Dados guardados no backend	55
4.7	Acessando o projeto	57
5	Conclusão	58
5.1	Contribuições do Trabalho	58
5.2	Dificuldades e Limitações do Trabalho	58
5.3	Considerações Finais	59

Lista de Abreviaturas/Siglas

VS Code: Visual Studio Code.

HTML: HyperText Markup Language.

CSS: Cascading Style Sheet.

NES: Nintendo Entertainment System.

NPM: Node Package Manager.

JSX: JavaScript XML.

.SVG: Scalable Vector Graphics

DOM: Document Object Model

SSR: Server-Side Rendering

XP: Experience

WCAG: Web Content Accessibility Guidelines

Lista de Gráficos

Figura 2 Gráfico 1 - Núcleo do Conhecimento

Figura 3 Gráfico 2 - Núcleo do Conhecimento

Lista de Figuras

Figura 1 Tennis For Two.

Figura 2 Gráfico 1 - Núcleo do Conhecimento

Figura 3 Gráfico 2 - Núcleo do Conhecimento

Figura 9 Projeto do jogo no Figma - Página Inicial.

Figura 10 Projeto do jogo no Figma - Sobre nós.

Figura 11 Projeto do jogo no Figma - Rotina Tarefas.

Figura 12 Projeto do jogo no Figma - Rotina Diárias.

Figura 13 Projeto do jogo no Figma - Rotina Hábitos.

Figura 14 Projeto do jogo no Figma - Rotina Contas.

Figura 15 Projeto do jogo no Figma - Adicionar Tarefas, Diárias ou Hábitos.

Figura 16 Projeto do jogo no Figma - Adicionar Contas.

Figura 17 Projeto do jogo no Figma - Loja.

Figura 18 Projeto do jogo no Figma - Carrinho.

Figura 19 Projeto do jogo no Figma - Perfil.

Figura 20 Projeto do jogo no Figma - Inventário.

Figura 21 Projeto do jogo no Figma - Criar Conta.

Figura 22 Projeto do jogo no Figma - Login.

1 Introdução

A necessidade de jogos para *gamificar* a rotina surge principalmente da constatação de que muitas pessoas têm dificuldade em manter a motivação e o engajamento em tarefas rotineiras. Isso pode se dar por diversos fatores, mas é acentuado em pessoas com alguma dificuldade intelectual. Atividades como fazer exercícios físicos, estudar ou trabalhar todos os dias, pode ser cansativo e monótono, e muitas vezes essas pessoas precisam de uma atenção maior, pois acabam se desmotivando.

Ao aplicar elementos de gamificação em soluções como jogos, o uso de recompensas, desafios e pontuações, associadas a tarefas cotidianas, permite tornar as atividades mais interessantes e motivadoras, o que pode ajudar tais pessoas a manter a consistência e a disciplina para realizá-las. Além disso, os jogos também podem criar um senso de competição e de colaboração, incentivando as pessoas a se esforçarem mais para alcançar seus objetivos.

Sobre os jogos de rotina, também conhecidos como *gamificação* da rotina, eles são um tipo de abordagem inovadora que envolve a aplicação de recompensas, desafios e pontuações para incentivar a realização de tarefas diárias. A ideia da *gamificação* da rotina é transformar tarefas que podem ser monótonas e/ou cansativas em atividades mais agradáveis e interessantes, tornando a rotina não mais como um fardo e sim como algo divertido.

A aplicação criada no contexto deste trabalho, chamada *Task Complete*, foi projetada como uma alternativa para ajudar os jogadores a se tornarem mais produtivos, bem como estabelecerem hábitos saudáveis e se envolverem de forma mais divertida e estimulante com suas tarefas diárias e metas pessoais.

O jogo pode auxiliar não só pessoas com dificuldade intelectual, mas também pessoas que sentem a necessidade de melhorar a sua qualidade de vida no geral.

A construção do jogo levou em consideração diretrizes de acessibilidade, que são recomendações e padrões para tornar sites e aplicativos Web mais acessíveis. Um exemplo de recurso adotado neste trabalho, é a utilização elementos de marcação *HTML*, como o uso da tag ”*alt*” nas imagens, assim tendo descrições alternativas para imagens. O jogo também permite a navegação por teclado, tem contraste adequado entre cores e uma alternativa de cores para pessoas

daltônicas.

1.1 Objetivo Geral

Esta pesquisa teve como principal objetivo o design e desenvolvimento de um jogo para dispositivos Web contendo recursos de acessibilidade. O jogo visa melhorar a motivação, engajamento e consistência nas tarefas diárias, especialmente para pessoas com dificuldade em manter a disciplina e o interesse em atividades rotineiras. A gamificação da rotina utiliza elementos de jogos, como recompensas, desafios e pontuações, para tornar essas tarefas mais interessantes e divertidas.

1.2 Objetivos Específicos

A seguir estão elencados os objetivos específicos para que o objetivo geral fosse alcançado:

- Identificar diretrizes de acessibilidade para jogos Web, a partir de propostas em trabalhos e pesquisas acadêmicas e também em organizações que buscam garantir o direito da acessibilidade para pessoas com algum tipo de deficiência a esses dispositivos;
- Elaborar uma prototipação de alta fidelidade do jogo para validar os requisitos coletados;
- Aplicar os conhecimentos com o *framework React.js* para desenvolver o *Front End* da aplicação;
- Conhecer uma estrutura para acomodar os dados apresentados para o *Back End*.

1.3 Organização do Documento

Esta monografia está organizada da seguinte maneira: o Capítulo 2 descreve o Referencial Teórico utilizado para auxiliar no desenvolvimento da monografia e também do projeto, tratando da contextualização da *gamificação*, acessibilidade e outros tópicos; o Capítulo 3 descreve os métodos usados no *design* e prototipação, acessibilidade e desenvolvimento; o Capítulo 4 são as partes funcionais do projeto, em que são detalhadas as tecnologias usadas, *frameworks*,

até mesmo as bibliotecas que serviram para o desenvolvimento funcional do projeto; por fim, o Capítulo 5 trata sobre a conclusão, dificuldades e limitações, e também das considerações finais do trabalho.

2 Referencial Teórico

2.1 Contextualização dos jogos

A história dos jogos é tão antiga quanto a história da humanidade. Desde tempos pré-históricos, os seres humanos têm criado jogos e brincadeiras como forma de entretenimento, aprendizado e socialização, e isso foi evoluindo até mesmo os dias de hoje.

Tratando dos jogos mais antigos de que se tem registro, eles acabam incluindo jogos de tabuleiro e jogos de azar, como por exemplo o *Senet*, que por sua vez, é um jogo egípcio com mais de 5.000 anos, e também o Jogo *Real de Ur*, sendo ele um jogo mesopotâmico com cerca de 4.500 anos. Esses dois jogos eram usados com o objetivo de ensinar habilidades estratégicas e matemáticas para os usuários que os jogassem, bem como para passar o tempo e se divertir.

Ao longo dos séculos, os jogos evoluíram e se diversificaram, passando a incluir diversos tipos de jogos, sendo eles: jogos de cartas, jogos de xadrez, jogos de damas, jogos de corrida e outros. Com o advento da tecnologia, surgiram os jogos eletrônicos, que se tornaram uma forma cada vez mais popular de entretenimento.

Os primeiros jogos eletrônicos surgiram na década de 1950, com jogos simples como *"Tennis for Two"*, criado pelo físico americano William Higinbotham, conforme ilustrado na Figura 1.

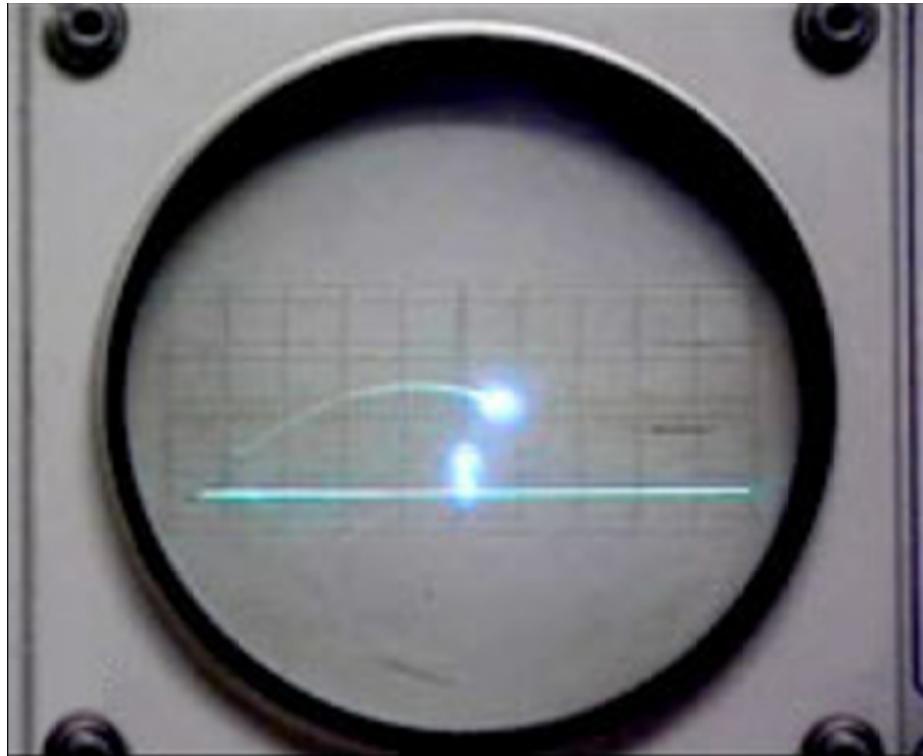


Figura 1: Tennis For Two (Fonte: Game Hall Network 2015);

Na década de 1970, surgiram os primeiros jogos de arcade, como "Pong" e "Space Invaders", que rapidamente se tornaram populares em todo o mundo.

Na década de 1980, surgiram os primeiros consoles de jogos domésticos, como o Atari 2600 e o *Nintendo Entertainment System (NES)*, que abriram caminho para a popularização dos jogos eletrônicos em casa. Desde então, a indústria dos jogos eletrônicos tem crescido exponencialmente, se tornando um dos maiores setores do entretenimento em todo o mundo.

Hoje, os jogos são usados de uma outra forma, sendo essa uma forma popular de entretenimento, além de serem usados em áreas como educação, treinamento empresarial, terapia e até mesmo em pesquisas científicas. A história dos jogos é uma prova da capacidade humana de criar, inovar e se divertir, e a evolução dos jogos ao longo do tempo é um reflexo das mudanças culturais, sociais e tecnológicas

que moldam nossa sociedade. (PACHECO, 2013)

2.2 Associação dos jogos ao termo gamificação

Embora a gamificação seja uma abordagem relativamente nova, os jogos sempre foram uma fonte de inspiração para a criação de sistemas de incentivo e motivação. Desde os jogos de tabuleiro antigos até os jogos eletrônicos modernos, os jogos sempre foram projetados para serem desafiadores e motivadores, oferecendo recompensas e *feedback* imediato para manter os jogadores engajados.

A ideia de gamificação, como uma técnica para incentivar o engajamento e a motivação em atividades cotidianas, surgiu no final dos anos 2000, com a popularização das mídias sociais e dos aplicativos móveis. A *gamificação* aplica princípios e mecânicas de jogos em áreas como educação, saúde, trabalho e *marketing*, visando tornar atividades cotidianas mais interessantes e motivadoras.

A *gamificação* é inspirada pela história dos jogos, incorporando elementos como recompensas, desafios, pontuações e *feedback* imediato em atividades não lúdicas. Por exemplo, um aplicativo de exercícios pode oferecer recompensas por completar determinadas metas, ou um programa de treinamento empresarial pode usar jogos para ensinar habilidades de liderança e colaboração.

A *gamificação* pode ser aplicada em várias áreas diferentes, inclusive na educação, desde a educação infantil até o ensino superior, e em diversas disciplinas, como matemática, ciências, línguas e história. Um exemplo é um experimento chamado “Ensino e educação: uso da *gamificação* na matemática”, que foi divulgado pela Revista Científica Multidisciplinar Núcleo do Conhecimento pelo autor Adriano Carlos Moraes Rosa. A imagem a seguir ilustra os resultados obtidos pela pesquisa.

Figura 2 - Nota do 5º Ano que Não Participou da Gamificação

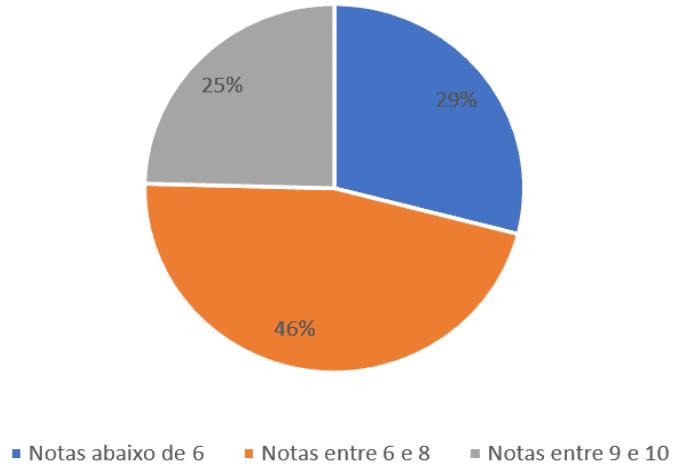


Figura 2: Fonte: Núcleo do Conhecimento (gamificação na matemática.)

No gráfico da figura acima é mostrado como os alunos que não participaram da dinâmica de *gamificação* do ensino demonstraram resultados não tão animadores.

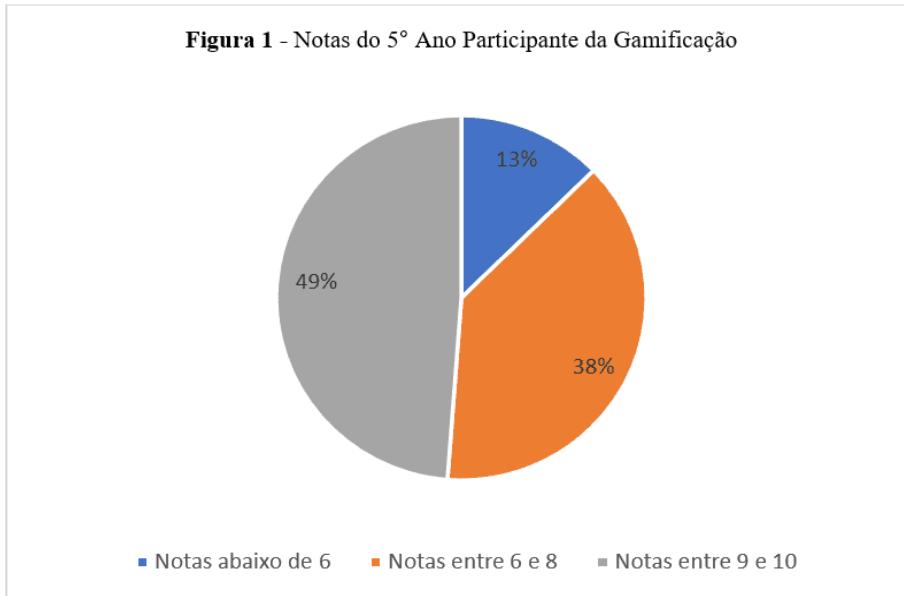


Figura 3: Fonte: Núcleo do Conhecimento (gamificação na matemática)

Contudo, nesse outro é demonstrado resultados melhores ao avaliar as crianças que tiverem acesso ao ensino gamificado, demonstrando assim, uma efetividade na *gamificação* do ensino.

Em resumo, a história dos jogos fornece a base conceitual e prática para a gamificação, demonstrando como os jogos são capazes de motivar e engajar as pessoas em tarefas desafiadoras e aparentemente entediantes. A gamificação é uma aplicação prática dessa ideia, que busca aproveitar esses princípios em contextos não lúdicos para promover engajamento, motivação e aprendizagem. (ROSA, 2021)

2.3 Acessibilidade no geral e na tecnologia

Aprofundando nas questões de acessibilidade, esse é um termo que se refere à capacidade das pessoas, independentemente de suas habilidades ou limitações, terem acesso igualitário e sem barreiras a todos os recursos, serviços e espaços da sociedade. Isso inclui o acesso a produtos, tecnologias, edifícios, informações, transporte, comunicação e outros aspectos da vida cotidiana.

Mais especificamente na tecnologia, ela é considerada a inclusão de pessoas com deficiência ou limitações para que elas tenham acesso aos recursos e serviços de tecnologia, bem como usá-los de maneira efetiva. Isso inclui o acesso a computadores, dispositivos móveis, *software* e aplicativos, entre outros.

É importante essa inclusão porque muitas pessoas com deficiência ou limitações dependem de tecnologias assistivas para realizar tarefas cotidianas, trabalhar, se comunicar e se envolver em atividades sociais. A acessibilidade na tecnologia pode incluir recursos como leitores de tela, teclados especiais, interfaces de usuário personalizáveis, legendas e outras tecnologias que tornam as informações e os serviços digitais mais acessíveis.

Por fim, a acessibilidade também pode ajudar a promover a inclusão social e a igualdade de oportunidades para pessoas com deficiência ou limitações. A tecnologia pode ajudar a remover barreiras e obstáculos que podem impedir ou dificultar a participação plena das pessoas na sociedade. É por isso que muitos governos e organizações internacionais desenvolvem normas e diretrizes para garantir a acessibilidade na tecnologia e incentivam a adoção de práticas de *design* acessível (CARDOSO e RAIMUNDO, 2017).

2.4 Jogos similares encontrados

Para ter uma melhor base na construção da solução, foi preciso estudar alguns jogos já existentes, e a partir deles conseguir utilizar as suas qualidades para o projeto *Task Complete*. Para exemplificar melhor essa seção, os jogos serão listados e descritos em uma lista.

2.4.1 Habitica

Habitica é um aplicativo de produtividade e gerenciamento de tarefas que usa gamificação para motivar os usuários a alcançarem seus objetivos e criar hábitos saudáveis. O aplicativo foi originalmente lançado em 2013 e desde então se tornou popular entre pessoas que buscam aumentar sua produtividade e desenvolver rotinas positivas.

No Habitica, as tarefas são elementos centrais para ajudá-lo a organizar suas atividades diárias, estabelecer metas e desenvolver hábitos saudáveis. As tarefas podem ser divididas em três categorias principais: tarefas diárias, hábitos e tarefas únicas, onde é possível

colocar as suas tarefas na lista e em seguida completa-las em troca de recompensas.



Figura 4: Fonte: Habitica - Jogo Web.



Figura 5: Fonte: Habitica - Jogo Web.

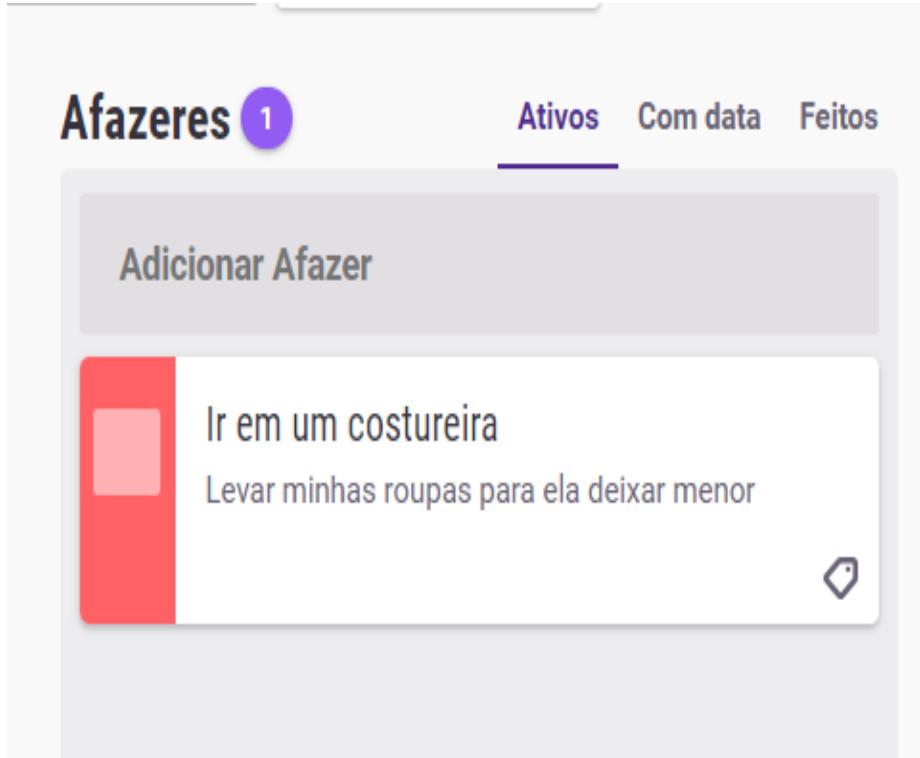


Figura 6: Fonte: Habitica - Jogo Web.

Ao criar a conta no Habitica, o usuário tem a oportunidade de personalizar seu personagem, escolhendo seu gênero, aparência, penteado, roupas e acessórios. Essas opções de personalização permitem que ele crie um avatar único que o representa no jogo.

Conforme avança e realiza tarefas, desenvolve hábitos e alcança metas, o usuário ganha pontos de experiência e moedas virtuais. Esses recursos podem ser usados para adquirir itens no mercado do Habitica, que inclui uma variedade de armas, armaduras, animais de estimação, montarias e outros acessórios. Cada item tem seus próprios atributos e benefícios, que podem ajudar a melhorar seu personagem e desempenho no jogo.

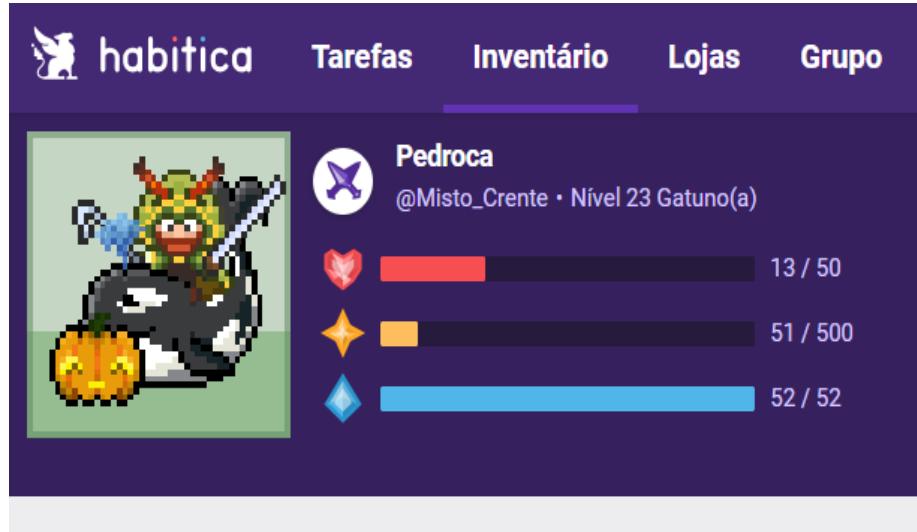


Figura 7: Fonte: Habitica - Jogo Web)

No projeto foi utilizado a ideia do jogo de se ter uma gamificação da rotina, trazendo elementos de recompensa para o usuário ao cumprir as suas tarefas. Desta forma, esse jogo foi de grande importância para desenvolver o *Task Complete*.

2.4.2 Uno edição com mais Acessibilidade

Uno foi o outro jogo que o Task Complete se baseou para alcançar mais acessibilidade aos seus usuários. Mais especificamente a edição especial do jogo de cartas Uno que é voltada para pessoas com daltonismo, no qual as cartas são marcadas com o selo do código de cores universal para daltonismo, conhecido como *ColorADD*.

Ele funciona da seguinte forma: Os números das cartas são acompanhados de um sinal próximo a eles. Cada cor: vermelho, azul e amarelo possuem um símbolo único, e quando combinados, esses símbolos representam uma nova cor. Por exemplo, ao combinar o traço do cartão amarelo com o triângulo do azul, obtemos um símbolo que representa a cor verde no cartão. Dessa forma, as cores não são apenas identificadas pelos sinais, mas também são reconstruídas através de suas combinações. Isso auxilia uma pessoa daltônica a associar e memorizar as cores corretamente (GUILMARÃES CRUZ, 2017).

Na Figura 8 é possível visualizar como funciona visualmente esse sistema para daltônicos.

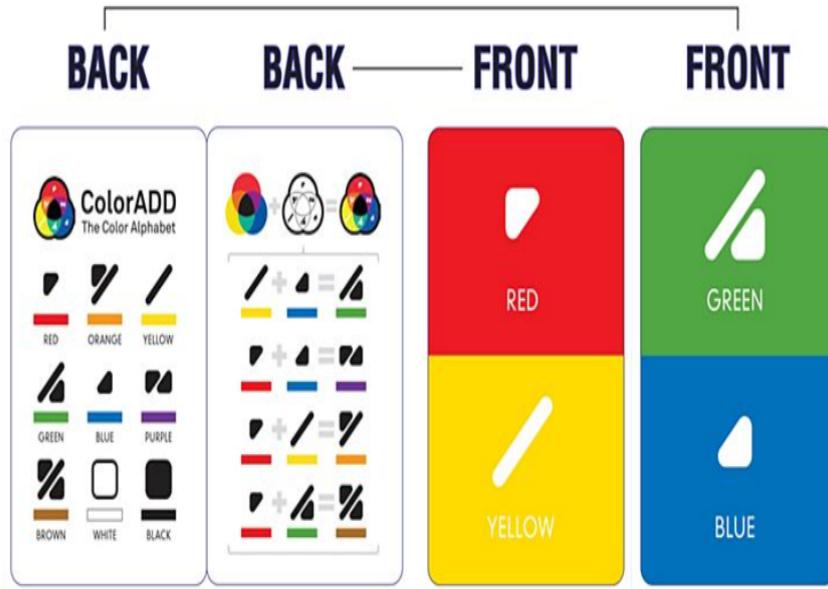


Figura 8: Uno - ColorADD

2.5 Diretrizes de Acessibilidade WCAG

WCAG significa "Web Content Accessibility Guidelines" (Diretrizes de Acessibilidade para Conteúdo Web, em português). É um conjunto de diretrizes internacionais desenvolvido pelo World Wide Web Consortium (W3C) para ajudar a tornar o conteúdo da web mais acessível para pessoas com deficiência.

As diretrizes WCAG definem critérios e recomendações técnicas que os desenvolvedores da web devem seguir para garantir que seu conteúdo seja percebido, operável e compreensível por uma ampla variedade de usuários, incluindo aqueles com deficiências visuais,

auditivas, cognitivas ou motoras.

Também é importante relatar que o WCAG define três níveis de conformidade: A, AA e AAA. Esses níveis são destinados a indicar o grau de acessibilidade de um site ou aplicativo de acordo com as diretrizes WCAG (WCAG, 2008).

- **Nível A:** O Nível A contém os requisitos mais básicos de acessibilidade. Atender a esses requisitos é considerado o ponto de partida para tornar o conteúdo da web acessível. O cumprimento do Nível A é crucial para fornecer uma acessibilidade mínima. Ao atender aos critérios do Nível A, é abordado algumas das principais barreiras para pessoas com deficiência.
- **Nível AA:** O Nível AA inclui critérios adicionais para acessibilidade. Cumprir com o Nível AA é considerado um nível intermediário de conformidade. Os critérios do Nível AA abordam uma gama mais ampla de barreiras de acessibilidade e garantem que uma maior variedade de pessoas com deficiência possa acessar e interagir com o conteúdo da web.
- **Nível AAA:** O Nível AAA é o mais alto nível de conformidade e inclui os critérios mais rigorosos para acessibilidade. Cumprir com o Nível AAA é um objetivo mais desafiador e nem sempre é viável para todos os tipos de conteúdo. O cumprimento do Nível AAA garante uma experiência de usuário altamente acessível, abordando uma ampla gama de necessidades de pessoas com deficiência.

2.5.1 Diretrizes WCAG utilizadas no projeto

No projeto foram utilizadas algumas das diretrizes WCAG para tornar acessível a experiência de usuários que possuam algum tipo de deficiência. As diretrizes que constam no projeto são de maior parte do Nível A, e algumas do Nível AA, sendo elas as listadas abaixo (WCAG, 2008):

- **Conteúdo não textual [A]:** Qualquer conteúdo ”não textual” e relevante para compreensão da informação, deve trazer uma descrição alternativa em texto (visível ou não) para identificar o conteúdo (inclusive captcha, por exemplo);

- **Utilização de cores [A]:** Cores não devem ser utilizadas como única maneira de transmitir conteúdo ou distinguir elementos visuais;
- **Sequência com significado [A]:** Seja qual for o método de interação, a apresentação das informações na tela sempre deverá ter uma sequência lógica;
Dica: uma mensagem de erro em um formulário deve trazer um ícone de alerta, uma mensagem informando como corrigir o erro e também uma cor destacando a informação (e não apenas a mudança de cor);
- **Página com título [A]:** Todas as telas devem ter um título principal e que descreva claramente a sua finalidade;
- **Ordem do foco [A]:** A interação por elementos focáveis na tela sempre deverá ser sequencial e lógica de acordo com o conteúdo apresentado;
- **Orientação [AA]:** Nenhuma funcionalidade deve depender de uma determinada orientação de tela (exemplo: virar o celular na horizontal), a não ser que seja imprescindível para execução da função.

2.6 Pessoas com dificuldade intelectual

Pessoas com dificuldade intelectual são indivíduos que apresentam limitações no funcionamento intelectual e nas habilidades adaptativas. Essas limitações podem afetar o seu aprendizado, comunicação, compreensão de conceitos complexos, habilidades sociais e o desempenho de atividades diárias, atrapalhando por sua vez também a sua rotina.

É importante compreender que a dificuldade intelectual não define a totalidade da pessoa. Cada indivíduo possui suas próprias capacidades, talentos e potenciais, independentemente de suas limitações intelectuais. É essencial valorizar e respeitar a singularidade de cada pessoa com dificuldade intelectual, reconhecendo suas habilidades e oferecendo oportunidades para o seu pleno desenvolvimento.

Pessoas com dificuldade intelectual podem enfrentar desafios em diferentes aspectos da vida, como a educação, o emprego, a inde-

pendência e as relações sociais. No entanto, com o suporte adequado, elas têm a capacidade de aprender, crescer e se envolver ativamente na sociedade.

A inclusão e a acessibilidade são fundamentais para garantir a participação plena das pessoas com dificuldade intelectual na comunidade. Isso envolve a remoção de barreiras físicas, a adaptação de materiais de aprendizagem e a criação de ambientes inclusivos que promovam a igualdade de oportunidades. Bem como garantir a inclusão dessas pessoas a acesso a tecnologias assistivas e jogos interativos que podem auxiliar em suas atividades rotineiras, de estudo, entre outras atividades.

Finalmente, é importante promover a conscientização e a compreensão sobre a dificuldade intelectual, desmistificando estereótipos e preconceitos. Ao oferecer um ambiente acolhedor e apoio adequado, podemos criar oportunidades para que essas pessoas desenvolvam suas habilidades, alcancem suas metas e se tornem membros ativos e valorizados da sociedade (DA ROCHA, 2008).

2.6.1 Jogos para pessoas com dificuldade intelectual

Jogos são uma forma maravilhosa de entretenimento e aprendizado, e também podem ser acessíveis e inclusivos para pessoas com dificuldade intelectual. Esses jogos oferecem oportunidades valiosas para estimular habilidades cognitivas, sociais e emocionais, promovendo o desenvolvimento e a diversão para todos.

Ao projetar jogos para pessoas com dificuldade intelectual, é importante considerar a simplicidade, a clareza e a adaptabilidade. Os jogos devem ser intuitivos e fáceis de entender, com instruções claras e recursos visuais ou sonoros que auxiliem na compreensão.

Tratando sobre jogos sérios, eles podem ajudar muito pessoas com algum tipo de dificuldade. Nos últimos anos, tem havido uma crescente compreensão do papel das tecnologias computacionais e de smartphones no apoio ao ensino de habilidades funcionais, sociais e comportamentais em indivíduos com dificuldade intelectual. Diversos estudos foram realizados nessa área, revelando que intervenções baseadas em tecnologia podem ser altamente eficazes nesse contexto.

Em pesquisas recentes, foram apresentadas abordagens inovadoras que exploram o potencial das tecnologias para auxiliar pessoas

no desenvolvimento e aprimoramento de suas competências emocionais e sociais. Essas soluções inovadoras aproveitam as vantagens das novas tecnologias para criar intervenções personalizadas e interativas.

A demanda por novas pesquisas nesse campo está em constante crescimento, uma vez que há um interesse crescente em explorar soluções baseadas em tecnologia para promover o desenvolvimento de habilidades emocionais. Essas soluções podem incluir o uso de aplicativos móveis, jogos computadorizados ou outras ferramentas tecnológicas projetadas especificamente para atender às necessidades individuais de pessoas com dificuldades.

Essas pesquisas visam fornecer recursos e estratégias inovadoras que possam complementar e enriquecer as abordagens tradicionais de intervenção, oferecendo novas formas de aprendizado e desenvolvimento emocional para esses indivíduos. O objetivo é capacitar essas pessoas a alcançar um maior nível de independência, bem-estar emocional e participação social. (DANTAS, MELO, NEVES, MILESSI, NASCIMENTO, 2019)

2.7 Considerações Finais

Essa seção de referencial teórico desempenhou um papel crucial no projeto, pois reflete vários aspectos de importância na construção do Task Complete. Esse conteúdo auxiliou na contextualização e fundamentação do trabalho, fornecendo um embasamento teórico para compreender as atividades realizadas. Ao apresentar principais teorias, conceitos e ideias existentes sobre o tema, foi possível situar o trabalho em um corpo de conhecimento.

3 Metodologias Empregadas

Neste capítulo serão apresentados os métodos de design utilizado para a construção do *Mock up* do jogo, as metodologias usadas para idealizar a aplicação, fazendo com que ela tivesse recursos de acessibilidade para públicos com deficiências ao utilizar tecnologias Web, e também métricas utilizadas para o desenvolvimento do sistema Web em que é abrigado o jogo apresentado para o projeto.

3.1 Etapa de Design

Nesta etapa, uma pesquisa sobre o público alvo foi realizada para entender melhor as demandas do mesmo. Essas pesquisas auxiliaram na construção do protótipo. Erros nessa etapa são mais fáceis de serem corrigidos, por ainda não estarem na aplicação final. Finalmente, para um melhor aproveitamento, nesta etapa foi feita uma abordagem baseada em tópicos importantes para a construção do design do projeto.

3.1.1 Considerações na criação do *Design*

Ao elaborar o protótipo é preciso considerar vários aspectos. Aqui estão algumas coisas que foram levadas em conta:

- **Objetivo do protótipo:** Foi definido claramente o objetivo do protótipo, assim ajudando a determinar quais interações e funcionalidades devem ser incluídas, bem como o nível de detalhe necessário no *design*;
- **Público-alvo:** Foi considerado quem será o público alvo do protótipo, pensando nas necessidades e expectativas dos usuários para atender os seus requisitos. Sendo os usuários principais da aplicação, pessoas com dificuldade intelectual que serão ajudadas em suas atividades rotineiras com essa solução;
- **Fluxo de navegação:** Foi planejado o fluxo de navegação do protótipo. Determinando quais telas serão necessárias e como os usuários irão interagir com elas. Utilizando-se de recursos como: links e transições, para criar um fluxo de navegação realista;
- **Consistência visual:** Foi mantido a consistência visual em todo o protótipo. Usando estilos de texto, cores e componentes reutilizáveis para garantir uma aparência coesa em todas as telas. Desta forma, ajudando os usuários a entenderem e navegarem pelo protótipo de forma mais fácil.

Esses aspectos foram considerados ao fazer uma prototipação na ferramenta de prototipação *Figma*¹, para criar um protótipo eficaz, centrado no usuário e que atendesse às necessidades do projeto.

¹<https://www.figma.com/>

3.1.2 Ferramentas utilizadas

Como ferramenta de desenvolvimento do *Mock Up* foi utilizado o *Figma*, que é um editor gráfico de prototipagem e *Photoshop CS6*, que é um editor de imagens.

Em relação aos recursos usados para a criação dessas páginas: Todas as imagens e derivados para a prototipação foram retiradas do site <https://www.pexels.com/pt-br/>, e todos os ícones .svg foram retirados do site <https://www.flaticon.com/>. Também foram utilizadas algumas imagens pessoais, sendo algumas criadas inteiramente no *photoshop* especialmente para ser utilizada no *Task Complete*.

3.2 Etapa de desenvolvimento

Nesta etapa são conduzidas as atividades de implementação, onde é focado no ciclo de vida do projeto em que são determinadas atividades para atingir os objetivos declarados pelo design. Essas etapas são planejadas seguindo uma sequência de acordo com o desenvolvimento incremental para cada etapa do projeto. Como boas práticas de desenvolvimento, procedimentos de *clean code* e revisão de código foram adotadas.

3.2.1 Desenvolvimento Incremental

O desenvolvimento incremental é uma abordagem de desenvolvimento de software em que o projeto é dividido em incrementos ou partes menores, que são desenvolvidos e entregues de forma iterativa e incremental ao longo do tempo. Cada incremento representa uma funcionalidade ou conjunto de funcionalidades completas que agregam valor ao produto final.

Em vez de desenvolver o *software* completo de uma só vez, o desenvolvimento incremental permite que os desenvolvedores construam o *software* em pequenas partes, avaliando e ajustando continuamente o projeto com base no *feedback* dos usuários e nas necessidades em constante evolução. Cada incremento passa por todas as fases de desenvolvimento, desde a análise de requisitos até o teste e a entrega, resultando em um produto funcional a cada incremento.

3.2.2 Clean Code

Clean code, ou código limpo, refere-se a um estilo de programação que enfatiza a legibilidade, simplicidade e manutenção do código-fonte. É uma abordagem que visa produzir código claro, organizado e de fácil compreensão, tanto para o programador que o escreve quanto para os outros que irão lê-lo e mantê-lo no futuro.

Princípios e boas práticas são seguidos para alcançar o *clean code*. Alguns dos princípios mais comuns incluem:

- **Nomes significativos:** Os nomes de variáveis, funções, classes e métodos devem ser descritivos e significativos, facilitando a compreensão do propósito e funcionalidade do código;
- **Funções e métodos concisos:** As funções e métodos devem ser pequenos e focados em realizar uma única tarefa. Isso torna o código mais legível e mais fácil de entender;
- **Evitar duplicação de código:** A duplicação de código deve ser evitada, pois leva a uma maior dificuldade de manutenção. O código duplicado deve ser reutilizado por meio da criação de funções ou métodos compartilhados;
- **Comentários claros:** O código deve ser autoexplicativo na medida do possível, mas quando necessário, comentários devem ser utilizados para explicar partes complexas ou não triviais do código;
- **Estrutura e organização:** O código deve ser organizado em uma estrutura lógica, com identação adequada e separação de seções para facilitar a leitura e a compreensão.

A prática do *clean code* traz vários benefícios, como melhor compreensão do código, facilitação da colaboração entre desenvolvedores, redução de erros e maior facilidade de manutenção e evolução do software ao longo do tempo.

3.2.3 Ferramentas Utilizadas

Para o desenvolvimento, a ferramenta de edição principal escolhida foi o *Visual Studio Code*, e o restante das ferramentas utilizadas estão na lista abaixo:

- Utilização das ferramentas de desenvolvimento web: *JavaScript/HTML/CSS*;
- O projeto todo foi desenvolvido com auxílio do *framework React.js*²;
- Integração com *Back End* e também base de dados foi feita através de fetch via a biblioteca *Json-Server*.

3.3 Etapa de Avaliação

A etapa de avaliação do projeto será realizada futuramente com usuários. Onde serão realizados testes de usabilidade que envolvem a observação direta dos usuários enquanto eles interagem com o *software*. Os usuários serão solicitados a realizar tarefas específicas e seus comportamentos e *feedbacks* são registrados. Isso irá ajudar a identificar problemas de usabilidade, pontos de confusão e áreas de melhoria.

4 Jogo Acessível

4.1 Etapa de coleta de requisitos

A coleta de requisitos ajuda a entender e definir claramente as necessidades e expectativas dos usuários e stakeholders. Para esse projeto foi utilizado principalmente a coleta de requisitos através de **análise de documentos e fontes existentes**, onde é revisado os documentos, relatórios, manuais e fontes de informação existentes relacionados ao domínio do sistema. Com isso foi possível obter insights sobre processos existentes, requisitos e outras informações relevantes.

Uma das principais fontes analisadas foi uma solução já existente chamada Habitica, que também é um jogo *web* de gamificação de rotina.

Finalmente, o problema identificado foi a dificuldade intelectual do público alvo e a suas dificuldades de estabelecer tarefas rotineiras, então também foi estudados maneiras de tornar essa solução acessível para diversos públicos com deficiências no uso de tecnologias, e devido a isso foi proposto essa solução de jogo acessível.

²<https://react.dev/>

4.2 Prototipação de alta fidelidade

A prototipação de alta fidelidade é um método de criação de protótipos que visa criar uma representação detalhada do design final do produto. Nesse tipo de prototipação, são utilizadas ferramentas e técnicas que permitem criar um protótipo visual e interativo que se assemelha o máximo possível ao produto final. No caso dessa solução foi feita essa prototipagem em foram de Mock Up se utilizando da ferramenta de prototipação Figma.

4.2.1 Criação do *mock up* no figma

Nesta etapa, utilizando a ferramenta de prototipação Figma, foram desenvolvidas 14 páginas para o *mock up*, se utilizando do mesmo design, sendo elas:

- Páginas Principais
 - Página Inicial;
 - Sobre nós;
- Páginas de jogabilidade
 - Rotina (Tarefas);
 - Rotina (Diárias);
 - Rotina (Hábitos);
 - Rotina (Contas);
 - Rotina (Adicionar Tarefas, Diárias, Hábitos);
 - Rotina (Adicionar Contas);
- Produtos
 - Produtos - Roupas;
 - Produtos - Carrinho;
- Área de usuários
 - Usuário - Perfil;
 - Usuário - Inventário
 - Usuário - Criar conta;
 - Usuário - Login;

4.2.2 Prototipação - Páginas Principais

As páginas principais são que fornecem informações fundamentais e são frequentemente visitadas pelos usuários para obter uma visão geral do site e de seus conteúdos.

- Página Inicial;

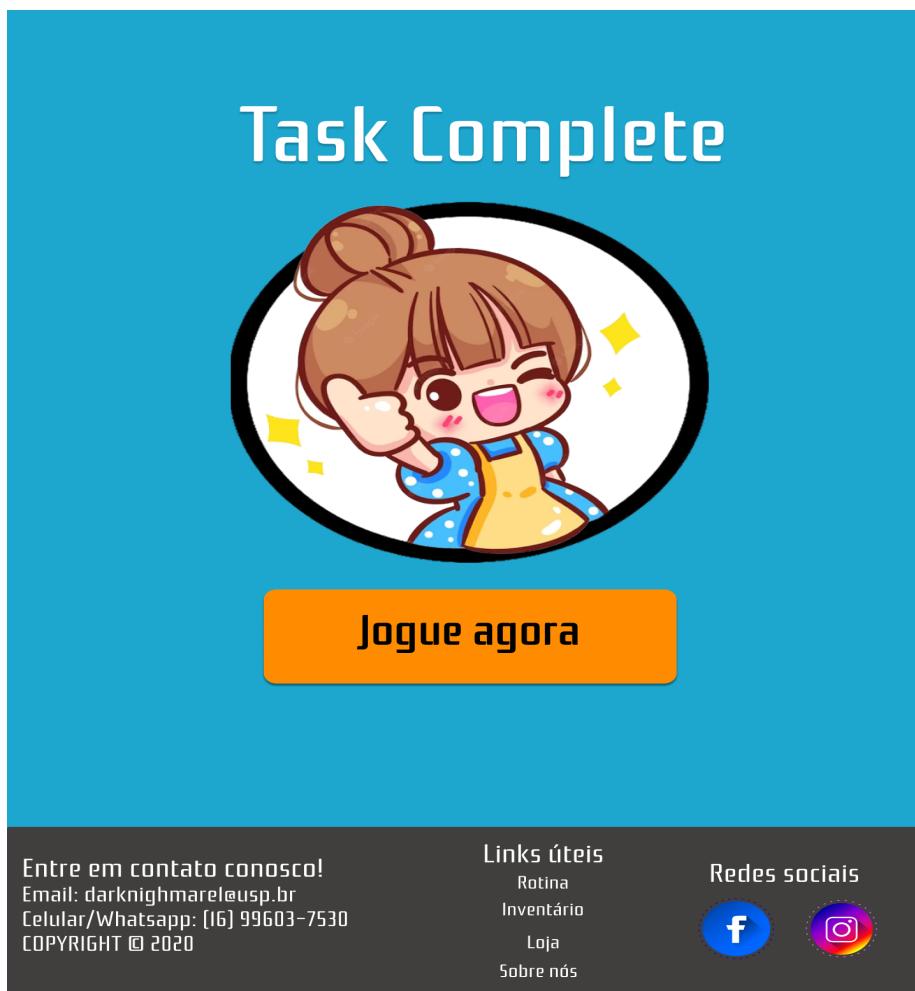


Figura 9: Tela do Figma - Página Inicial

A seção ”Página Inicial” é a apresentação da aplicação, em que apenas apresenta uma tela visual chamativa para convidar o usuário a jogar o *Task Complete*, nome dado ao jogo.

- Sobre nós



Figura 10: Tela do Figma - Sobre nós

A seção ”Sobre” que fornece informações e detalhes relevantes sobre o projeto de graduação apresentado no site. É um lugar em

que os visitantes podem aprender mais sobre a história, objetivos e valores do projeto. Também conta com a equipe chave que foi designada para o desenvolvimento desse projeto. É uma parte importante do site, pois permite aos visitantes conhecerem melhor o assunto da aplicação, estabelecer confiança e obter informações relevantes antes de se envolverem mais profundamente com o projeto em si.

4.2.3 Páginas de jogabilidade

No setor de jogabilidade, é onde são feitas as tarefas de rotina, que por sua vez, é onde o usuário passará a maior parte do tempo durante a utilização da aplicação, pois é aqui onde a gamificação realmente acontece. Ele está dividido em 4 áreas, sendo elas: tarefas, diárias, hábitos e contas.

- Rotina (Tarefas);

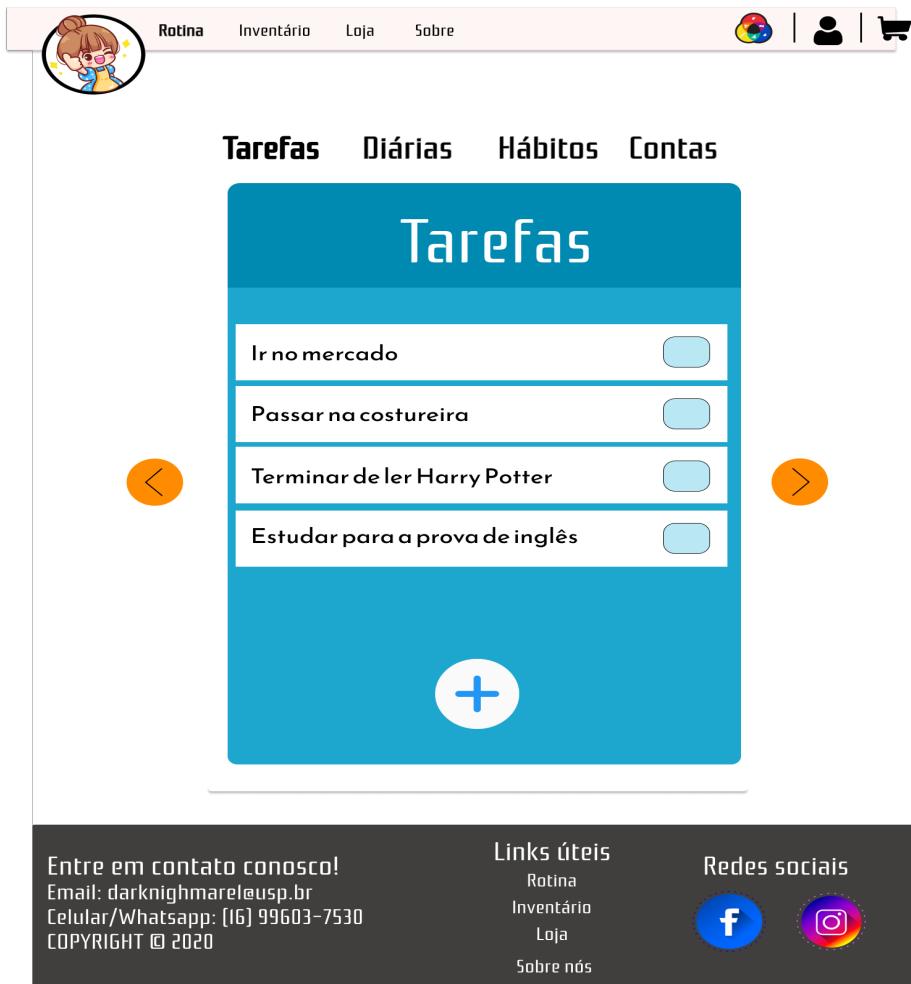


Figura 11: Tela do Figma - Rotina Tarefas

Tarefas: serão tarefas pontuais criadas a partir das necessidades do usuário, onde ele poderá adicionar quantas quiser, e ao cumpri-las, o *check* de tarefa completa será marcada e o usuário ganhará moedas como recompensa por cumprir as suas tarefas estipuladas para si próprio.

- Rotina (Diárias)

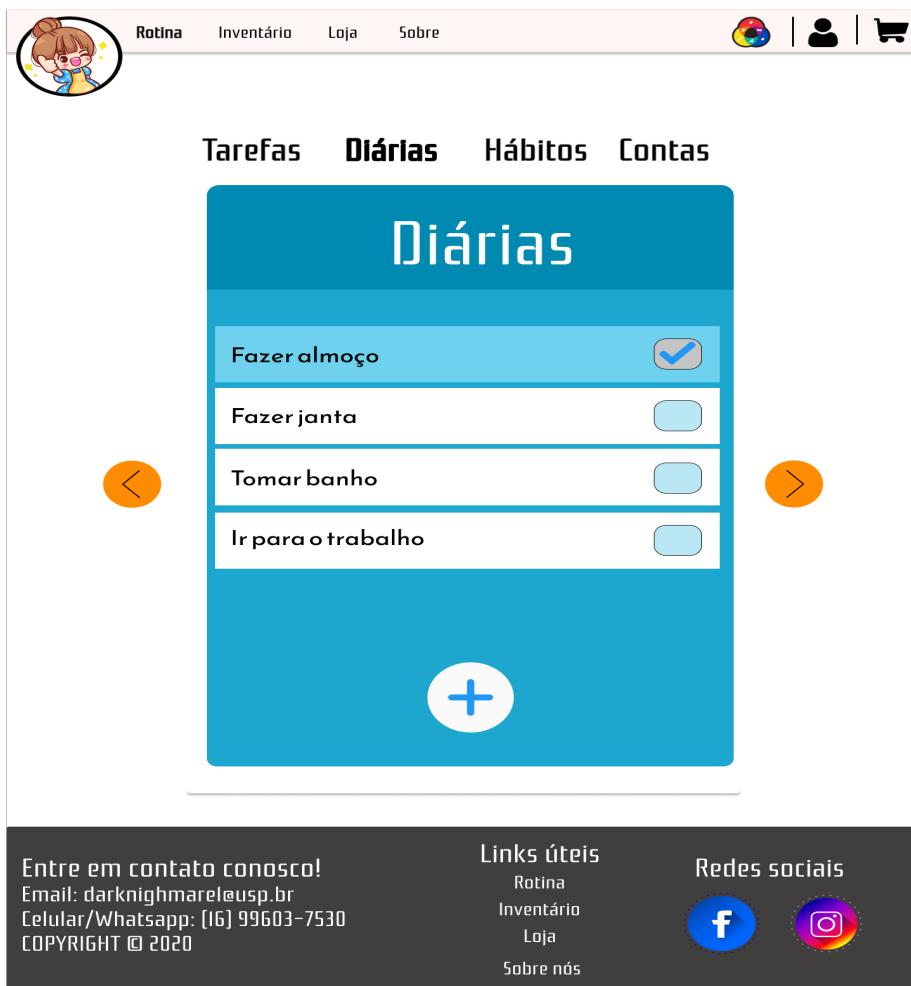


Figura 12: Tela do Figma - Rotina Diárias

Diárias: as tarefas diárias em uma rotina referem-se às atividades que são realizadas regularmente, geralmente em um dia de trabalho típico ou em uma programação diária. Essa área possui um *check* que marcará se a tarefa diária foi ou não cumprida no dia. E elas iniciaram todos os dias para que o usuário sempre refaça as mesmas diariamente.

- Rotina (Hábitos)

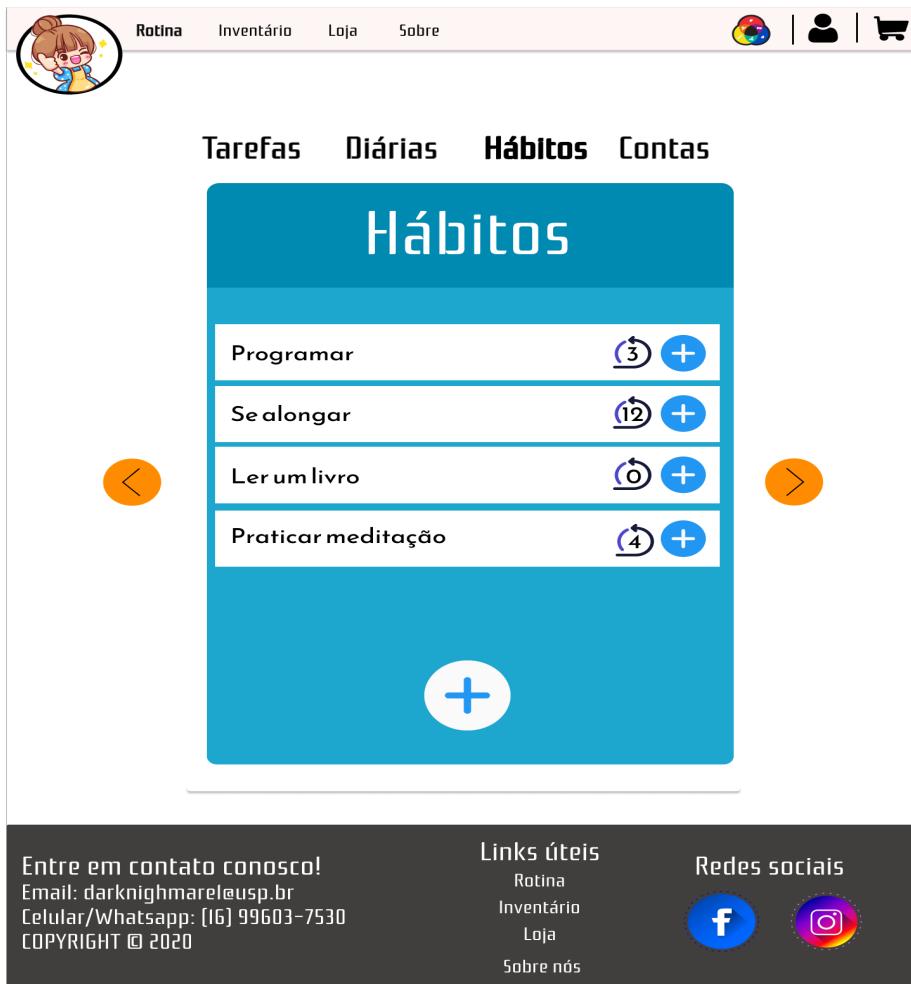


Figura 13: Tela do Figma - Rotina Hábitos.

Hábitos: são comportamentos ou ações que são realizados repetidamente de forma regular, muitas vezes inconscientemente. São ações que se tornaram rotineiras e automáticas devido à repetição constante ao longo do tempo. E para ajudar na construção de bons hábitos é utilizado um contador, que a cada vez que é cumprido determinado hábito, o contador sobe, assim ajudando na motivação e responsabilidade do usuário para com aquele hábito em específico. Também auxilia no desenvolvimento de consistência e estabelecimento de metas.

- Rotina (Contas)

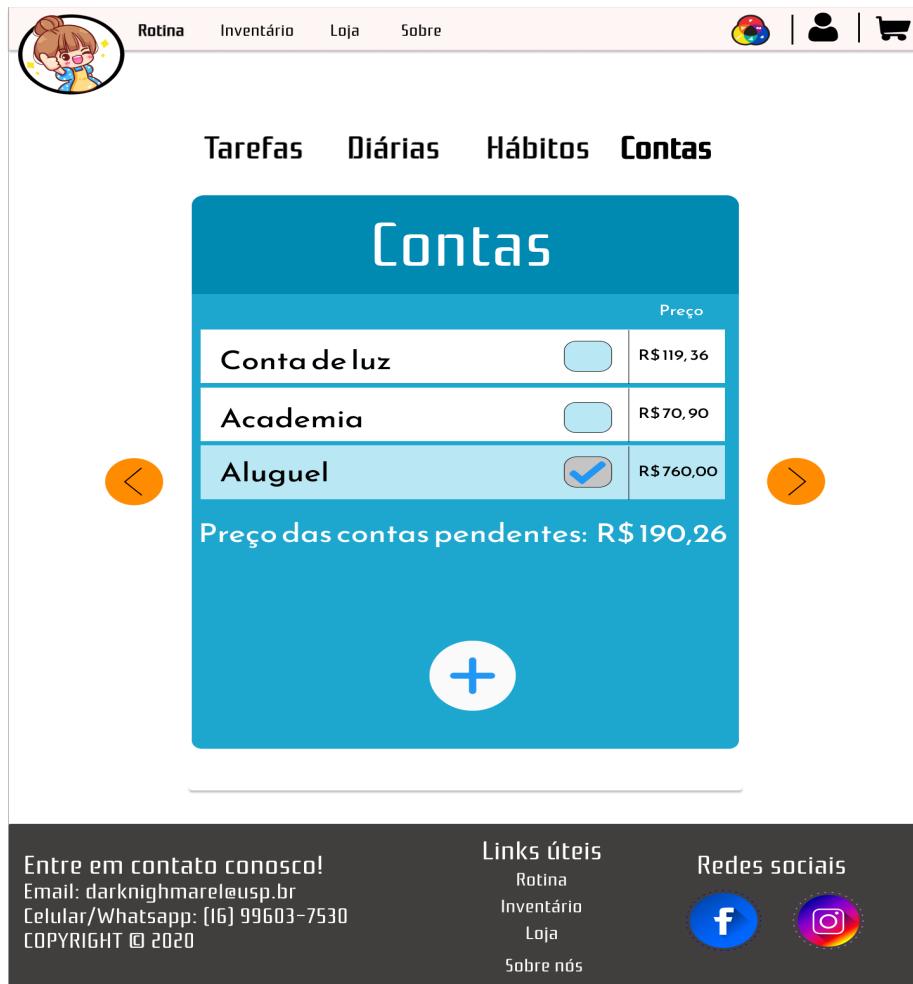


Figura 14: Tela do Figma - Rotina Contas

Contas: é uma forma de auxiliar o usuário no controle e gestão de suas contas, aqui ele poderá adicionar a data limite de suas contas e o valor das mesmas. Esse setor foi projetado para ajudar os usuários a acompanhar suas despesas de forma conveniente e eficiente.

- Rotina (Adicionar Tarefas, Diárias, Hábitos)



Figura 15: Tela do Figma - Adição Tarefas, Diárias ou Hábitos

Essa tela representa o *pop up* de adição de tarefas, diárias e hábitos.

- Rotina (Adicionar Contas)

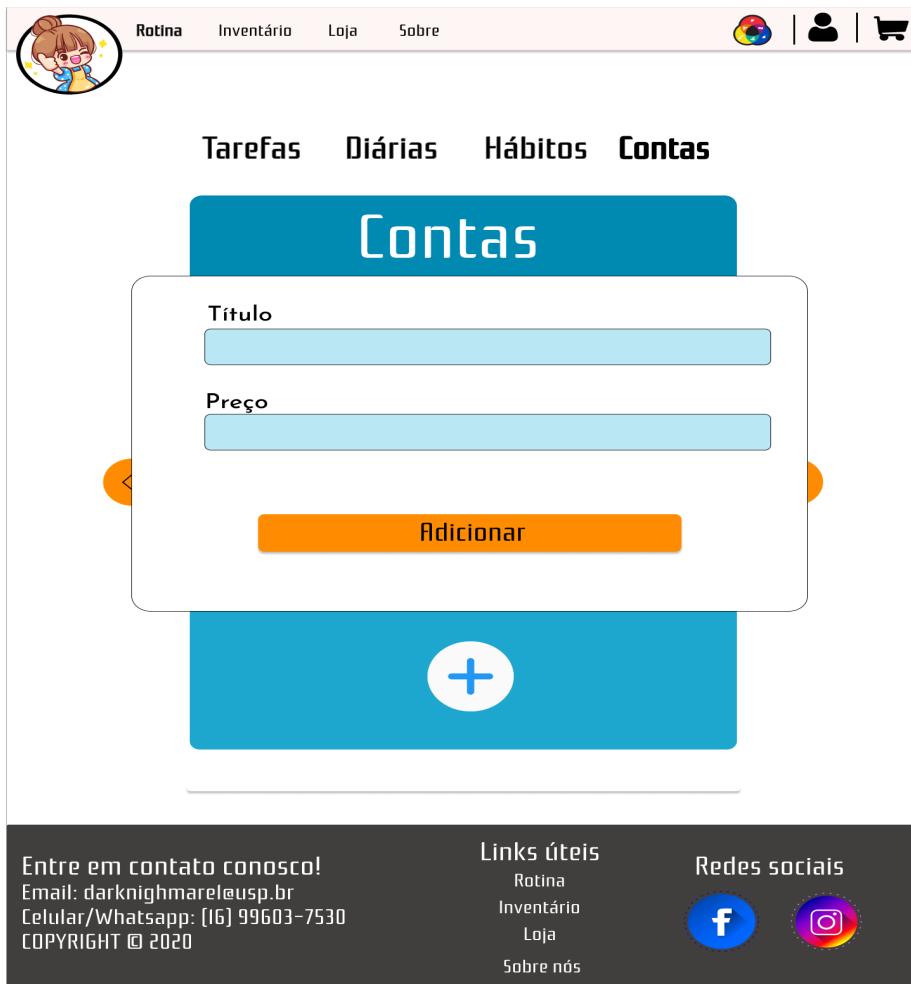


Figura 16: Tela do Figma - Adição de Contas

Essa tela representa o pop up de adição de contas.

4.2.4 Páginas de Produtos

A página de produtos é a loja do aplicativo, que é uma plataforma que auxilia na produtividade e gamificação, ajudando a transformar suas tarefas diárias em uma experiência divertida e recompensadora. Essa loja é um recurso onde os usuários podem gastar moedas virtuais para adquirir camisas, calças e cabelos para o seu personagem. É oferecido uma variedade de itens virtuais que podem ser usados

para personalizar o avatar do usuário e tornar a experiência de jogo mais envolvente.

Os usuários podem adquirir moedas virtuais completando tarefas normais, tarefas diárias, hábitos e pagando suas contas estabelecidos por eles mesmos dentro da plataforma. Essas moedas podem então ser gastas na loja para comprar os itens desejados. É importante notar que a loja *Task Complete* é uma parte integrante do jogo e não envolve transações de dinheiro real. Todas as transações ocorrem no ambiente virtual do jogo e os itens adquiridos na loja têm impacto apenas dentro da plataforma. Essa loja é projetada para fornecer aos jogadores uma sensação de progresso, recompensa e personalização, incentivando-os a realizar suas tarefas e hábitos de forma mais divertida e motivadora.

- Roupas

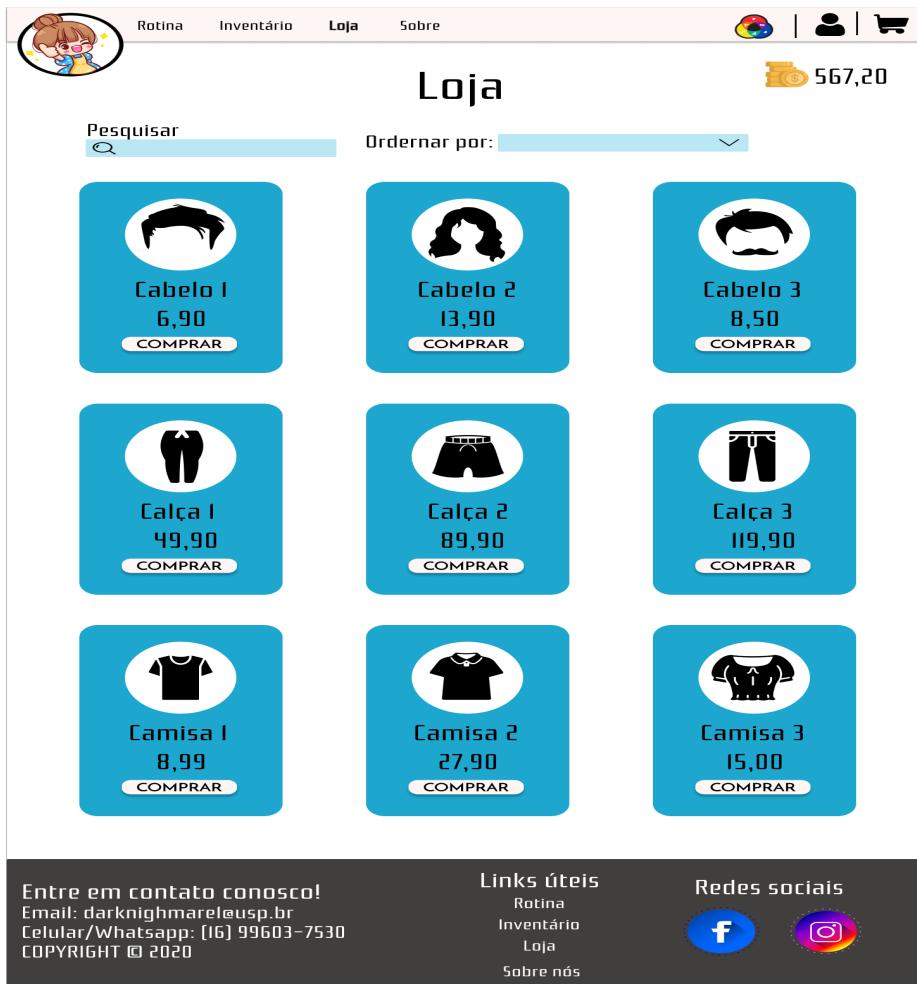


Figura 17: Tela do Figma - Roupas

Seção onde é localizado os itens da loja, que são basicamente roupas para o personagem do usuário.

- Carrinho

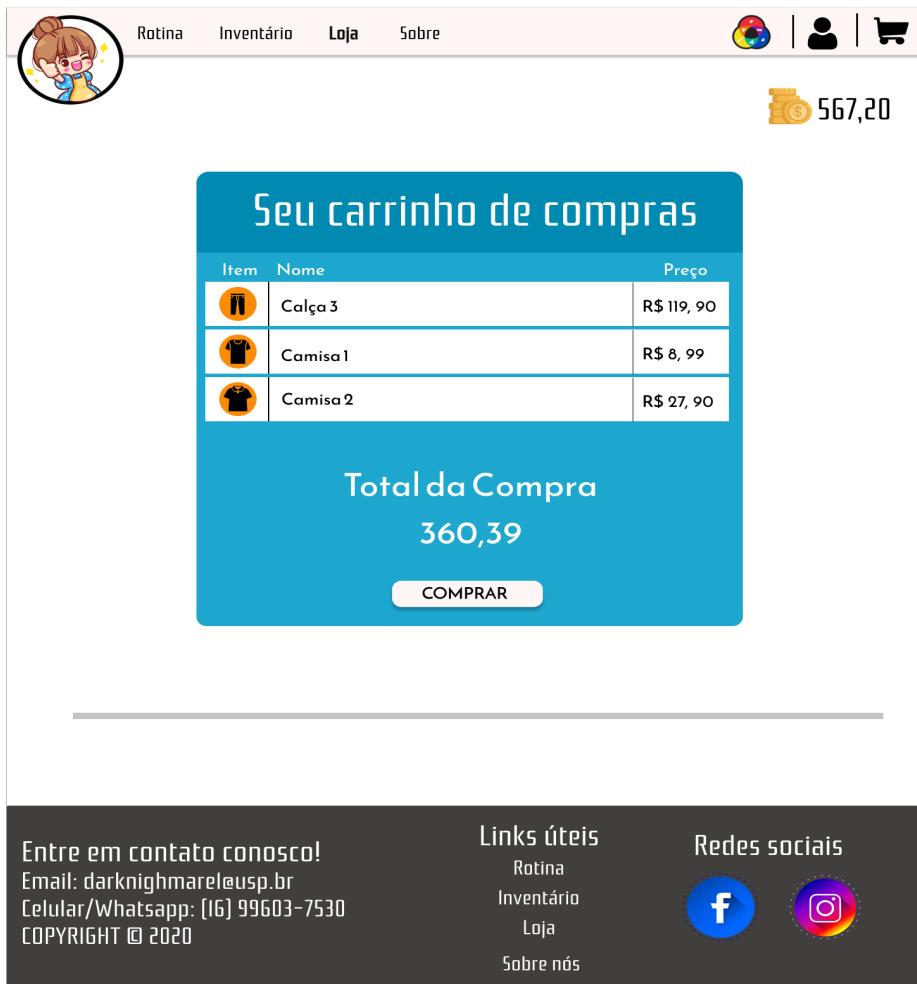


Figura 18: Tela do Figma - Carrinho

Seção onde os itens comprados irão para que seja aprovado a compra e então adicionados ao inventário do usuário.

4.2.5 Área de usuários

- Usuário - Perfil

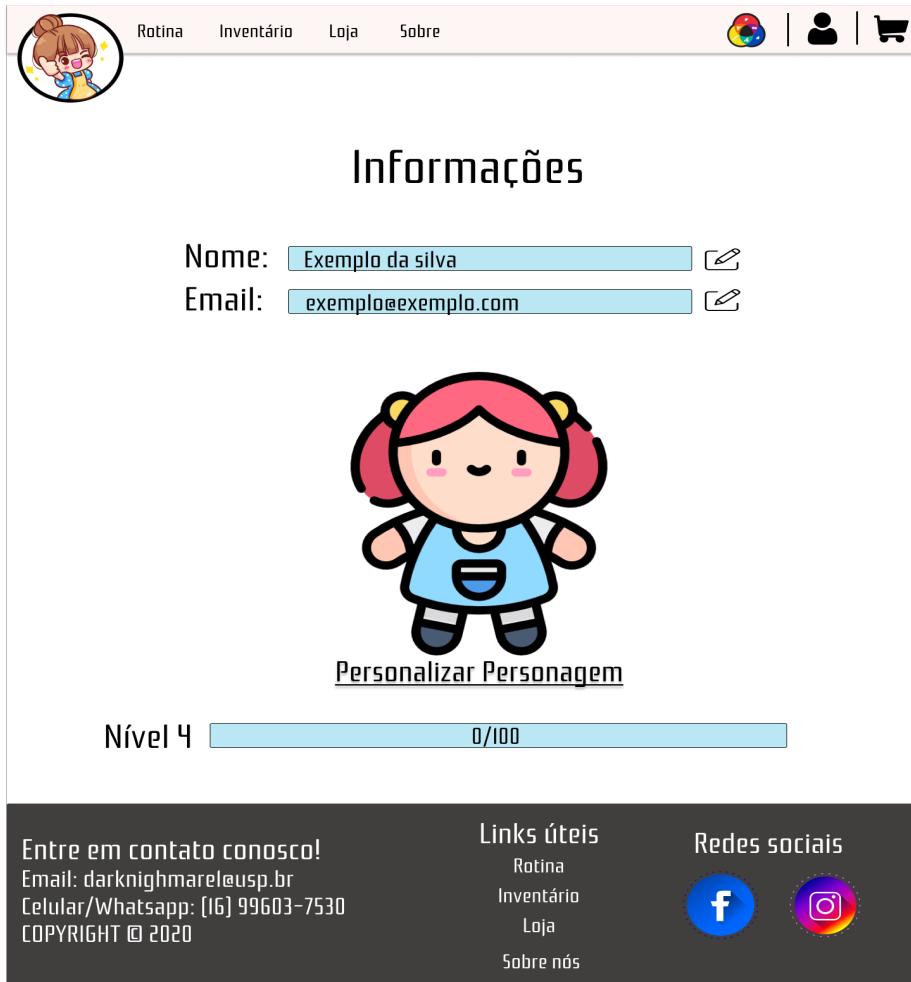


Figura 19: Tela do Figma - Perfil

O setor de perfil é utilizado para visualizar as informações básicas dos usuários, sendo elas nome e e-mail. Também como visualizar o seu progresso por meio do nível do seu personagem, onde é possível ver a barra de progresso e o nível atual que o seu personagem se encontra.

- Usuário - Inventário

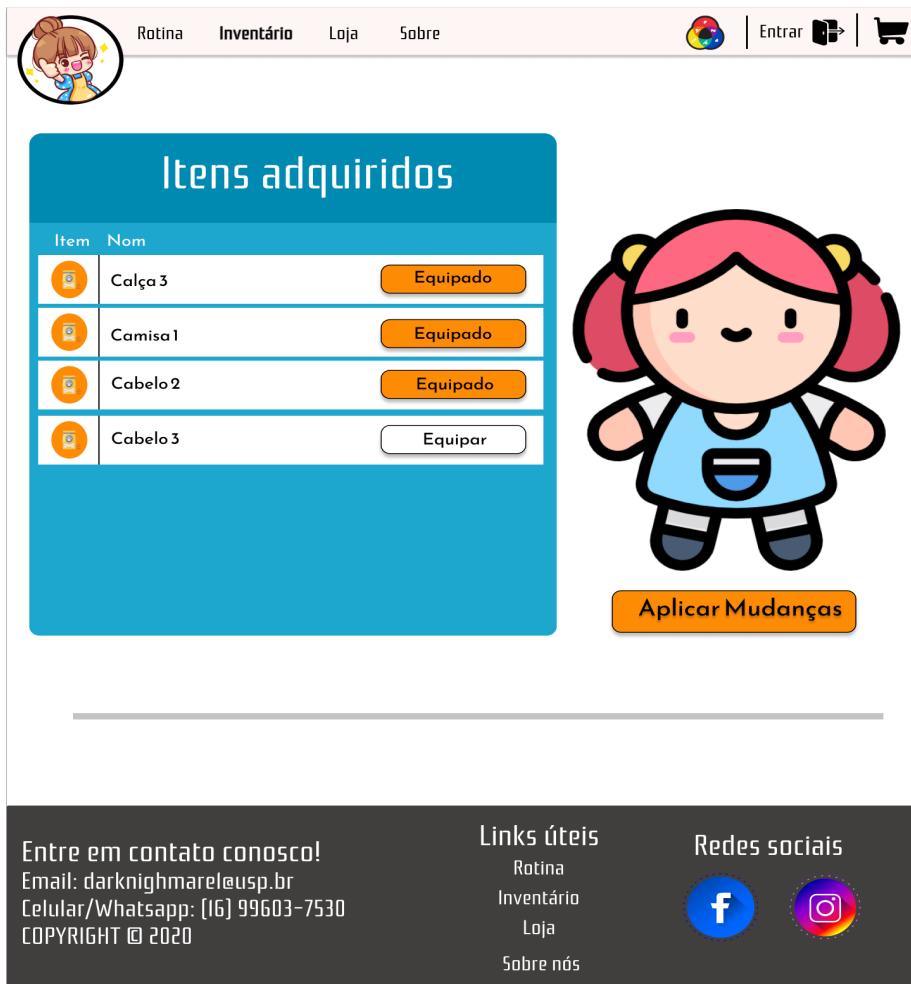


Figura 20: Tela do Figma - Inventário

O inventário é uma seção primordialmente de customização, onde os usuários podem personalizar o seu avatar no jogo. Ele fornece uma visão geral dos itens possuídos dentro da plataforma.

Essa é uma forma dos jogadores acompanharem seu progresso, celebrarem suas conquistas e interagirem com os itens disponíveis no jogo. Ele oferece uma visão geral do desempenho individual por meio dos itens adquiridos e da liberdade de criação que o usuário

tem com o seu personagem virtual.

- Usuário - Criar conta

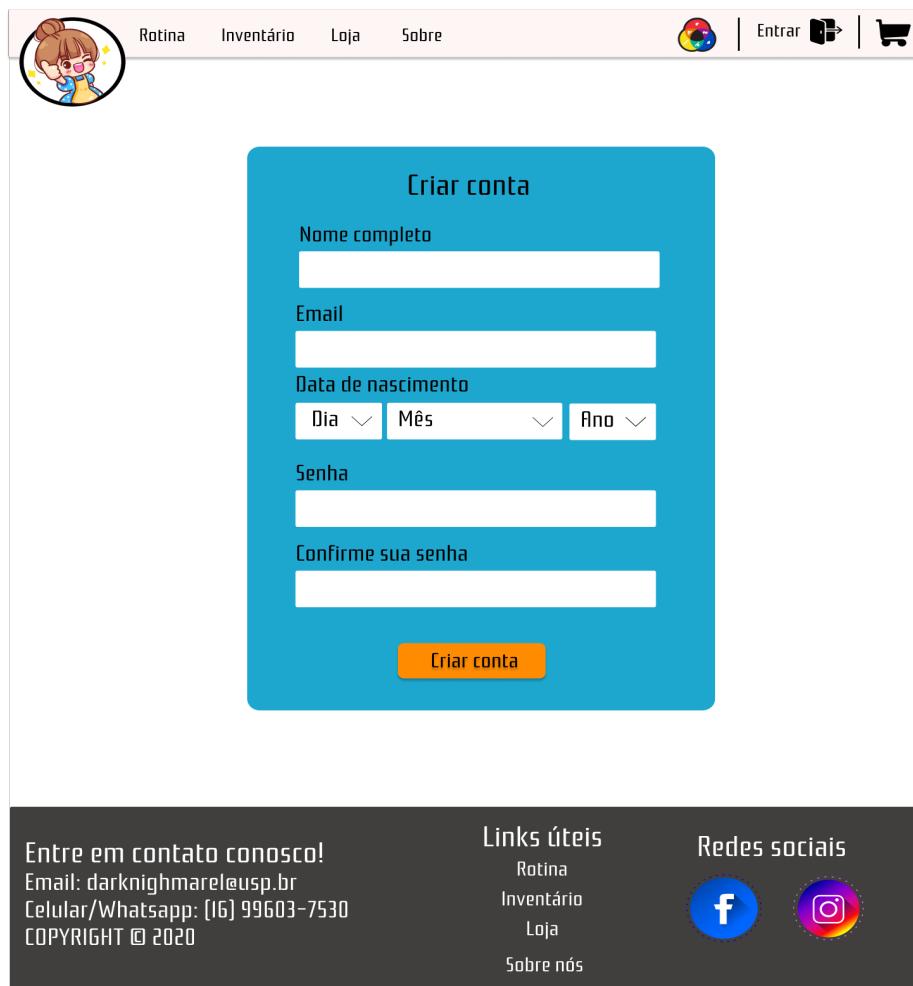


Figura 21: Tela do Figma - Criar Conta

Setor usado para a criação de conta de novos usuários que ainda não possuem uma conta na aplicação.

- Usuário - Login



Figura 22: Tela do Figma - Login

Setor usado para realizar a entrada de um usuário na aplicação, assim é possível carregar os seu progresso anteriormente salvo.

4.3 Implementação do jogo: Front-End

4.3.1 Descrição da aplicação

A solução para o Frontend foi a utilização do framework React. Mais a fundo, React é um framework JavaScript de código aberto mantido pelo *Facebook*. Ele é projetado para facilitar a criação de

interfaces de usuário interativas e escaláveis para aplicações *Web*. O React segue o conceito de ”componentização”, dividindo a interface do usuário em componentes reutilizáveis e independentes, que são atualizados de forma eficiente quando o estado da aplicação muda.

Ele foi escolhido, pois é um ótimo *framework* e bastante utilizado no mercado atualmente, tendo suporte para a linguagem. Além das diversas razões pelas quais o *React* <https://react.dev/> é amplamente utilizado e considerado bom para o desenvolvimento de front-end, tendo diversas vantagens sobre alguns outros frameworks em questão, sendo algumas delas:

- **Componentização:** O React utiliza um modelo de componentes reutilizáveis, permitindo que os desenvolvedores dividam a interface do usuário em componentes independentes e coesos. Esses componentes podem ser compostos e combinados para construir interfaces complexas, facilitando a criação e manutenção do código;
- **Virtual DOM:** O *React* utiliza um *Virtual DOM* (*Documento de Objeto Modelo Virtual*) para renderizar as alterações na interface do usuário de forma eficiente. Em vez de atualizar diretamente o DOM do navegador, o React cria uma representação virtual do DOM em memória e compara-a com o estado anterior para determinar as mínimas alterações necessárias. Essa abordagem otimizada melhora o desempenho e a responsividade das aplicações;
- **JSX:** O React permite a escrita de código utilizando a sintaxe *JSX*, que combina *HTML* e *JavaScript*. O *JSX* facilita a criação de componentes e a manipulação do *DOM*, fornecendo uma forma mais intuitiva e expressiva de criar interfaces do usuário;
- **Reatividade:** O *React* possui um sistema de atualização automática eficiente, o que significa que quando o estado de um componente muda, o *React* atualiza automaticamente as partes relevantes da interface do usuário. Isso torna o desenvolvimento mais produtivo, pois os desenvolvedores podem se concentrar na lógica e nos dados, em vez de se preocuparem com a atualização manual do *DOM*;

- **Comunidade e Ecossistema:** O *React* possui uma comunidade ativa e um ecossistema robusto. Existem muitas bibliotecas, ferramentas e recursos disponíveis que facilitam o desenvolvimento com *React*. Além disso, o *React* é mantido pelo *Facebook* e possui um grande número de desenvolvedores contribuindo para seu crescimento e evolução contínua.
- **Renderização do lado do servidor:** O *React* também suporta a renderização do lado do servidor (*Server-Side Rendering - SSR*), o que significa que as páginas podem ser renderizadas no servidor e enviadas para o navegador, reduzindo o tempo de carregamento inicial e melhorando o SEO (Search Engine Optimization);
- **Mobile Development:** O *React Native*, uma extensão do *React*, permite o desenvolvimento de aplicativos móveis nativos para *iOS* e *Android* usando *JavaScript*. Compartilhando grande parte do código com o *React*, o *React Native* permite uma abordagem eficiente e produtiva para o desenvolvimento de aplicativos móveis multiplataforma.

4.4 Setores da aplicação

4.4.1 Login

A solução para o Frontend conta com as funcionalidades de *login* e de criação de conta para o usuário

Na parte do *login* de usuários, tem as funcionalidades:

- Login
- Cadastro de usuário

O setor de *login* foi feito utilizando uma integração com o servidor via json para realizar as chamadas ao backend que, por sua vez, salva os usuários cadastrados, e quando for feito o *login* ele verifica com esses dados do backend para autorizar ou não o *login* do usuário. Após feito o *login*, o usuário terá o acesso permitido para utilizar o site livremente.

4.4.2 Rotina

Nesse setor do código foi utilizado funções que definem qual tela mostrar baseado na escolha do usuário, que poderá ser "Tarefas", "Diárias", "Hábitos" ou "Contas". E então será mostrado as suas telas com suas respectivas funcionalidades de adição, remoção, contadores, conclusão de tarefas, bem como suas recompensas adicionando na moeda virtual ao completar essas atividades.

4.4.3 Loja

Na parte da loja é pego os itens restantes através da sincronização com o Back End, e então mostrado na loja para que o usuário possa usar a moeda virtual para comprar esses itens. Quando os itens são comprados, eles são redirecionados a base de dados do inventário e retirados visualmente e virtualmente da tela da loja.

4.4.4 Perfil

Aqui no perfil é feito um sistema de nível onde é aumentado a barra de *XP* e caso chegue a 100, o nível aumentará em um para, mostrando isso visualmente embaixo do personagem do usuário. Também é possível verificar as informações como nome e e-mail do usuário, bem como alterar essas informações a qualquer momento com as ferramentas de edição que chamarão funções que farão chamadas de edição ao *Back End*.

4.4.5 Inventário

Ao ser comprado os itens na loja, eles serão redirecionados ao inventário, onde podem ser vestidos pelo personagem do usuário, porém apenas podendo vestir um item para cabeça, um item para o tronco e outro para a parte inferior do personagem. Assim, o usuário clica em "Aplicar Mudanças" e então será salvo as alterações e alterar a aparência do personagem.

4.4.6 Sobre

Nesta área só estão alocados as informações básicas do projeto e também é dado um contexto melhor para as pessoas que estão curiosas para saber mais sobre o Task Complete e de sua criação.

4.5 Implementação do jogo: Back-End

4.5.1 Descrição da aplicação

A solução para o *Backend* conta com a adição do *json-server*, sendo ele uma biblioteca *JavaScript* que permite criar rapidamente uma *API RESTful (Representational State Transfer)* completa com dados persistentes. Ele é frequentemente usado durante o desenvolvimento de aplicativos *Front End* para simular uma *API* real e fornecer dados fictícios para testar e prototipar interfaces de usuário.

Também foi uma ótima opção de escolha para ser utilizado no projeto, pois é fácil de usar e não requer nenhuma configuração de servidor complexa. Ele usa um arquivo *JSON* como fonte de dados e expõe endpoints *RESTful* para executar operações *CRUD (Create, Read, Update, Delete)* nesses dados.

O *JSON Server* utiliza os conceitos da *API REST (Representational State Transfer)*. O *JSON Server* é uma biblioteca *Node.js* que permite criar rapidamente uma *API REST* simulada a partir de um arquivo *JSON*. Ele segue os princípios da arquitetura *REST*, que é um estilo de arquitetura de software amplamente utilizado na construção de serviços *web*.

Além disso, ele possui vários recursos, mas tratando apenas de alguns principais que o *JSON Server* conta que foram utilizados no projeto são:

- **Roteamento:** Define automaticamente rotas para cada chave no arquivo *JSON*. Por exemplo, se o arquivo *JSON* contiver uma matriz de objetos chamada "usuários", o *JSON Server* criará uma rota `/usuarios` que retorna todos os usuários e rotas individuais para cada usuário com base em seu identificador exclusivo;
- **Operações *CRUD*:** Suporta as operações básicas de criação, leitura, atualização e exclusão de dados (*CRUD*). Então é possível fazer solicitações *HTTP POST, GET, PUT e DELETE* para interagir com os dados;
- **Filtragem:** Permite filtrar dados com base em parâmetros de consulta. Por exemplo, você pode fazer uma solicitação *GET* para `/usuarios?idade=30` para obter todos os usuários com idade igual a 30;

- **Ordenação:** É possível ordenar os dados por qualquer campo usando o parâmetro ”*sort*”. Por exemplo, uma solicitação *GET* para a rota de usuário com *sort* sendo igual a nome, então retornará os usuários ordenados pelo campo ”nome”;
- **Paginação:** Oferece suporte à paginação dos resultados usando os parâmetros ”*page*” e ”*limit*”. Por exemplo, uma solicitação *GET* para a rota de usuários com o número da página e limite de 10, então retornará a segunda página de 10 usuários;
- **Relacionamentos:** Permite criar relacionamentos entre os dados usando campos de identificador exclusivo. Por exemplo, você pode ter um objeto ”*post*” com um campo ”*userId*” que se refere ao identificador exclusivo de um objeto ”usuário”;
- **Persistência:** Salva as alterações feitas nos dados em tempo de execução em um arquivo *JSON*. Isso permite que você retome o estado dos dados entre as execuções.

O *JSON Server* é uma ferramenta útil para desenvolvedores que desejam criar rapidamente uma API RESTful sem a necessidade de configurar um servidor complexo.

4.5.2 API Rest

Para essa aplicação, será utilizado o conceito de *API Rest (Representational State Transfer)*. Sendo ela basicamente um estilo de arquitetura de software para sistemas de comunicação distribuída. É um conjunto de princípios e convenções que define como as solicitações e respostas devem ser feitas entre clientes e servidores na *web*.

Uma *API RESTful* é baseada no protocolo *HTTP (Hypertext Transfer Protocol)* e utiliza seus métodos, como *GET*, *POST*, *PUT* e *DELETE*, para permitir a comunicação e interação entre os sistemas. Ela é chamada de ”representacional” porque os recursos (como dados ou serviços) são representados e acessados através de *URLs* (Uniform Resource Locators).

4.6 Dados guardados no backend

No backend *json-server* foram guardados diversos tipos de dados de setores diferentes para comportar as diferentes áreas da aplicação,

eles são descritos mais abaixo:

- Usuários: Neste setor é salvo as informações dos usuários para gerenciar as informações do usuário que está logado no momento. Os campos salvos são:
 - id;
 - nome;
 - email;
 - senha;
 - dinheiro;
- Loja: Neste setor é salvo as informações dos itens da loja que poderão ser comprados pelos usuários. Os campos salvos são:
 - id;
 - nome;
 - preço;
 - imagem;
- Inventário: Neste setor é salvo as informações dos itens comprados pelo usuário, que estarão alocados em seu inventário. Os campos salvos são:
 - id;
 - nome;
 - preço;
 - imagem;
 - equipado (booleano);
- Tarefas: Neste setor é salvo as informações das tarefas criadas pelo usuário, elas aparecem uma única vez e quando são cumpridas desaparecem. Os campos salvos são:
 - id;
 - texto;
 - dia;
 - importante (booleano);

- Diárias: Neste setor é salvo as informações das tarefas diárias criadas pelo usuário, elas aparecerão todo dia para serem cumpridas. Os campos salvos são:
 - id;
 - texto;
 - completado (booleano);
- Hábitos: Neste setor é salvo as informações os hábitos do usuário, eles nunca desaparecem e terão um contador para marcar como o usuário está se saindo com determinado hábito. Os campos salvos são:
 - id;
 - texto;
 - contador;
- Contas: Neste setor é salvo as informações das contas a serem pagas pelo usuário. Os campos salvos são:
 - id;
 - texto;
 - dia;
 - valor;
 - importante (booleano);

4.7 Acessando o projeto

Para iniciar o projeto é preciso ter o Node instalado pois será usado suas funcionalidades relacionadas ao npm, que por sua vez, é uma ferramenta que permite aos desenvolvedores instalar, compartilhar e gerenciar as dependências de um projeto de forma fácil e eficiente.

É possível instalá-lo através desse link: <https://nodejs.org/pt-br/download/>.

É importante que o usuário também tenha o React.js instalado: <https://react-cn.github.io/react/downloads.html>.

Após instalar Node, é preciso instalar o npm, para isso rode:
`npm install`.

Agora é preciso do json-server para rodar o Backend, e para isso será usado uma funcionalidade do próprio npm, portanto rode:
`npm install json-server`.

E esse outro comando é preciso para conseguir iniciar o backend, também utilizado no terminal:

`npm run server`.

Por fim, é preciso iniciar o frontend utilizando o comando:
`npm start`.

5 Conclusão

5.1 Contribuições do Trabalho

Uma solução de gamificação de rotina, como o *Task Complete*, pode oferecer uma série de benefícios para a sociedade. Ao transformar tarefas diárias em um jogo divertido, com recompensas e desafios, essa abordagem motiva e engaja as pessoas, ajudando-as a cumprir suas atividades e estabelecer hábitos saudáveis, desta forma auxiliando pessoas com dificuldades intelectuais, que é o tema principal desse projeto e até mesmo pessoas sem essas dificuldades.

Além disso, a gamificação de rotina pode aumentar a produtividade, eficiência e autodisciplina das pessoas, ao mesmo tempo em que promove a autogestão e o desenvolvimento de habilidades de planejamento.

A existência de uma comunidade e colaboração entre os usuários também cria um ambiente de apoio e incentivo mútuo. No geral, a gamificação de rotina pode melhorar a qualidade de vida das pessoas, tornando suas rotinas mais motivadoras.

5.2 Dificuldades e Limitações do Trabalho

A principal dificuldade do projeto foi encontrar pessoas para testar e avaliar a aplicação, portanto essa questão de testabilidade é

uma limitação do projeto, que será sanada futuramente quando a solução for entregue a usuários para poder ser testada.

Outro ponto importante foi o conhecimento limitado nas tecnologias como React, usado para o Front End e também tecnologias de para o Back End, o que acabou atrapalhando um pouco no desenvolvimento do projeto, porém essa limitação foi contornada através de pesquisa ao decorrer do a construção do mesmo.

5.3 Considerações Finais

Ao concluir este projeto de graduação sobre o desenvolvimento de um jogo web que gamifica a rotina, é possível perceber que durante todo o processo foi buscado explorar o potencial da gamificação como abordagem inovadora para tornar as atividades diárias mais interessantes e motivadoras

A proposta consistiu em criar um site interativo que transforma tarefas cotidianas em desafios divertidos, recompensando os usuários à medida que eles avancem em suas rotinas. Através de elementos como recompensas virtuais, sendo em forma de nível e roupas para o personagem, foi buscado estabelecer um ambiente gamificado que encoraje as pessoas a se envolverem mais profundamente em suas obrigações diárias.

Durante o desenvolvimento do site, utilizamos tecnologias modernas, como o framework React.js para o Front End e o JSON Server para o Back End, a fim de criar uma experiência de usuário fluida e responsiva. Além disso, o projeto conta com uma prototipação de alta fidelidade utilizando a ferramenta Figma, o que permitiu visualizar o produto final e realizar ajustes antes mesmo da implementação completa.

O projeto implica que a gamificação da rotina tem o potencial de impactar positivamente a vida das pessoas, tornando suas tarefas diárias mais envolventes e incentivando a manutenção de hábitos saudáveis. No entanto, é reconhecido que este projeto é apenas o começo de um processo contínuo de desenvolvimento e aprimoramento.

Finalmente, é de se esperar que este projeto possa servir para futuras pesquisas e desenvolvimentos na área de gamificação de rotina. Ao continuar explorando essa abordagem, há a esperança de contribuir para a melhoria do bem-estar e da produtividade das

pessoas.

Referências

PACHECO, MÁRCIO. **Tennis for Two, o primeiro game da história, completa 55 anos.** [S. l.], 2013. Disponível em: <https://gamehall.com.br/tennis-for-two-o-primeiro-game-da-historia-completa-55-anos/>. Acesso em: 12 jun. 2023. Citado nas páginas 16 e 17.

CARLOS MORAES ROSA, ADRIANO. **ENSINO E EDUCAÇÃO: USO DA GAMIFICAÇÃO NA MATEMÁTICA.** [S. l.], 2021. Disponível em: <https://www.nucleodoconhecimento.com.br/educacao/gamificacao-na-matematica>. Acesso em: 12 jun. 2023. Citado nas páginas 17, 18 e 19.

KALINOWSKI, Marcos; OLIVEIRA SPÍNOLA, Rodrigo; CLAUDIO DIAS NETO, Arilo; BOTT, Andrew; HORTA TRAVASSOS, Guilherme. **Inspeções de Requisitos de Software em Desenvolvimento Incremental: Uma Experiência Prática.** [S. l.], 2007. Disponível em: <https://sol.sbc.org.br/index.php/sbqs/article/view/15591>. Acesso em: 12 jun. 2023.

SANTOS, DAVID. **Desenvolvimento Iterativo e Incremental.** [S. l.], 2019. Disponível em: <https://codejourney.com.br/desenvolvimento-iterativo-e-incremental/>. Acesso em: 12 jun. 2023.

PRO, LUDOS. **Gamificação: aprendizado e motivação na sua vida cotidiana.** [S. l.], 2020. Disponível em: <https://www.ludospro.com.br/blog/gamificacao-da-vida#:~:text=A20gamificaCA7C3A3o20pode20ajudar20as,mesmo20na20correria20da20rotina>. Acesso em: 12 jun. 2023.

FILHO, CARDOSO; JOSÉ, RAIMUNDO. **Uma visão sobre avaliação de usabilidade e acessibilidade no IHC.** Orientador: Yuska Paola Costa Aguiar. 2017. Trabalho de conclusão de curso (CIENCIAS EXATAS E DA TERRA::CIENCIA DA COMPUTACAO) - Universidade Federal da Paraíba, [S. l.], 2017. Disponível em: https://repositorio.ufpb.br/jspui/handle/123456789/3352?locale=pt_BR. Acesso em: 12 jun. 2023. Citado nas páginas 19 e 20

FLÁVIA FIDELIS FERNANDES, Ana. **Análise de requisitos de IHC em aplicativos para comunicação com**

surdos em LIBRAS. Orientador: Mirella Silva Junqueira. 2017. Trabalho de conclusão de curso (CIENCIAS EXATAS E DA TERRA::CIENCIA DA COMPUTACAO::SISTEMAS DE COMPUTACAO::SOFTWARE BASICO) - Universidade Federal de Uberlândia, [S. l.], 2017. Disponível em: https://repositorio.ufpb.br/jspui/handle/123456789/3352?locale=pt_BR. Acesso em : 12 jun. 2023.

SHAFFER, JEFFREY. Cinco dicas para criar visualizações adaptadas para daltônicos. [S. l.], 2022. Disponível em: <https://www.tableau.com/pt-br/blog/examining-data-viz-rules-dont-use-red-green-together#:text=Por20exemplo2C20azul2Flaranja20C3A9,distC3BArbios20geralmente20enxergam20o20azul>. Acesso em: 12 jun. 2023.

FERNANDES, JULIANA. Acessibilidade para Daltônicos — Casos de falha e sucesso. [S. l.], 2017. Disponível em: <https://medium.com/acessibilito/acessibilidade-para-dalt%C3%B4nicos-casos-de-falha-e-sucesso-d76bf6e4261e>. Acesso em: 12 jun. 2023.

MARINO SILVA, TIAGO. Design, Desenvolvimento e Avaliação do Jogo de Narrativa Invertida para a Plataforma RUFUS. 2022. Trabalho de conclusão de curso (Bacharelado em Ciência da Computação) - Universidade de São Paulo, [S. l.], 2022. Disponível em: https://tcc.icmc.usp.br/monografias/monografias_original/2022/2104/26492104 – Tiago.pdf. Acesso em : 12 jun. 2023.

MOTTA, JANINE. Marketing de Conteúdo: como tornar os materiais acessíveis para pessoas com deficiência visual. [S. l.], 2018. Disponível em: <https://resultadosdigitais.com.br/marketing/conteudo-acessivel-para-deficientes-visuais/>. Acesso em: 12 jun. 2023.

GUIMARÃES CRUZ, AGNES. Uno ganha versão para daltônicos. [S. l.], 2017. Disponível em: <https://www.b9.com.br/78728/uno-ganha-versao-para-daltonicos/>. Acesso em: 12 jun. 2023. Citado nas páginas 24 e 25.

- WCAG, EQUIPE (org.). Guia WCAG. [S. l.], 2023. Disponível em: <https://guia-wcag.com/>. Acesso em: 12 jun. 2023. Citado nas páginas 26 e 27.

- WCAG, EQUIPE (org.). COGNITIVE ACCESSIBILITY COMMUNITY GROUP. [S. l.], 2023. Disponível em: <https://www.w3.org/community/coga-community/>. Acesso em: 12 jun. 2023.
- FELIPE PATIKOWSKI CHEIRAN, Jean (org.). Jogos Inclusivos: diretrizes de acessibilidade para jogos digitais. [S. l.], 2013. Disponível em: <https://lume.ufrgs.br/handle/10183/77230>. Acesso em: 12 jun. 2023.
- WACG, EQUIPE (org.). Web Content Accessibility Guidelines – WCAG 2.1. [S. l.], 2018. Disponível em: <https://cta.ifrs.edu.br/web-content-accessibility-guidelines-wcag-2-1/:text=O%20documento%20da%20WCAG%20consiste,organiza%C3%A7%C3%B5es%20em%20todo%20o%20mundo>. Acesso em: 12 jun. 2023. Citado nas páginas 25 e 26.
- FELIPE PATIKOWSKI CHEIRAN, Jean. Diretrizes de Acessibilidade para Jogos (DAJ). [S. l.], 2013. Disponível em: <https://www.inf.ufrgs.br/jfp-cheiran/diretrizes/>. Acesso em: 12 jun. 2023. Citado nas páginas 25 e 26.
- DE JESUS VILELA DA ROCHA, Tânia. Acessibilidade e Usabilidade na Internet para Pessoas com Deficiência Intelectual. [S. l.], 2008. Disponível em: <https://www.proquest.com/openview/f70a32592a879a3c0d7e7cbacb3961eb/1?pq-origsite=gscholar&cl=2026366&diss=y>. Acesso em: 12 jun. 2023. Citado nas páginas 27 e 28.
- C. DANTAS, Adilmar et al. Michelzinho: Jogo serio para o ensino de habilidades emocionais em pessoas com autismo ou deficiencia intelectual. 2019. Trabalho de conclusão de curso (Ciencias de Computação) - Faculdade de Ciencia da Computação Universidade Federal de Uberlandia, [S. l.], 2019. Citado nas páginas 28 e 29.