

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

\$tudy - um sistema gamificado de educação financeira

Autor: Rafael Santos Teodosio

Orientador: Prof. Dr. Sergio Antônio Andrade de Freitas

Brasília, DF

2022



Rafael Santos Teodosio

\$tudy - um sistema gamificado de educação financeira

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Sergio Antônio Andrade de Freitas

Brasília, DF

2022

Rafael Santos Teodosio

\$tudy - um sistema gamificado de educação financeira/ Rafael Santos Teodosio.
– Brasília, DF, 2022-
89 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Sergio Antônio Andrade de Freitas

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2022.

1. Educação financeira. 2. Gamificação. I. Prof. Dr. Sergio Antônio Andrade de Freitas. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. \$tudy - um sistema gamificado de educação financeira

CDU 02:141:005.6

Rafael Santos Teodosio

\$tudy - um sistema gamificado de educação financeira

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 29 de abril de 2022:

**Prof. Dr. Sergio Antônio Andrade de
Freitas**
Orientador

Profa. Msc Cristiane Soares Ramos
Convidado 1

Prof. Dr. Tiago Barros Pontes e Silva
Convidado 2

Brasília, DF
2022

Agradecimentos

Agradeço a Deus pela minha vida e tudo que me foi proporcionado de aprendizado e experiências durante o período de escrita desse trabalho.

Agradeço também a minha família por todo apoio e toda motivação durante todo o meu processo de formação.

Aos meus amigos e amigas, em especial Érico Bandeira, Samuel Borges, João Vitor, Max Henrique, Thais Gomes e Michaela Janson por me ajudarem e me alegrarem em meus momentos de desmotivação.

Ao meu orientador Sergio Antônio Andrade de Freitas por me auxiliar na realização desse trabalho, sempre me auxiliar e se importando com minha educação e saúde.

Ao professor Edson Alves da Costa Júnior por me motivar desde o começo da minha jornada na faculdade a ser a melhor versão de mim mesmo e lutar para alcançar meus objetivos de vida.

Resumo

O objetivo deste trabalho é projetar e implementar um sistema gamificado para educação financeira. O método utilizado foi a construção típica da engenharia de software com levantamento de requisitos bem como o estudo da literatura referentes aos frameworks de gamificação, em especial ao *Octalysis*, juntamente com o estudo das metodologias de desenvolvimento de software utilizadas para o planejamento do sistema. No presente trabalho é apresentados o levantamento de requisitos e sua priorização bem como as especificações necessárias para o sistema implementado, também é apresentado um protótipo onde é definida a identidade visual e o protótipo das telas presentes no sistema juntamente com a modelagem relacional dos dados e suas entidades, além de apresentar a arquitetura da aplicação onde mostra como foi estruturada a aplicação, e sua versão final.

Palavras-chaves: Educação financeira. Gamificação. Desenvolvimento de sistemas.

Abstract

The objective of this work is to design and implement a gamified system for financial education. The method used was the typical software engineering construction with requirements gathering as well as a literature study of the The method used was a typical software engineering construction with a survey of requirements and a literature study of gamification frameworks, especially *Octalysis*, along with the study of software development methodologies used to plan the system. In this work we present the requirements survey and its prioritization as well as the necessary specifications for the implemented system. We also present a prototype where we define the visual identity and the prototype of the screens present in the system along with the relational modeling of the data and its entities, and present the application architecture where we show how the application was structured, and its final version.

Key-words: Financial education. Gamification. Systems development.

Lista de ilustrações

| | |
|--|----|
| Figura 1 – Cálculo do IPCA (BCB, 2022b) | 24 |
| Figura 2 – Definição de jogo por Chris Crawford (CRAWFORD, 1991) | 26 |
| Figura 3 – Sistema Octalysis e seus Core Drives (LOGICEARTH, 2022) | 27 |
| Figura 4 – Ciclo de trabalho do Scrum (TECNICON, 2019) | 37 |
| Figura 5 – Quadro do método Kanban (VALENTE, 2020) | 38 |
| Figura 6 – Ciclo de uma entrega na programação extrema (SOMMERVILLE, 2010) | 39 |
| Figura 7 – Paleta de cores do <i>Study</i> | 57 |
| Figura 8 – Identidade visual do <i>Study</i> | 58 |
| Figura 9 – Tela de login | 59 |
| Figura 10 – Tela de cadastro | 60 |
| Figura 11 – Tela do menu principal e tutoria | 61 |
| Figura 12 – Tela de <i>dashboard</i> | 62 |
| Figura 13 – Tela de investimento | 63 |
| Figura 14 – Tela da loja virtual | 64 |
| Figura 15 – Tela de notícias | 65 |
| Figura 16 – Tela de perguntas e respostas | 66 |
| Figura 17 – Tela das missões | 67 |
| Figura 18 – Modelo entidade-relacionamento do <i>Study</i> | 68 |
| Figura 19 – Relação dos Investimentos com fatores externos | 69 |
| Figura 20 – Relação entre loja, modulo e usuário | 70 |
| Figura 21 – Relação entre questões, missão e modulo | 71 |
| Figura 22 – Relação entre Usuário e Investimento | 72 |
| Figura 23 – Relação entre Usuário, Progresso monetário e Progresso educacional | 73 |
| Figura 24 – Representação arquitetural do <i>Study</i> | 76 |
| Figura 25 – Tela de login | 79 |
| Figura 26 – Tela de cadastro | 80 |
| Figura 27 – Tela do menu principal e tutorial | 81 |
| Figura 28 – Tela de investimento | 82 |
| Figura 29 – Tela de perguntas e respostas | 83 |
| Figura 30 – Tela das missões | 84 |

Lista de tabelas

| | |
|---|----|
| Tabela 1 – Técnicas presentes no Octalysis (CHOU, 2015) | 30 |
| Tabela 2 – Técnicas por <i>Core Drive</i> (CHOU, 2015) | 41 |
| Tabela 3 – Requisito por técnica | 42 |
| Tabela 4 – Requisitos Funcionais/Não Funcionais | 54 |
| Tabela 5 – Ferramentas utilizadas e suas versões | 76 |
| Tabela 6 – Requisitos para o mínimo produto viável | 77 |
| Tabela 7 – Tarefas realizadas por <i>sprint</i> | 78 |

Lista de abreviaturas e siglas

| | |
|-----|-------------------------|
| RF | Requisito funcional |
| RNF | Requisito não funcional |

Sumário

| | | |
|------------|-----------------------------------|-----------|
| 1 | INTRODUÇÃO | 21 |
| 1.1 | Motivação | 21 |
| 1.2 | Objetivo | 21 |
| 1.3 | Metodologia | 21 |
| 1.4 | Estrutura da monografia | 22 |
| 2 | REVISÃO DA LITERATURA | 23 |
| 2.1 | Educação financeira | 23 |
| 2.1.1 | Princípio da educação financeira | 23 |
| 2.1.2 | Inflação | 23 |
| 2.1.3 | Juros | 24 |
| 2.1.3.1 | Taxa de juros | 24 |
| 2.1.3.2 | Juros simples | 24 |
| 2.1.3.3 | Juros compostos | 25 |
| 2.1.4 | Renda fixa | 25 |
| 2.1.5 | Renda variável | 25 |
| 2.1.6 | Despesa fixa | 25 |
| 2.1.7 | Despesa variável | 25 |
| 2.1.8 | Ativos | 25 |
| 2.1.9 | Passivos | 26 |
| 2.2 | Gamificação | 26 |
| 2.2.1 | Jogo | 26 |
| 2.2.2 | Modelo octalysis | 26 |
| 2.2.3 | Significado Épico & Chamado | 27 |
| 2.2.4 | Desenvolvimento & Realização | 28 |
| 2.2.5 | Empoderamento & Feedback | 28 |
| 2.2.6 | Propriedade & Posse | 28 |
| 2.2.7 | Influência Social & Pertencimento | 28 |
| 2.2.8 | Escassez & Impaciência | 29 |
| 2.2.9 | Imprevisibilidade & Curiosidade | 29 |
| 2.2.10 | Perda & Rejeição | 29 |
| 2.2.11 | Técnicas de jogos | 29 |
| 2.2.12 | Narrativa | 30 |
| 2.2.13 | Almoço grátis | 31 |
| 2.2.14 | Pontuação | 31 |

| | | |
|------------|--|-----------|
| 2.2.15 | Barra de progresso | 31 |
| 2.2.16 | Lista de missões | 31 |
| 2.2.17 | Coroação | 31 |
| 2.2.18 | Tutorial | 31 |
| 2.2.19 | Percepção de escolha | 32 |
| 2.2.20 | Escolhas significativas | 32 |
| 2.2.21 | Pontos Trocáveis | 32 |
| 2.2.22 | Monitor de apego | 32 |
| 2.2.23 | Avatar | 32 |
| 2.2.24 | Mentoria | 32 |
| 2.2.25 | Fossa | 32 |
| 2.2.26 | Ovo de páscoa | 32 |
| 2.2.27 | Loteria | 33 |
| 2.2.28 | Tragédia de custo irre recuperável | 33 |
| 2.2.29 | Perda de progresso | 33 |
| 2.3 | Elicitação de requisitos | 33 |
| 2.3.1 | Definição de requisito | 33 |
| 2.3.2 | Requisito funcional | 33 |
| 2.3.3 | Requisito não funcional | 34 |
| 2.3.4 | Técnicas de elicitación de requisitos | 34 |
| 2.3.5 | Brainstorming | 34 |
| 2.3.6 | Entrevista | 34 |
| 2.3.7 | Introspecção | 35 |
| 2.3.8 | Prototipação | 35 |
| 2.4 | Priorização MoSCoW | 35 |
| 2.5 | Metodologias de desenvolvimento de software | 36 |
| 2.5.1 | Metodologia ágil | 36 |
| 2.5.2 | Scrum | 36 |
| 2.5.3 | Kanban | 38 |
| 2.5.4 | Programação extrema | 39 |
| 3 | PROJETO DO STUDY | 41 |
| 3.1 | Modelo <i>Octalysis</i> e elicitación de requisitos do <i>Study</i> | 41 |
| 3.2 | Significado Épico & Chamado | 43 |
| 3.2.1 | Narrativa | 43 |
| 3.2.2 | Almoço grátis | 43 |
| 3.3 | Desenvolvimento & Realização | 44 |
| 3.3.1 | Pontuação | 44 |
| 3.3.2 | Barra de progresso | 45 |
| 3.3.3 | Lista de missão | 45 |

| | | |
|-------------|---|-----------|
| 3.3.4 | Coroação | 47 |
| 3.3.5 | Tutorial | 47 |
| 3.4 | Empoderamento & Feedback | 48 |
| 3.4.1 | Percepção de escolha | 48 |
| 3.4.2 | Escolhas significativas | 49 |
| 3.5 | Propriedade & Posse | 50 |
| 3.5.1 | Pontos Trocáveis | 50 |
| 3.5.2 | Monitor de apego | 51 |
| 3.5.3 | Avatar | 51 |
| 3.6 | Influência Social & Pertencimento | 51 |
| 3.6.1 | Mentoria | 51 |
| 3.7 | Escassez & Impaciência | 52 |
| 3.7.1 | Fossa | 52 |
| 3.8 | Imprevisibilidade & Curiosidade | 52 |
| 3.8.1 | Ovo de páscoa | 52 |
| 3.8.2 | Loteria | 53 |
| 3.9 | Perda & Rejeição | 53 |
| 3.9.1 | Tragédia de custo irrecuperável | 53 |
| 3.9.2 | Perda de progresso | 53 |
| 3.10 | Resumo e priorização dos requisitos para o \$study | 54 |
| 4 | MODELAGEM E PROTOTIPAÇÃO DO STUDY | 57 |
| 4.1 | Prototipo | 57 |
| 4.1.1 | Identidade visual | 57 |
| 4.1.2 | Tela de login | 59 |
| 4.1.3 | Tela de cadastro | 60 |
| 4.1.4 | Tela do menu principal e tutorial | 61 |
| 4.1.5 | Tela da <i>dashboard</i> | 62 |
| 4.1.6 | Tela de investimento | 63 |
| 4.1.7 | Tela da loja virtual | 64 |
| 4.1.8 | Tela de notícias | 65 |
| 4.1.9 | Tela de perguntas e respostas | 66 |
| 4.1.10 | Tela das missões | 67 |
| 4.2 | Modelagem de dados | 68 |
| 4.2.1 | Modelo entidade-relacionamento | 68 |
| 4.3 | Relação entre investimento e fatores externos | 69 |
| 4.4 | Relação entre loja, modulo e usuário | 70 |
| 4.5 | Relação entre questões, missão e modulo | 71 |
| 4.6 | Relação entre Usuário e Investimento | 72 |
| 4.7 | Relação entre Usuário, Progresso monetário e Progresso educacional | 73 |

| | | |
|------------|---------------------------------------|-----------|
| 5 | IMPLEMENTAÇÃO | 75 |
| 5.1 | Arquitetura | 75 |
| 5.2 | Processo de <i>software</i> | 77 |
| 5.3 | Versão final do <i>\$study</i> | 79 |
| 5.3.1 | Tela de login | 79 |
| 5.3.2 | Tela de cadastro | 80 |
| 5.3.3 | Tela do menu principal e tutorial | 81 |
| 5.3.4 | Tela de investimento | 82 |
| 5.3.5 | Tela de perguntas e respostas | 83 |
| 5.3.6 | Tela das missões | 84 |
| 6 | CONCLUSÃO | 85 |
| 6.1 | Trabalhos futuros | 85 |
| | REFERÊNCIAS | 87 |

1 Introdução

Educação financeira tem se tornado uma questão importante para os diversos segmentos da sociedade. Tanto para investidores seniores quanto para jovens, permitindo tanto um melhor controle de gastos quanto uma rentabilidade extra (OCDE, 2005).

A gamificação tem sido utilizada para motivar e engajar usuários em atividades cotidianas. Um grande uso da gamificação tem sido na área de educação. Os resultados obtidos mostram-se interessantes (FARDO, 2013).

1.1 Motivação

Após uma pesquisa de mercado foi percebido uma oportunidade de preencher um espaço de mercado, visto que as aplicações presentes são pouco atrativas.

Neste contexto, este trabalho se propõe construir um sistema gamificado para educação financeira em molde semelhante ao *Duolingo*¹ que permita engajar os usuários na aprendizagem dos conceitos financeiros modelados.

1.2 Objetivo

Construir um sistema computacional gamificado, via web, para a educação financeira de usuários nas variadas faixas etárias.

1.3 Metodologia

Desenvolvimento de sistema, através da pesquisa bibliográfica sobre framework de gamificação, educação financeira, seguindo os modelos de desenvolvimento de software preconizados na área de engenharia de software.

¹ <https://pt.duolingo.com/>

1.4 Estrutura da monografia

No capítulo 2 é feita a revisão da literatura sobre educação financeira, gamificação e metodologias de desenvolvimento de software. No capítulo 3 são apresentados os requisitos do sistema *Study*, bem como a priorização dos mesmos. No capítulo 4, são apresentados o protótipo e a modelagem conceitual do *Study*. No capítulo 5 é apresentada a implementação do *Study* e sua arquitetura e por fim no capítulo 6 é apresentado a conclusão.

2 Revisão da literatura

Este capítulo apresenta uma revisão da literatura dos conceitos necessários para o entendimento do *Study*. Os conceitos abordados são conceitos da educação financeira, gamificação, eliciação de requisitos e metodologias de desenvolvimento de software.

2.1 Educação financeira

Planejamento financeiro é indispensável para propiciar estabilidade e promover o bem-estar tanto pessoal como familiar (MOREIRA, 2020).

2.1.1 Princípio da educação financeira

A educação financeira é um instrumento para a estabilidade e crescimento econômico sendo como investidores, por meio da informação e instrução podem aperfeiçoar seu entendimento a respeito de questões financeiras, construindo confiança para compreender riscos e oportunidades, gerando assim uma estabilidade e conseqüentemente um bem-estar financeiro (OCDE, 2005).

O foco da educação financeira são questões nacionais podendo abordar planejamento financeiro até noções de economia com a sua promoção e oferta de forma justa e imparcial para que seja difundida em todas as camadas de uma sociedade (OCDE, 2005).

2.1.2 Inflação

No Brasil para determinar as metas inflacionárias é utilizado o Índice Nacional de Preços ao Consumidor Amplo (IPCA), que consiste em um conjunto de itens de consumo básico, a taxa de inflação é calculada a partir da variação do custo por período. Então pode-se concluir que o termo "inflação" é o aumento dos preços de bens e serviços gerando assim uma diminuição no poder de compra de determinada moeda (BCB, 2022b).

A consequência dessa variação de preços é o desestímulo de investimento interno e externo, impactando o avanço econômico, causando assim uma instabilidade econômica (BCB, 2022b).

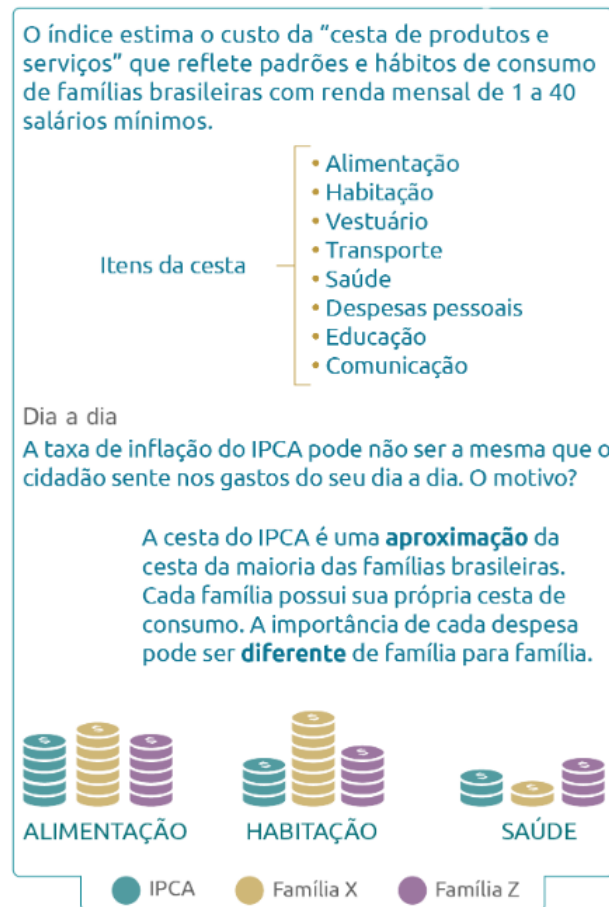


Figura 1 – Cálculo do IPCA (BCB, 2022b)

2.1.3 Juros

Juros é o lucro do valor em relação ao tempo (BCB, 2022a).

Um exemplo clássico de juros é a realização de empréstimo em uma instituição financeira, essa instituição empresta o dinheiro, porém por conta do serviço um valor será acrescido ao que deverá ser pago no final do empréstimo, esse valor acrescido são os juros.

2.1.3.1 Taxa de juros

A taxa de juros representa o quão rentável é um determinado investimento, tendo sua temporalidade podendo variar de anos até dias (STIELER, 2009).

2.1.3.2 Juros simples

Juros simples é uma forma de capitalização onde a taxa incide sobre o valor inicial e não é somado para um novo cálculo de juros (CAIUSCA, 2019).

2.1.3.3 Juros compostos

Juros compostos, ou juros sobre juros, é uma forma de capitalização onde a taxa é aplicada ao valor total do período anterior (PUCPR, 2021). Gerando assim maior lucro em relação ao regime de capitalização por juros simples.

2.1.4 Renda fixa

Renda fixa significa que ao final de um prazo determinado é recebido o lucro de um investimento, esse lucro é feito na forma de juros ou correção monetária (XP, 2022).

O investimento em renda fixa pode ser realizado através de títulos privados como o CDB (Certificado de Depósito Bancário) que seria basicamente emprestar dinheiro para instituições bancárias privadas e receber o pagamento desse empréstimo com juros, ou através de títulos públicos como o Tesouro Direto, o qual possui o mesmo funcionamento do CDB, porém empresta-se dinheiro ao governo (BRASIL, 2022).

2.1.5 Renda variável

Renda variável são aqueles investimentos nos quais o rendimento varia a depender do mercado financeiro podendo haver lucro ou não no decorrer do tempo.

Um exemplo de renda variável são Ações de empresas nacionais ou internacionais, o investimento em Ações nada mais é do que adquirir uma fração de determinada empresa. O valor dessas Ações é medido através da oferta e demanda (FORTES, 2021).

2.1.6 Despesa fixa

Despesas fixas são os gastos que variam pouquíssimo, possibilitando assim realizar uma previsão de gastos futuros. Um exemplo de despesa fixa é uma parcela de financiamento ou o aluguel de uma casa (GRAÇAS, 2021).

2.1.7 Despesa variável

Diferente de uma despesa fixa, a despesa variável é influenciada pela intensidade do consumo e caso não haja um controle financeiro ficam mais suscetíveis a saírem do controle. Um exemplo de despesa variável é o gasto com alimentação. As despesas variáveis também são mais suscetíveis a serem afetadas pela inflação (GRAÇAS, 2021).

2.1.8 Ativos

Um investimento é considerado um ativo quando ele pode ser convertido em dinheiro rapidamente, um exemplo é quando os acionistas de uma empresa possuem parti-

cipação nos lucros, essa parte nos lucros é chamado dividendo (RICO, 2021).

2.1.9 Passivos

Um investimento é considerado passivo quando ele não gera uma renda. Um exemplo de investimento passivo é a compra de um imóvel (RICO, 2021).

2.2 Gamificação

Segundo Domínguez et al. (2013) a definição de gamificação no contexto de um produto de software é a aplicação de elementos de jogos para gerar engajamento e incrementar a experiência do usuário. Exemplos desses elementos de jogos são a narrativa, pontuação, integração com a comunidade dentre outros (KIM, 2015).

2.2.1 Jogo

A definição conservadora para o que pode ser chamado jogo seria caracterizada pela existência de regras e competição ou disputa por resultados ou objetivos junto a interação humana (DETERDING et al., 2011).

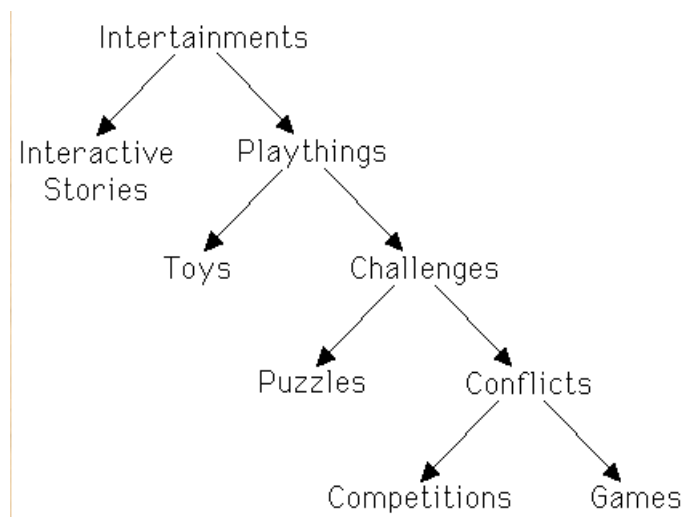


Figura 2 – Definição de jogo por Chris Crawford (CRAWFORD, 1991)

2.2.2 Modelo octalysis

Neste projeto para a construção do modelo de gamificação foi utilizado o modelo Octalysis que é um framework para a modelagem da gamificação.

O modelo Octalysis com a finalidade de engajar e motivar é dividido em 8 sentimentos principais, também chamados Core Drives. Os 8 Core Drives são:

1. Significado Épico & Chamado.
2. Desenvolvimento & Realização.
3. Empoderamento & Feedback.
4. Propriedade & Posse.
5. Influência Social & Pertencimento.
6. Escassez & Impaciência.
7. Imprevisibilidade & Curiosidade.
8. Perda & Rejeição.

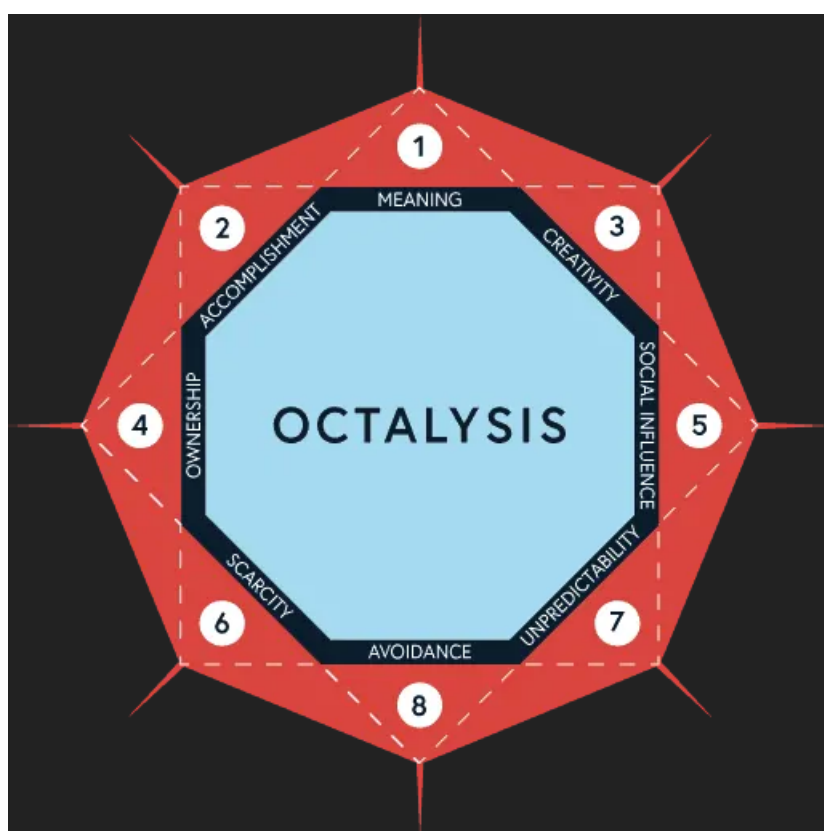


Figura 3 – Sistema Octalysis e seus Core Drives ([LOGICEARTH, 2022](#))

2.2.3 Significado Épico & Chamado

O primeiro Core Drive chamado Significado Épico e Chamado é ativado quando o jogador ou usuário acredita que ele está realizando algo que é maior até do que a sua própria existência ([CHOU, 2015](#)).

Um exemplo de Significado Épico seria realizar uma contribuição para um software livre como sistema operacional Linux, essa contribuição não é feita por dinheiro, mas sim por conta da pessoa que realizou a contribuição acreditar que está aperfeiçoando algo utilizado por milhões de pessoas no mundo.

Para a ativação deste Core Drive, neste projeto foram utilizadas as seguintes técnicas de jogos.

2.2.4 Desenvolvimento & Realização

O segundo Core Drive chamado Desenvolvimento e Realização é ativado quando o jogador ou usuário é movido por um sentimento de aperfeiçoamento e pela necessidade de realizar seus objetivos, e na maioria das vezes um desafio está associado a esse objetivo (CHOU, 2015).

Um exemplo de Desenvolvimento e Realização é o que a vitória representa para o um piloto do automobilismo, para quem é um espectador aquele troféu não tem valor, mas para o piloto é a realização de uma conquista e ele é motivado por isso.

2.2.5 Empoderamento & Feedback

O terceiro Core Drive chamado Empoderamento e Feedback é ativado quando o jogador ou usuário participa ativamente de um processo criativo, e essa participação se torna relevante (CHOU, 2015).

Um exemplo de Empoderamento e Feedback são brinquedos de montagem por blocos, os blocos podem ser montados a depender da criatividade da pessoa e essa criatividade se torna relevante quando algo é montado (CHOU, 2015).

2.2.6 Propriedade & Posse

O quarto Core Drive chamado Propriedade e Posse é ativado quando o jogador ou usuário sente que algo o pertence (CHOU, 2015).

Um exemplo de Propriedade e Posse são os recursos monetários ou de troca em que o usuário decide qual será a finalidade do recurso.

2.2.7 Influência Social & Pertencimento

O quinto Core Drive chamado Influência Social e Pertencimento é ativado quando o jogador ou usuário tem qualquer interação social seja ela pertencimento a uma comunidade ou até mesmo uma competição (CHOU, 2015).

Um exemplo de Influência Social e Pertencimento são os jogos de multijogador massivo, quando o jogador cria um grupo de amigos e eles realizam tarefas juntos e cooperam entre si.

2.2.8 Escassez & Impaciência

O sexto Core Drive chamado Escassez e Impaciência é ativado quando o jogador ou usuário encontra algo raro, ou especial que só pode ser adquirido após o cumprimento de um requisito (CHOU, 2015).

Um exemplo de Escassez e Impaciência em um jogo de exploração se o jogador encontrar um objeto raro, mas o qual só poderia ser adquirido por uma certa habilidade ainda não possuída certamente ele voltará para pega-lo.

2.2.9 Imprevisibilidade & Curiosidade

O sétimo Core Drive chamado Imprevisibilidade e Curiosidade é ativado quando o jogador ou usuário não pode prever os próximos acontecimentos, logo a sua atenção é focada para o imprevisível (CHOU, 2015).

Um exemplo de Imprevisibilidade e Curiosidade seriam os jogos de azar justamente por conta da imprevisibilidade e pela remota possibilidade de um grande prêmio.

2.2.10 Perda & Rejeição

O oitavo Core Drive chamado Perda e Rejeição é ativado quando o jogador ou usuário sente medo de perder um progresso ou uma oportunidade (CHOU, 2015).

Um exemplo de Perda e Rejeição são os jogos onde uma derrota resulta na perda de todos os pontos acumulados e esses pontos eram necessários para o progresso do jogador.

2.2.11 Técnicas de jogos

Segundo Chou (2015) cada *Core Drive* apresenta uma vasta quantidade de técnicas de jogos associadas a ele. A Tabela 1 lista todas as técnicas de jogos presentes no livro do Yu-kai Chou citado neste parágrafo. As técnicas utilizadas no *Study* foram destacadas em negrito.

Tabela 1 – Técnicas presentes no Octalysis (CHOU, 2015)

| Core Drive | Técnicas do Octalysis |
|---------------------------------------|--|
| 1 - Significado Épico & Chamado | Narrativa, Almoço grátis , Sorte de principiante, Elitismo, Herói humanitário, Grande significado, Criança predestinada, Cocriador |
| 2 - Desenvolvimento & Realização | Pontuação, Barra de progresso, Lista de quest, Coroação, Tutorial , Medalhas, Placar, Escolha brilhante, Oásis no deserto, Efeito <i>rockstar</i> , Recompensa por ação fixa, Premio vencedor, <i>High five</i> , Sinfonia do aprimoramento, Efeito de aura, Luta contra grande inimigo |
| 3 - Empoderamento & Feedback | Percepção de escolha, Escolhas significativas , Mecânicas perene, Cenoura e graveto, Controle em tempo real, Combos em cadeia, Comentário instantâneo, Preenchimento em branco, Autonomia voluntaria, Desbloqueio de marco, Reforço |
| 4 - Propriedade & Posse | Pontos Trocáveis, Monitor de apego, Avatar , Bens virtuais, Construir do zero, Conjunto de coleção, Efeito Alfred, Curva de aprendizado, Proteção, Recrutamento, Monitoramento |
| 5 - Influência Social & Pertencimento | Mentoria , Botões de vanglória, Prateleiras de troféus, Missão em grupo, Tesouros sociais, Orgulho social, Âncora de conformidade, bebedouro, Amizade, Colisão de gangorra, Divulgação, Vanglória, Economia de agradecimento |
| 6 - Escassez & Impaciência | Fossos , Dinâmica de consulta, Intervalos fixos, Oscilação, Justaposição ancorada, Estimulação de prêmio, Estimulação de opções, Feedback do paciente, Contagem regressiva, Aceleradores, Quebra de tortura, Interface de usuário envolvida |
| 7 - Imprevisibilidade & Curiosidade | Ovo de páscoa, Loteria , Recompensas repentinas, Narrativa visual, Maravilha óbvia, Recompensas rolan-tes, Quebra-cabeça de travessuras, Efeito oráculo, Mini missões, Escolha brilhante, Caixas misteriosas |
| 8 - Perda & Rejeição | Tragédia de custo irrecuperável, Perda de progresso , Legado legítimo, Oportunidade evanescente, Contagem regressiva, Preguiça de Status Quo, Medo de perder, Letra escarlata, Cova visual, Canção de choro |

2.2.12 Narrativa

A Narrativa faz parte do primeiro Core Drive do sistema Octalysis, a ideia central da Narrativa é propor ao jogador um contexto para o qual ele deveria jogar (CHOU, 2015). Seguindo esse conceito foi pensando para a aplicação um protagonista onde o próprio jogador o escolheria, esse protagonista é encarregado de contextualizar o jogador no decorrer da aplicação.

2.2.13 Almoço grátis

Almoço grátis também faz parte do primeiro Core Drive, o Almoço grátis consiste em dar brindes ao jogador para que o mesmo se sinta especial, esse brinde serve como um incentivo para o jogador continuar a jogar (CHOU, 2015). Para a aplicação foi pensado em uma recompensa em moedas por finalizar a contextualização por meio da Narrativa.

2.2.14 Pontuação

Pontuação faz parte do segundo Core Drive, a técnica de Pontuação é utilizada para manter o controle do progresso (CHOU, 2015). Na aplicação ela esta materializada em forma de moedas virtuais, a forma utilizada para mensurar o progresso seria no ganho ou perda das moedas.

2.2.15 Barra de progresso

A Barra de progresso é uma forma de medir o progresso no contexto (CHOU, 2015). Na aplicação o progresso é representado por meio do desempenho do usuário em relação aos seus investimentos

2.2.16 Lista de missões

A Lista de missões é utilizada para motivar através do desenvolvimento e conquista (CHOU, 2015). Tendo em vista esse conceito a Lista de missões é utilizada na aplicação para motivar por meio das recompensas em moedas ao concluir uma missão e o sentimento de conquista é ativado por meio da liberação de funcionalidades novas na aplicação.

2.2.17 Coroação

A Coroação é usada para recompensar o jogador por superar um grande desafio (CHOU, 2015). Na aplicação a Coroação é usada através de uma recompensa em moedas ao usuário caso ele complete todas as missões.

2.2.18 Tutorial

O Tutorial é utilizado como um passo a passo que orienta o jogador através das mecânicas (CHOU, 2015). Na aplicação não será diferente o usuário será guiado para que ele entenda as funcionalidades da aplicação.

2.2.19 Percepção de escolha

A Percepção de escolha é uma técnica que gera um engajamento justamente pelo fato de permitir ao jogador múltiplas opções que podem ser escolhidas (CHOU, 2015). Na aplicação essas escolhas serão por meio dos investimentos que podem ser adquiridos pelo usuário.

2.2.20 Escolhas significativas

Escolhas significativas como o próprio nome já explica se trata de escolhas que tem grande impacto e influenciam outros fatores (CHOU, 2015).

2.2.21 Pontos Trocáveis

A ideia da técnica Pontos Trocáveis é permitir ao jogador utilizar seus pontos de forma estratégica de modo a obter outros objetos de valor (CHOU, 2015). Na aplicação as formas de utilizar os pontos são através de compras na loja ou aplicações em investimentos.

2.2.22 Monitor de apego

A técnica Monitor de apego tem a finalidade de fazer com que o jogador se apegue e sinta-se dono de algo por conta da observação constante (CHOU, 2015).

2.2.23 Avatar

O Avatar tem a finalidade de criar um apego do jogador a aquele avatar fazendo o mesmo passar mais tempo jogando (CHOU, 2015).

2.2.24 Mentoria

A Mentoria tem a finalidade de manter a motivação do jogador através de orientações (CHOU, 2015).

2.2.25 Fossa

Fossa são oportunidades bloqueadas pelo fato do jogador não ter determinado pré-requisito ou habilidade (CHOU, 2015).

2.2.26 Ovo de páscoa

Ovo de páscoa é uma recompensa inesperada após um determinado acontecimento (CHOU, 2015).

2.2.27 Loteria

Loteria é técnica que permite recompensar alguns jogadores e outros não (CHOU, 2015).

2.2.28 Tragédia de custo irrecuperável

Tragédia de custo irrecuperável é uma técnica que causa ao jogador um sentimento de perda em potencial caso ele pare de jogar (CHOU, 2015).

2.2.29 Perda de progresso

Perda de progresso é uma técnica que permite reduzir o progresso do jogador (CHOU, 2015).

2.3 Elicitação de requisitos

Segundo a língua portuguesa "elicitación" vem do verbo "elicitar" que significa: "Causar o aparecimento de algo" ou "Fazer com que algo passe a existir" (DICIO, 2009). Considerando este conceito, na engenharia de software elicitación de requisitos é o processo de criação e compreensão dos requisitos de um software (VALENTE, 2020).

2.3.1 Definição de requisito

Requisito é a descrição da necessidade de um software ou sistema, essas necessidades podem ser referentes a uma funcionalidade que deve estar presente ou uma restrição de funcionamento (SOMMERVILLE, 2010).

Um exemplo de requisito seria uma aplicação possuir autenticação para que o usuário possa acessar sua conta pessoal em uma rede social.

2.3.2 Requisito funcional

O requisito funcional descreve a funcionalidade que deve estar presente no software, podendo também ser uma descrição do comportamento que o software necessita para o seu funcionamento (MALL, 2018).

Um exemplo de requisito funcional citado anteriormente, é o fato de uma rede social possuir autenticação, assim a autenticação presente na rede social é uma necessidade que deve ser atendida para o funcionamento da mesma.

2.3.3 Requisito não funcional

O requisito não funcional não está exatamente ligado a uma funcionalidade de um software e sim a ele como um todo. Requisito não funcional restringe o software ou sistema, podendo ser relacionado a segurança das informações do software, ou quão manutenível ele é (SOMMERVILLE, 2010).

Um exemplo de requisito não funcional é o fato de um software não exibir dados pessoais de seus usuários. Este requisito não é uma funcionalidade e sim uma restrição de como os dados pessoais devem ser tratados.

2.3.4 Técnicas de elicitação de requisitos

As técnicas de elicitação são as formas utilizadas para a descoberta e criação de requisitos funcionais e não funcionais, podendo ocorrer uma interação direta com o cliente para qual o software está sendo produzido. Cada técnica de elicitação apresenta uma abordagem diferente para a criação do requisito (WIEGERS; BEATTY, 2013).

Segundo Prado (2021) as técnicas de elicitação mais citadas na literatura são: Entrevista, Questionário, Prototipação, Casos de uso, Brainstorming, Cenário, Feedback, Workshop, Grupo focal e História de usuário.

Tendo isso em vista, as técnicas utilizadas para elicitar os requisitos presentes no *Study* foram, Brainstorming, Entrevista, Introspecção e Prototipação.

2.3.5 Brainstorming

Segundo o MIT (2022) Brainstorming é um processo para uma lista de ideias em resposta a um questionamento ou ideia. Utilizando essa definição no contexto da engenharia de software, brainstorming é uma lista de requisitos proveniente de uma discussão para suprir as necessidades do software para qual está sendo elicitado os requisitos (TIWARI; RATHORE, 2017).

Inicialmente todas as ideias de requisitos são aceitas e anotadas, após o término da discussão os requisitos anotados passam por uma análise e validação (TIWARI; RATHORE, 2017).

2.3.6 Entrevista

Entrevista é a mais tradicional técnica para elicitar requisitos, possibilitando o envolvimento direto não só do desenvolvedor de software, mas também das partes contratantes (WIEGERS; BEATTY, 2013).

A entrevista é uma conversa de preferência presencial entre desenvolvedor e cliente onde são expostos fatos e opiniões. Essa conversa pode ser realizada de duas maneiras diferentes (TIWARI; RATHORE, 2017):

- **Entrevista fechada:** Quem faz a entrevista já possui todos os questionamentos pré-definidos e busca por meio das respostas compreender as necessidades do software.
- **Entrevista aberta:** O desenvolvedor tenta descobrir as necessidades do software através de uma conversa aberta com o entrevistado, desse modo às duas partes participam de forma ativa no processo.

às duas formas podem ser mescladas durante o processo da entrevista.

2.3.7 Introspecção

O significado de introspecção no dicionário da língua portuguesa é: "Análise íntima e reflexiva que alguém faz sobre si mesmo." (DICIO, 2009). Trazendo esse conceito para a engenharia de software pode-se afirmar que a introspecção é uma análise íntima e reflexiva a respeito de um software.

A introspecção seria o mesmo que refletir sobre quais requisitos estariam presentes para que as necessidades do software fossem supridas (GOGUEN; LINDE, 1993).

2.3.8 Prototipação

Prototipação é uma representação visual das funcionalidades do software com a finalidade de validar e identificar requisitos (TIWARI; RATHORE, 2017).

Existem 2 categorias de protótipos:

- **Protótipo de baixa fidelidade:** Representação visual das funcionalidades construído de forma simples com a finalidade de compreender e descobrir requisitos.
- **Protótipo de alta fidelidade:** Representação visual das funcionalidades, construído de modo que simule a utilização do software para o cliente, com a finalidade de validar as funcionalidades presentes.

2.4 Priorização MoSCoW

Segundo Oliveira (2014) MoSCoW é uma maneira de classificar e priorizar requisitos para inclusão em um sistema de informação.

MoSCoW é dividido em quatro níveis de priorização sendo eles (OLIVEIRA, 2014):

- **Precisa ter (*Must*)**: Determina um requisito que deve ser completado para atender as necessidades do software
- **Deve ter (*Should*)**: Determina um requisito de alta prioridade
- **Poderia ter (*Could*)**: Determina um requisito que não é necessário para atender as necessidades do software, porém caso haja tempo será incluído a entrega final.
- **Não irá ser feito (*Will not have*)**: Determina um requisito que não será atendido.

2.5 Metodologias de desenvolvimento de software

No dicionário a palavra "metodologia" significa um conjunto de maneiras, ou métodos, de se realizar algo (CAMBRIDGE, 2022). Com base nesse conceito, pode-se entender que na engenharia de software uma metodologia de desenvolvimento de software é um conjunto de formas, ou maneiras, de gerenciar e desenvolver um projeto de software (YOUNG, 2013).

2.5.1 Metodologia ágil

Na década de 90 a insatisfação e ineficiência das metodologias utilizadas naquele momento levou a comunidade de desenvolvedores de software a criar métodos ágeis. Possibilitando a equipe de desenvolvimento ser responsável por desenvolver o software e não pela sua concepção. As metodologias ágeis possuem uma filosofia de entrega contínua com o foco de entregar novas funcionalidades para o cliente rapidamente (SOMMERVILLE, 2010).

2.5.2 Scrum

O foco do Scrum é a gerência de software, ele é dividido em três fases (SOMMERVILLE, 2010):

- **Planejamento**: Onde ocorre a definição dos objetivos do projeto e da arquitetura.
- **Ciclos ou *Sprints***: Ciclos são semanas de desenvolvimento e a cada ciclo é realizado o incremento do software e suas funcionalidades.
- **Finalização**: A finalização é onde é criado os documentos do software construídos. Por exemplo, manual do usuário ou quadro de ajuda.

Apos a equipe definir a duração das *sprints* é realizada uma lista de funcionalidades para a construção do projeto, para que as necessidades dele sejam atendidas, essa

lista é chamada *backlog* do produto. Após definido o *backlog* do produto a equipe decide quais funcionalidades serão entregues durante a *sprint*, cada dia da *sprint* é marcado por reuniões diárias onde é relatado o andamento das tarefas. Ao fim de uma *sprint* as novas mudanças no software são apresentadas ao cliente e também é realizado uma revisão da última semana de trabalho (SOMMERVILLE, 2010).

A Figura 4 ilustra o funcionamento do Scrum.

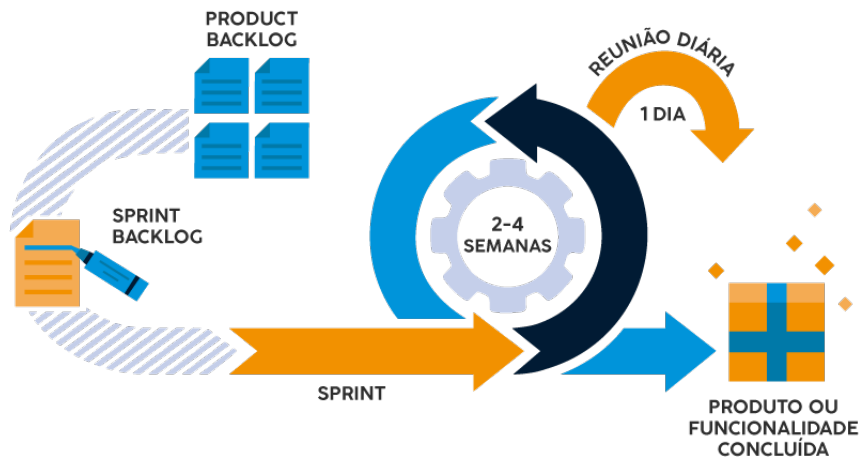


Figura 4 – Ciclo de trabalho do Scrum (TECNICON, 2019)

2.5.3 Kanban

O Kanban tem sua origem no Japão na década de 50, utilizado a primeira vez na fábrica de carros Toyota ele tem a finalidade de controlar o fluxo de produção (VALENTE, 2020).

A principal característica do Kanban são as colunas responsáveis por ilustrar e documentar o andamento de uma tarefa do projeto. A primeira coluna do Kanban seria a lista de tarefas disponíveis para desenvolvimento, seguido pela coluna de "trabalho em andamento", revisão, e conclusão (VALENTE, 2020).

As colunas do Kanbak são criadas para suprir e ilustrar as necessidades do fluxo de produção, como é mostrado na Figura 5.

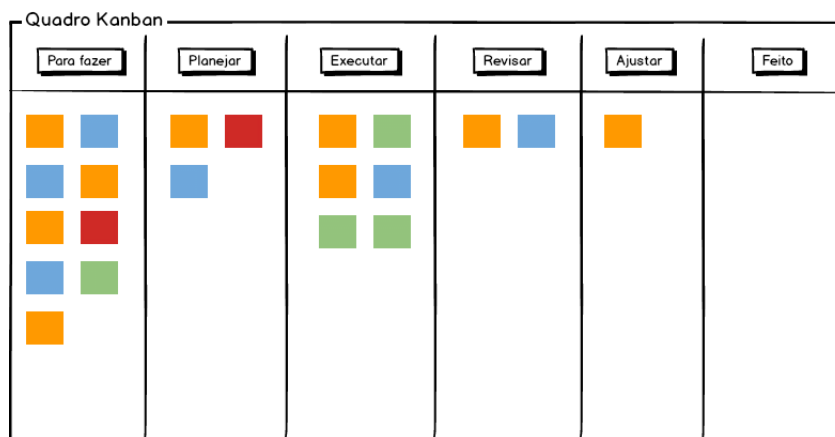


Figura 5 – Quadro do método Kanban (VALENTE, 2020)

2.5.4 Programação extrema

Programação extrema é uma metodologia ágil para o desenvolvimento de softwares com requisitos vagos e de constantes mudanças (VALENTE, 2020).

Para promover o nivelamento de habilidades a programação extrema realiza o pareamento de desenvolvedores com habilidades diferentes, os requisitos são retratados como cenários, esses cenários são chamados estórias do usuário e implementados como um conjunto de tarefas, cada tarefa completada deve ter a execução dos testes bem-sucedido (SOMMERVILLE, 2010).

A Figura 6 mostra o ciclo de uma entrega, chamada *release*, na programação extrema.

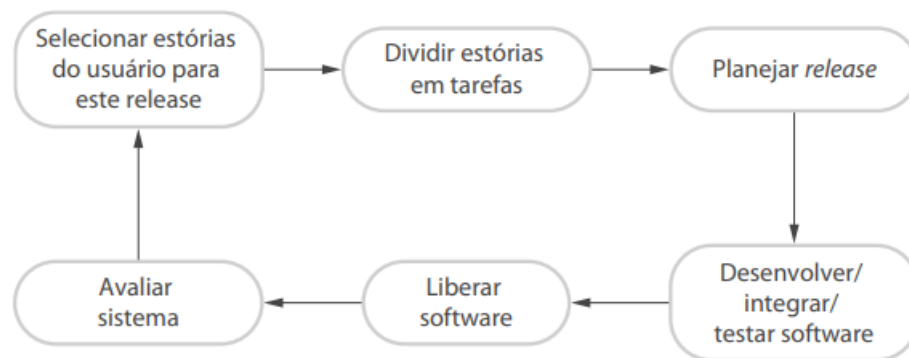


Figura 6 – Ciclo de uma entrega na programação extrema (SOMMERVILLE, 2010)

3 Projeto do *\$study*

O *\$study* é um software pensando para o aprendizado da educação financeira por meio de interações teóricas e práticas, de modo a ensinar e/ou aprimorar as habilidades do usuário nos conceitos da educação financeira. Para o usuário sentir-se engajado durante o uso do *\$study* é utilizado o modelo *Octalysis* de modo a implantar a gamificação na aplicação.

Este capítulo apresenta os requisitos elicitados para o *\$study* assim como a descrição de cada requisito elicitado.

3.1 Modelo *Octalysis* e elicitación de requisitos do *\$study*

Para realizar a gamificação do projeto foi utilizado o modelo *Octalysis* que é dividido em 8 *Core Drives* representados pela figura 3.

Referente a cada *Core Drive*, são utilizadas as técnicas de jogos para a implantação da gamificação na aplicação. Como é mostrado na tabela 2.

Tabela 2 – Técnicas por *Core Drive* (CHOU, 2015)

| Core Drive | Técnicas Utilizadas |
|---------------------------------------|--|
| 1 - Significado Épico & Chamado | Narrativa, Almoço grátis |
| 2 - Desenvolvimento & Realização | Pontuação, Barra de progresso, Lista de missão, Coroação, Tutorial |
| 3 - Empoderamento & Feedback | Percepção de escolha, Escolhas significativas |
| 4 - Propriedade & Posse | Pontos Trocáveis, Monitor de apego, Avatar |
| 5 - Influência Social & Pertencimento | Mentoria |
| 6 - Escassez & Impaciência | Fossos |
| 7 - Imprevisibilidade & Curiosidade | Ovo de páscoa, Loteria |
| 8 - Perda & Rejeição | Tragédia de custo irre recuperável, Perda de progresso |

A tabela 3 mostra como cada técnica de jogo utilizada é abrangida para que fosse implementada no *\$study*. Podendo assim ativar seus respectivos *Core Drives*.

Tabela 3 – Requisito por técnica

| Core Drive | Técnica | Requisito |
|------------|------------------------------------|--|
| 1 | Narrativa | Possuir tela de contextualização, Possuir protagonista na tela de contextualização |
| 1 | Almoço grátis | Ganhar pontos ao fim da narrativa |
| 2 | Pontuação | Possuir moeda virtual, Mostrar total de moedas |
| 2 | Barra de progresso | Possuir indicativo de rendimento atual, Detalhar rendimento |
| 2 | Lista de missão | Detalhar missão específica, Listar missões,Mostrar recompensa de conclusão, Escolher qual missão realizar, Desistir da missão |
| 2 | Coroação | Recompensar conclusão de todas as missões |
| 2 | Tutorial | Verificar se é a primeira utilização do App daquele usuário, Iniciar contextualização sobre a aplicação, Finalizar contextualização sobre a aplicação, Iniciar explicação das funcionalidades presentes na aplicação, Explicar todas as funcionalidades presentes na aplicação |
| 3 | Percepção de escolha | Possuir tela de investimentos, Selecionar investimento, Possuir níveis de risco para os investimentos, Escolher quantidade de moedas investidas, Cancelar investimento, Possuir lista com investimentos aplicados |
| 3 | Escolhas significativas | Possuir fatores externos lúdicos que afetem o rendimento dos investimentos, Calcular Acréscimo/Decréscimo dos pontos |
| 4 | Pontos Trocáveis | Possuir loja para troca de pontos, Possuir bônus de performance, Desbloquear outros investimentos por meio das moedas |
| 4 | Monitor de apego | Atualizar os rendimentos em tempo real |
| 4 | Avatar | Selecionar Avatar de perfil |
| 5 | Mentoria | Sugerir ao usuário a melhor abordagem para o novo investimento |
| 6 | Fossos | Bloquear investimentos/áreas que não cumpram os requisitos, Bloquear investimento que só é liberado por meio de compra na loja |
| 7 | Ovo de páscoa | Conceder bônus aleatório de moedas com base no desempenho dos rendimentos |
| 7 | Loteria | Noticiar fatos reais/irreais |
| 8 | Tragédia de custo irre recuperável | Fazer com que o usuário não possa pré determinar ações na aplicação |
| 8 | Perda de progresso | Calcular Acréscimo/Decréscimo dos pontos |

Com base nos *Core Drives* e nas técnicas escolhidas, foi feito o levantamento de requisitos para o *Study* apresentado nas seções a seguir.

3.2 Significado Épico & Chamado

Para este primeiro Core Drive foram escolhidas as técnicas de Narrativa e Almoço grátis.

3.2.1 Narrativa

A Narrativa faz parte do primeiro Core Drive do sistema Octalysis, a ideia central da Narrativa é propor ao jogador um contexto para o qual ele deveria jogar. Seguindo esse conceito foi pensando para a aplicação um protagonista no qual o próprio jogador o escolheria, esse protagonista é encarregado de contextualizar o jogador no decorrer da aplicação.

Requisito RF01: Possuir tela de contextualização

Para que o usuário possa ser contextualizado do objetivo da aplicação e venha a ter um sentimento de que ele faz parte de algo maior do que a sua existência, será criada uma tela réplica da tela inicial onde o usuário navega pela aplicação de forma roteirizada e recebe uma explicação do porquê da existência de suas funcionalidades, porém sem que seja possível a interação com a mesma. Essa contextualização fará parte da iteração inicial do usuário com a aplicação logo após a realização do login na aplicação.

O *back-end* será responsável pela verificação caso seja a primeira vez em que o usuário realiza o login na aplicação garantindo assim que a contextualização ocorra somente para usuários novatos e uma única vez, e o *front-end* fará a exposição dos textos que estarão contextualizando o usuário juntamente com o protagonista.

Requisito RF02: Possuir protagonista na tela de contextualização

Para que o usuário possa se identificar com o objetivo da aplicação, venha a criar um sentimento de empatia e projete a sua persona para a situação exposta, será exibido na tela de contextualização um protagonista que o usuário terá escolhido previamente onde esse personagem expõe a aplicação para o usuário por meio de textos predeterminados. Esse personagem fará parte da contextualização citada anteriormente.

O *back-end* será responsável por comunicar ao *front-end* qual protagonista foi escolhido pelo usuário e o *front-end* exibirá na tela o protagonista escolhido permitindo que o usuário visualize o mesmo durante toda a contextualização.

3.2.2 Almoço grátis

Almoço grátis também faz parte do primeiro Core Drive, o Almoço grátis consiste em dar brindes ao jogador para que o mesmo se sinta especial, esse brinde serve como um incentivo para o jogador continuar a jogar. Para a aplicação foi pensado em uma

recompensa em moedas por ter finalizado a contextualização por meio da Narrativa.

Requisito RF03: Ganhar pontos ao fim da narrativa

Para que o usuário se sinta motivado a continuar o aprendizado e garantir que ele tenha o necessário para começar a utilizar a aplicação e suas funcionalidades será garantido ao usuário uma quantidade de moedas adicionando essa quantidade de moedas a sua carteira esse acréscimo será realizado após o término da narrativa e tutorial.

O *back-end* será responsável por adicionar as moedas recebidas a carteira do usuário e armazenar esse novo valor ao banco de dados e o *front-end* será responsável por exibir a quantidade de moedas ganhas e o total de moedas que o usuário possui.

3.3 Desenvolvimento & Realização

Para o segundo Core Drive foram escolhidas as técnicas Pontuação, Barra de progresso, Lista de missão, Coroação e Tutorial.

3.3.1 Pontuação

Pontuação faz parte do segundo Core Drive, a técnica de Pontuação é utilizada para manter o controle do progresso. Dentro da aplicação ela esta materializada em forma de moedas virtuais, a forma utilizada para mensurar o progresso seria no ganho ou perda das moedas.

Requisito RF04: Possuir moeda virtual

Para que o usuário possa realizar investimentos ou compras na loja existirá uma moeda virtual para ser utilizada pelo usuário sempre que quiser essas moedas estarão sempre disponíveis para uso durante todo o tempo que o usuário estiver conectado.

O *back-end* será responsável por armazenar no banco de dados o total e realizar os cálculos de acréscimo e decréscimo das moedas, o *front-end* será responsável por exibir a quantidade de moedas ganhas ou perdidas e o total de moedas que o usuário possui.

Requisito RF05: Mostrar total de moedas

Para que o usuário possa ter sempre o controle do total de suas moedas e possa tomar decisões com base nisso será exibido na aba de status a quantidade total de moedas que o usuário possui os quais estarão sempre visíveis.

o *back-end* será responsável por fornecer a quantidade total de moedas para que o *front-end* possa exibir esse valor para o usuário.

3.3.2 Barra de progresso

A Barra de progresso é uma forma de medir o progresso dentro do contexto. Na aplicação o progresso é representado por meio do desempenho do usuário em relação aos seus investimentos.

Requisito RF06: Possuir indicativo de rendimento atual

Para que o usuário possa ter sempre o controle do seu rendimento geral e saber se está progredindo ou não será exibido na aba de status um resumo de todas as aplicações realizadas e indicará se o total dos investimentos foi rentável ou não, esse resumo estará sempre visível na barra de status.

O *back-end* será responsável por realizar o cálculo da rentabilidade total dos investimentos armazenar essa rentabilidade no banco de dados e disponibilizar o resultado para que o *front-end* possa exibir na barra de status.

Requisito RF08: Detalhar rendimento

Para que o usuário possa ter em detalhes todos os investimentos realizados e possa analisar quais foram as melhores decisões existirá uma tela que mostre com detalhes todos os investimentos que o usuário possui com seus respectivos valores, rendimentos, data da aquisição, valor inicial, valor atual etc. Essa tela poderá ser acessada pelo indicativo de rendimento atual.

O *back-end* será responsável por armazenar e realizar o cálculo da rentabilidade de todos os investimentos que o usuário possui o *front-end* será responsável por exibir todos os investimentos e seus detalhes em forma de lista possuindo um indicativo visual em caso de perda ou ganho de moedas.

3.3.3 Lista de missão

A Lista de missão é utilizada para motivar através do desenvolvimento e conquista. Tendo em vista esse conceito a Lista de missão é utilizada na aplicação para motivar por meio das recompensas em moedas ao concluir uma missão e o sentimento de conquista é ativado por meio da liberação de funcionalidades novas na aplicação.

Requisito RF10: Detalhar missão específica

Para que o usuário possa acompanhar seus estudos e saber quais serão as próximas missões, existirá uma tela que mostre a lista de todas as missões disponíveis e não disponíveis para o usuário realizar juntamente com suas recompensas e níveis de dificuldades além de exibir um botão para o início da missão, essa tela poderá ser acessada pela página inicial da aplicação ou pela barra de navegação.

O *back-end* será responsável por armazenar a lista de todas as missões, quais delas

foram concluídas, quais estão bloqueadas e suas recompensas, o *front-end* deverá exibir a lista de missões e um botão para iniciar a missão de acordo com os parâmetros passados pelo *back-end*.

Requisito RF09: Listar missões

Para que o usuário possa acompanhar o que ele irá aprender no decorrer da missão e quais são as recompensas para cada tarefa concluída ao selecionar uma missão será exibido todos os detalhes sobre ela e o que será tratado naquela missão como todas as tarefas e as recompensas por concluir uma das tarefas.

O *back-end* será responsável por armazenar todas as tarefas necessárias para se concluir a missão e suas recompensas, o *front-end* deverá exibir a lista de tarefas de acordo com os parâmetros passados pelo *back-end*.

Requisito RF12: Mostrar recompensa de conclusão

Para que o usuário possa sentir que está progredindo e sendo recompensado por isso, causando assim mais engajamento para com a aplicação, ao concluir uma missão uma pequena janela com um aviso parabenizando o usuário pela conquista aparecerá na tela essa janela aparecerá na frente da tela no momento da conclusão da missão, junto com as recompensas pela conclusão.

O *back-end* será responsável por registrar no banco de dados a conclusão da missão específica, além de salvar as recompensas, *front-end* deverá comunicar ao *back-end* a conclusão da missão além de ser responsável por exibir a janela de conclusão.

Requisito RF14: Escolher qual missão realizar

Para que o usuário tenha controle sobre o próprio progresso e decida como o seu estudo vai ser realizado na lista de missões haverá um botão para selecionar qual missão o usuário deseja realizar. Essa opção de iniciar a missão aparecerá sempre que o usuário selecionar qual missão ele deseja realizar.

O *back-end* será responsável por captar qual missão foi iniciada e enviar para o *front-end* a lista com o enunciado das perguntas juntamente com a resposta, o *front-end* deverá enviar para o *back-end* qual missão foi iniciada.

Requisito RF14: Desistir da missão

Para que o usuário tenha controle sobre o próprio progresso e decida por não realizar determinada missão que ele tenha começado durante a realização da missão haverá um botão para que o usuário possa desistir da mesma. A opção de desistir da missão aparecerá sempre que o usuário tenha iniciado uma missão.

O *front-end* deverá retornar o usuário para a tela de seleção de missões.

3.3.4 Coroação

A Coroação é usada como forma de recompensar o jogador por ter superado um grande desafio. Na aplicação a Coroação é usada por meio de uma recompensa em moedas ao usuário caso ele complete todas as missões.

Requisito RF15: Recompensar conclusão de todas as missões

Para que o usuário se sinta recompensado por seu empenho e seja motivado a continuar os estudos após a conclusão de todas as missões o ele será recompensado em forma de moedas. Ao final da última missão um aviso na tela dará os parabéns e mostrará a recompensa ganha pelo usuário.

O *front-end* deverá exibir na tela o aviso de que a última missão foi cumprida e o *back-end* adicionará a recompensa a carteira do usuário.

3.3.5 Tutorial

O Tutorial é utilizado como um passo a passo que orienta o jogador através das mecânicas. Na aplicação não será diferente o usuário será guiado para que ele entenda as funcionalidades da aplicação.

Requisito RF16: Verificar se é a primeira utilização da aplicação

Para que o usuário entenda as funcionalidades da aplicação e fique engajado nela, ao realizar o login, o sistema verificará se aquela conta já possui registro de entrada na aplicação, caso não ele será direcionado ao tutorial.

Após o *back-end* realizar a verificação de primeiro uso o *front-end* deverá iniciar o tutorial.

Requisito RF17: Iniciar contextualização sobre a aplicação

Para que a contextualização da aplicação se inicie é necessária a confirmação do sistema, o início do fluxo de tutorial começará pela contextualização através de um protagonista ao início do tutorial.

O *front-end* deverá iniciar a contextualização com o protagonista.

Requisito RF18: Finalizar contextualização sobre a aplicação

Será finalizada a contextualização para que se inicie a explicação das funcionalidades ao final da contextualização.

O *front-end* deverá finalizar a contextualização dando início ao tutorial explicando as funcionalidades e informar o *back-end* para que o mesmo realize as ações necessárias.

Requisito RF19: Iniciar explicação das funcionalidades presentes na aplicação

Após a contextualização, o tutorial começará a explicação das funcionalidades presentes na aplicação prosseguindo com o tutorial.

O *front-end* deverá iniciar o tutorial explicando as funcionalidades enquanto navega entre elas.

Requisito RF20: Explicar todas as funcionalidades presentes na aplicação

Para que o usuário possa fazer o uso da aplicação corretamente e tenha uma experiência completa será explicado, em todas as telas, o que cada funcionalidade realiza dentro da aplicação.

O *front-end* deverá navegar entre telas, destacar a funcionalidade que está sendo explicada e exibir os textos para o usuário. O *back-end* deverá registrar a recompensa ganha ao final do tutorial.

3.4 Empoderamento & Feedback

Para o terceiro Core Drive foram escolhidas as técnicas Percepção de escolha e Escolhas significativas.

3.4.1 Percepção de escolha

A Percepção de escolha é uma técnica que gera um engajamento justamente pelo fato de permitir ao jogador múltiplas opções que podem ser escolhidas. Na aplicação essas escolhas serão por meio dos investimentos que podem ser adquiridos pelo usuário.

Requisito RF21: Possuir tela de investimentos

Para que o usuário veja quais investimentos estão disponíveis juntamente com suas peculiaridades, haverá uma tela onde serão listados todos os investimentos disponíveis essa tela será uma das opções disponíveis na tela inicial.

O *back-end* envia para o *front-end* a lista de investimentos disponíveis junto com seu risco, valor caso tenha, rentabilidade e o *front-end* exibe de forma organizada essas informações.

Requisito RF22: Selecionar investimento

Para que o usuário possa escolher quais investimentos ele gostaria de aplicar suas moedas na tela de investimentos será possível selecionar o investimento desejado e aplicar moedas a ele.

O *front-end* comunica qual investimento foi selecionado pelo usuário e qual a quantidade de moedas aplicadas, o *back-end* registra essa aplicação no banco de dados.

Requisito RF23: Possuir níveis de risco para os investimentos

Para que exista um balanceamento entre rentabilidade e risco, na tela de investimentos haverá níveis de risco nos investimentos, representados por estrelas que variam entre 1 e 5 estrelas.

O *back-end* terá em seus registros o nível de risco do investimento que serão exibidos na tela pelo *front-end*.

Requisito RF24: Escolher quantidade de moedas investidas

Para que o usuário tenha controle sobre suas finanças, na tela de investimentos será possível escolher a quantidade de moedas aplicadas em um investimento.

O *front-end* enviará para o *back-end* a quantidade de moedas aplicadas e ele fará os cálculos necessários.

Requisito RF25: Cancelar investimento

Para que o usuário possa desistir ou repensar suas decisões, na tela de investimentos aplicados pelo usuário será possível cancelar uma aplicação realizada no mesmo dia do cancelamento.

O *front-end* mostrará uma janela para que seja confirmado o cancelamento, caso confirmado o *back-end* devolve para a carteira do usuário o valor gasto por ele naquele investimento.

Requisito RF26: Possuir lista com investimentos aplicados

Para que o usuário veja quais os investimentos aplicados e seus rendimentos, na barra de status haverá uma tela onde serão listados todos os investimentos aplicados pelo usuário.

O *back-end* envia para o *front-end* a lista de todos os investimentos aplicados pelo usuário e os dados de cada aplicação, o *front-end* mostra esses dados em forma de lista.

3.4.2 Escolhas significativas

Escolhas significativas como o próprio nome já explica se trata de escolhas que tem grande impacto e influenciam outros fatores.

Requisito RF27: Possuir fatores externos lúdicos que afetem o rendimento dos investimentos

Para causar uma sensação de que as ações do usuário têm peso e relevância, o rendimento dos investimentos é afetado positivamente ou negativamente por fatores lúdicos externos no momento em que um investimento é escolhido.

No *back-end* existe uma lista de eventos que ocorrem de maneira desordenada

em determinado espaço de tempo, determinado quais eventos ocorrem depois é feito o cálculo dos rendimentos, o resultado desse cálculo é enviado para o *front-end* onde é feita a exibição dos ganhos ou perdas.

Requisito RF28: Calcular Acréscimo/Decréscimo dos pontos

Para ter uma referência do progresso do usuário é realizado o cálculo para o aumento ou diminuição das moedas do usuário por conta de um evento em seus investimentos ou por compras feitas na loja.

A depender do evento acionado o valor a ser adicionado ou retirado é enviado para o *back-end* onde o mesmo realiza o cálculo e envia para o *front-end* o resultado. O *front-end* é responsável por exibir o resultado do cálculo.

3.5 Propriedade & Posse

Para o quarto Core Drive foram escolhidas as técnicas Pontos Trocáveis, Monitor de apego e Avatar.

3.5.1 Pontos Trocáveis

A ideia da técnica Pontos Trocáveis é permitir ao jogador utilizar seus pontos de forma estratégica de modo a obter outros objetos de valor. Dentro da aplicação as formas de utilizar os pontos são por meio de compras na loja ou aplicações em investimentos.

Requisito RF29: Possuir loja para troca de pontos

Para ter uma forma de utilizar estrategicamente as moedas possuídas, na tela inicial existe uma loja onde o usuário utiliza as moedas em troca de vantagens ou novas opções de investimento.

O *front-end* é responsável por realizar a transição da tela e listar as opções da loja. O *back-end* é responsável por enviar a lista de itens da loja junto com seus preços e requisitos para desbloqueio.

Requisito RF30: Possuir bônus de performance

Para que o usuário monte uma estratégia com o bônus e seus investimentos potencializando seus ganhos, existe um bônus de performance para ser comprado na loja que adiciona um multiplicador às moedas adquiridas.

O *front-end* é responsável por listar o preço, o multiplicador do bônus e a duração do efeito. O *back-end* é responsável por registrar o efeito e duração do bônus.

Requisito RF31: Desbloquear outros investimentos por meio das moedas

Para o usuário aumentar a variedade de investimentos disponíveis existem inves-

timentos para serem desbloqueados na loja que são adicionados à lista de investimentos disponíveis.

O *front-end* é responsável por listar o preço, requisito para desbloqueio e nível de risco do investimento. O *back-end* é responsável por registrar o investimento comprado na lista total dos investimentos.

3.5.2 Monitor de apego

A técnica Monitor de apego tem a finalidade de fazer com que o jogador se apegue e sinta-se dono de algo por conta da observação constante.

Requisito RF32: Atualizar os rendimentos em tempo real

Para o usuário estar sempre atento ao rendimento de suas moedas, o indicador de rendimento na barra de status e investimentos adquiridos pelo usuário, mudam em tempo real.

O *front-end* é responsável por atualizar a tela sempre que haver mudanças nos rendimentos. O *back-end* é responsável por enviar os dados dos rendimentos.

3.5.3 Avatar

O Avatar tem a finalidade de criar um apego do jogador a aquele avatar fazendo o mesmo passar mais tempo jogando.

Requisito RF37: Selecionar Avatar de perfil

Para o usuário personalize a sua identidade, existe uma lista de avatares disponíveis para escolha na tela de perfil.

O *front-end* é responsável por carregar a lista de avatares, exibir para o usuário e enviar ao *back-end* qual deles foi selecionado. O *back-end* é responsável por enviar a lista de avatares existentes e armazenar o avatar selecionado.

3.6 Influência Social & Pertencimento

Para o quinto Core Drive foi escolhida a técnica de Mentoria.

3.6.1 Mentoria

A Mentoria tem a finalidade de manter a motivação do jogador.

Requisito RF33: Sugerir ao usuário a melhor abordagem para o novo investimento

Para o usuário utilizar com eficiência máxima a capacidade do investimento, na tela de seleção de investimentos sugestões com as melhores abordagens para determinado investimento são disponibilizadas.

O *front-end* é responsável por exibir na tela as sugestões caso elas existam.

3.7 Escassez & Impaciência

Para o sexto Core Drive foi escolhida a técnica de Fossa.

3.7.1 Fossa

Fossa são oportunidades que são bloqueadas pelo fato do jogador não ter determinado pré-requisito ou habilidade.

Requisito RF34: Bloquear investimentos/áreas que não cumpram os requisitos

Para motivar o usuário a continuar os estudos, Na tela de seleção de investimentos ou na loja é feito o bloqueio de investimentos/áreas caso tenha um pré-requisito não cumprido.

O *front-end* inviabiliza o acesso por meio de um atributo enviado pelo *back-end*.

Requisito RF35: Bloquear investimento que só é liberado por meio de compra na loja

Para motivar o usuário a ganhar e gastar as moedas, na loja é feito o bloqueio de investimentos que são obtidos por compra na loja.

O *front-end* inviabiliza o acesso por meio de um atributo enviado pelo *back-end*.

3.8 Imprevisibilidade & Curiosidade

Para o sétimo Core Drive foram escolhidas as técnicas de Ovos de páscoa e Loteria.

3.8.1 Ovo de páscoa

Ovo de páscoa é uma recompensa inesperada após um determinado acontecimento.

Requisito RF36: Conceder bônus aleatório de moedas com base no desempenho dos rendimentos

Para o usuário sentir que está sendo recompensado pelo seu esforço é concedido ao usuário um bônus de moedas de forma inesperada.

O *front-end* exibe na tela um aviso sobre esse bônus, envia para o *back-end* a quantidade de moedas acrescentadas e o *back-end* registra a nova quantidade de moedas.

3.8.2 Loteria

Loteria é técnica que permite recompensar alguns jogadores porém outros não.

Requisito RF38: Noticiar fatos reais/irreais

Para o usuário sentir que ele pode ou não ser afetado por algo que ele não tem controle na tela de notícias é exibido uma lista de notícias reais ou irreais.

O *back-end* envia uma lista com as notícias e seus efeitos para o *front-end* que é responsável por exibir essa lista de notícias.

3.9 Perda & Rejeição

Para o oitavo Core Drive foram escolhidas as técnicas de Tragédia de custo irre recuperável e Perda de progresso.

3.9.1 Tragédia de custo irre recuperável

Tragédia de custo irre recuperável é uma técnica que causa ao jogador um sentimento de perda em potencial caso ele pare de jogar.

Requisito RNF01: Fazer com que o usuário não possa pré-determinar ações na aplicação

Para o usuário sentir que caso saia da aplicação possa perder uma oportunidade em potencial não é permitido realizar comandos pré-programados.

3.9.2 Perda de progresso

Perda de progresso é uma técnica que permite reduzir o progresso do jogador.

Requisito RF28: Calcular Decréscimo dos pontos

Requisito explicado anteriormente.

3.10 Resumo e priorização dos requisitos para o *Study*

A tabela 4 mostra os requisitos funcionais e não funcionais elicitados para o *Study* além da priorização dos requisitos considerando os *Core Drives* presentes no modelo *Octalysis*.

Tabela 4 – Requisitos Funcionais/Não Funcionais

| Id | Requisito | MoSCoW |
|-----------|--|---------------|
| RF01 | Possuir tela de contextualização | <i>MUST</i> |
| RF02 | Possuir protagonista na tela de contextualização | <i>MUST</i> |
| RF03 | Ganhar pontos ao fim da narrativa | <i>COULD</i> |
| RF04 | Possuir moeda virtual | <i>MUST</i> |
| RF05 | Mostrar total de moedas | <i>MUST</i> |
| RF06 | Possuir indicativo de rendimento atual | <i>MUST</i> |
| RF07 | Possuir indicativo colorido relacionado ao rendimento | <i>MUST</i> |
| RF08 | Detalhar rendimento | <i>MUST</i> |
| RF09 | Listar missões | <i>MUST</i> |
| RF10 | Detalhar missão específica | <i>MUST</i> |
| RF11 | Detalhar rendimento | <i>MUST</i> |
| RF12 | Mostrar recompensa de conclusão | <i>MUST</i> |
| RF13 | Escolher qual missão realizar | <i>MUST</i> |
| RF14 | Desistir da missão | <i>MUST</i> |
| RF15 | Recompensar conclusão de todas as missões | |
| RF16 | Verificar se é a primeira utilização da aplicação | <i>MUST</i> |
| RF17 | Iniciar contextualização sobre a aplicação | <i>MUST</i> |
| RF18 | Finalizar contextualização sobre a aplicação | <i>MUST</i> |
| RF19 | Iniciar explicação das funcionalidades presentes na aplicação | <i>MUST</i> |
| RF20 | Explicar todas as funcionalidades presentes na aplicação | <i>MUST</i> |
| RF21 | Possuir tela de investimentos | <i>MUST</i> |
| RF22 | Selecionar investimento | <i>MUST</i> |
| RF23 | Possuir níveis de risco para os investimentos | <i>MUST</i> |
| RF24 | Escolher quantidade de moedas investidas | <i>MUST</i> |
| RF25 | Cancelar investimento | <i>MUST</i> |
| RF26 | Possuir lista com investimentos aplicados | <i>MUST</i> |
| RF27 | Possuir fatores externos lúdicos que afetem o rendimento dos investimentos | <i>MUST</i> |
| RF28 | Calcular Acréscimo/Decréscimo dos pontos | <i>MUST</i> |
| RF29 | Possuir loja para troca de pontos | <i>MUST</i> |

| | | |
|-------|---|---------------|
| RF30 | Possuir bônus de performance | <i>MUST</i> |
| RF31 | Desbloquear outros investimentos por meio das moedas | <i>MUST</i> |
| RF32 | Atualizar os rendimentos em tempo real | <i>COULD</i> |
| RF33 | Sugerir ao usuário a melhor abordagem para o novo investimento | <i>SHOULD</i> |
| RF34 | Bloquear investimentos/áreas que não cumpram os requisitos | <i>MUST</i> |
| RF35 | Bloquear investimento que só é liberado por meio de compra na loja | <i>SHOULD</i> |
| RF36 | Conceder bônus aleatório de moedas com base no desempenho dos rendimentos | <i>COULD</i> |
| RF37 | Selecionar Avatar de perfil | <i>COULD</i> |
| RF38 | Noticiar fatos reais/irreais | <i>COULD</i> |
| RF39 | Realizar revisões de investimento | <i>COULD</i> |
| RF40 | Possuir perfil de usuário instrutor | <i>MUST</i> |
| RF41 | Possuir um sistema de login | <i>MUST</i> |
| RF42 | Possuir um sistema de cadastro | <i>MUST</i> |
| RNF01 | Fazer com que o usuário não possa pré determinar ações na aplicação | <i>COULD</i> |

4 Modelagem de dados e prototipação do *Study*

Neste capítulo é apresentado o resultado da modelagem de dados bem como suas entidades e atributos e também a prototipação do *Study* e sua identidade visual.

4.1 Prototipo

4.1.1 Identidade visual

Uma paleta de cores é o conjunto de cores utilizadas harmonicamente para criar uma identidade visual (IED, 2021).

A Figura 7 representa a paleta de cores utilizada para a criação da identidade visual do *Study*.



Figura 7 – Paleta de cores do *Study*

- Verde: É a cor predominante utilizado para os botões, divisões e logo.
- Amarelo: É utilizado para representar a questão monetária.
- Cinza: Cor de fundo com a função de criar um contraste entre as cores.
- Vermelho: Para representar o risco dos investimentos e a desvalorização de um investimento.
- Violeta escuro: É utilizado para os textos.

Depois de decidido a paleta de cores foi criada os elementos que compõem o *Study*.

A Figura 8 apresenta a identidade visual do *Study* bem como sua logo, padrões dos botões, menu superior, menu inferior, caixas de texto e planos de fundo utilizados.

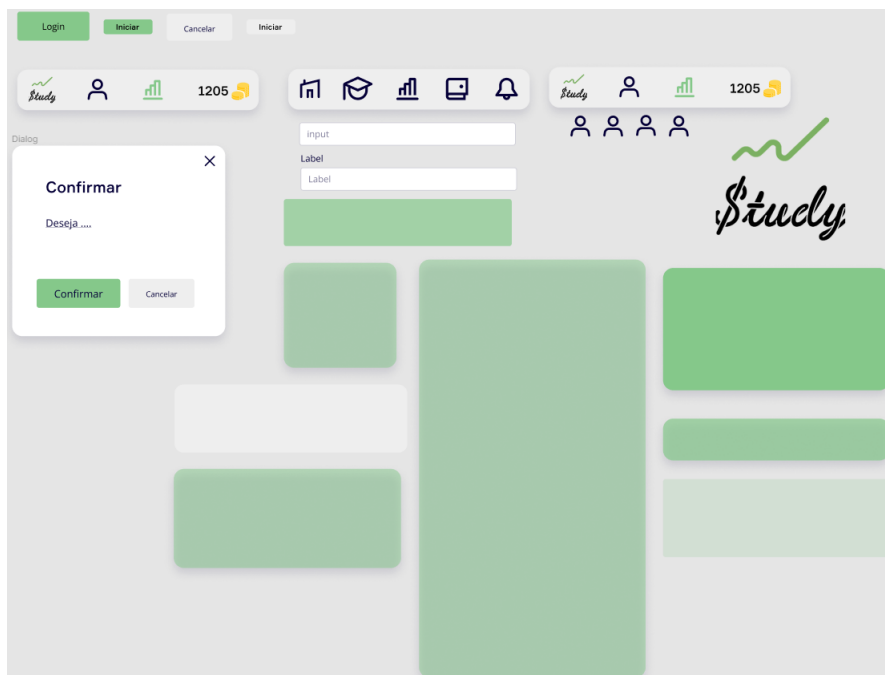


Figura 8 – Identidade visual do *Study*

4.1.2 Tela de login

A Figura 9 é o protótipo da tela de login, utilizada para autenticação do usuário no *Study*. O requisito associado a essa tela é o **RF41**



Figura 9 – Tela de login

4.1.3 Tela de cadastro

A Figura 10 é o protótipo da tela de cadastro, utilizada para o registro do usuário no *Study*. O requisito associado a essa tela é o **RF42**



Figura 10 – Tela de cadastro

4.1.4 Tela do menu principal e tutorial

A Figura 11 é o protótipo da tela de menu principal e tutorial, utilizada para a contextualização, navegação, e explicação das funcionalidades para o usuário no *Study*. Os requisitos associados a essa tela são os: **RF01**, **RF02**, **RF03**, **RF16**, **RF17**, **RF18**, **RF19**, **RF20**

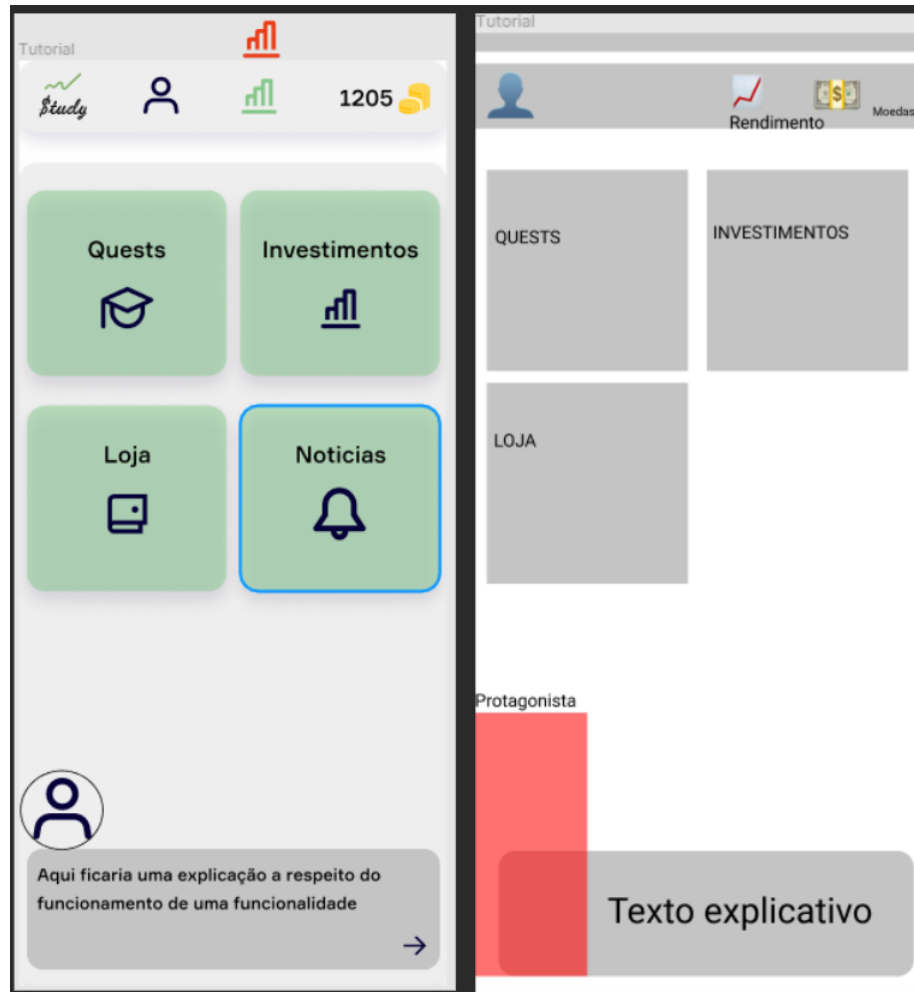


Figura 11 – Tela do menu principal e tutoria

4.1.5 Tela da *dashboard*

A Figura 12 é o protótipo da tela da *dashboard* utilizada para informar de forma visual os rendimentos do usuário no *Study*. Os requisitos associados a essa tela são os: **RF04, RF05, RF06, RF07, RF08, RF25, RF26, RF36, RF32, RNF01.**

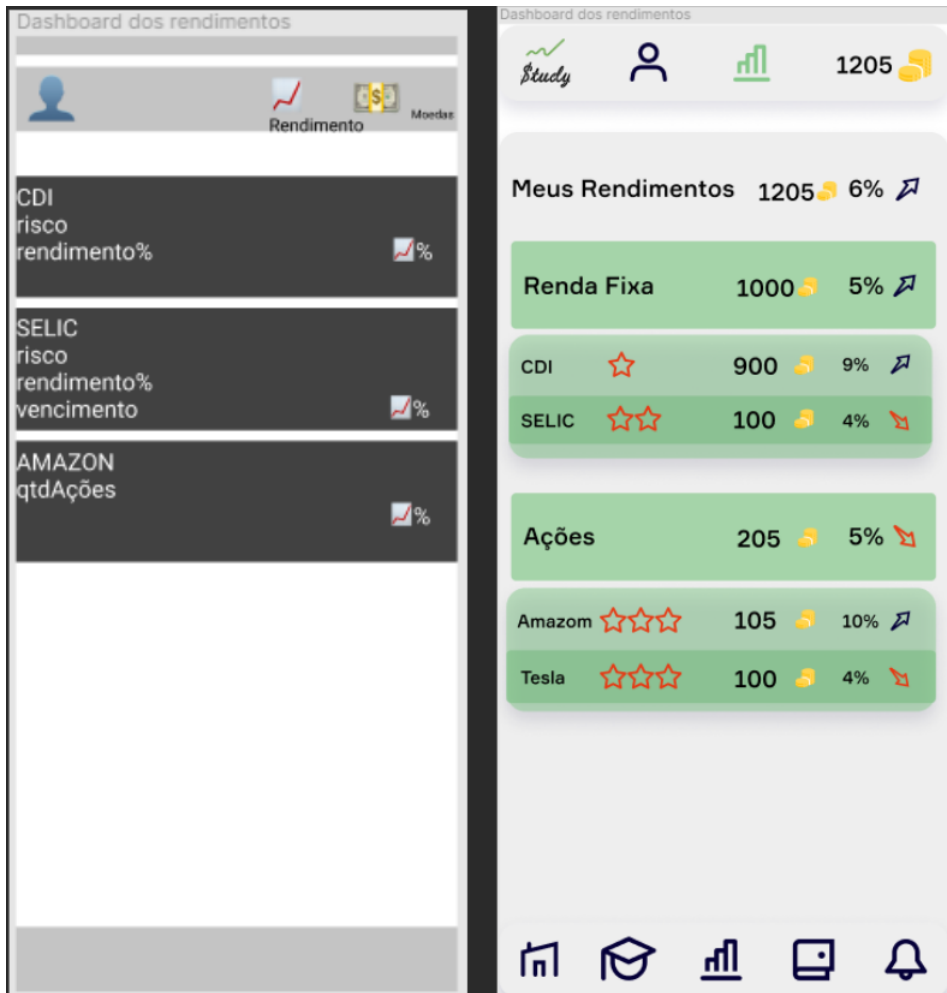


Figura 12 – Tela de *dashboard*

4.1.6 Tela de investimento

A Figura 13 é o protótipo da tela de investimento utilizada para informar quais o preço e risco de um investimento e também para adquirir um novo investimento para o usuário no *Study*. Os requisitos associados a essa tela são os: **RF21, RF22, RF23, RF24, RF33, RF34, RF35.**

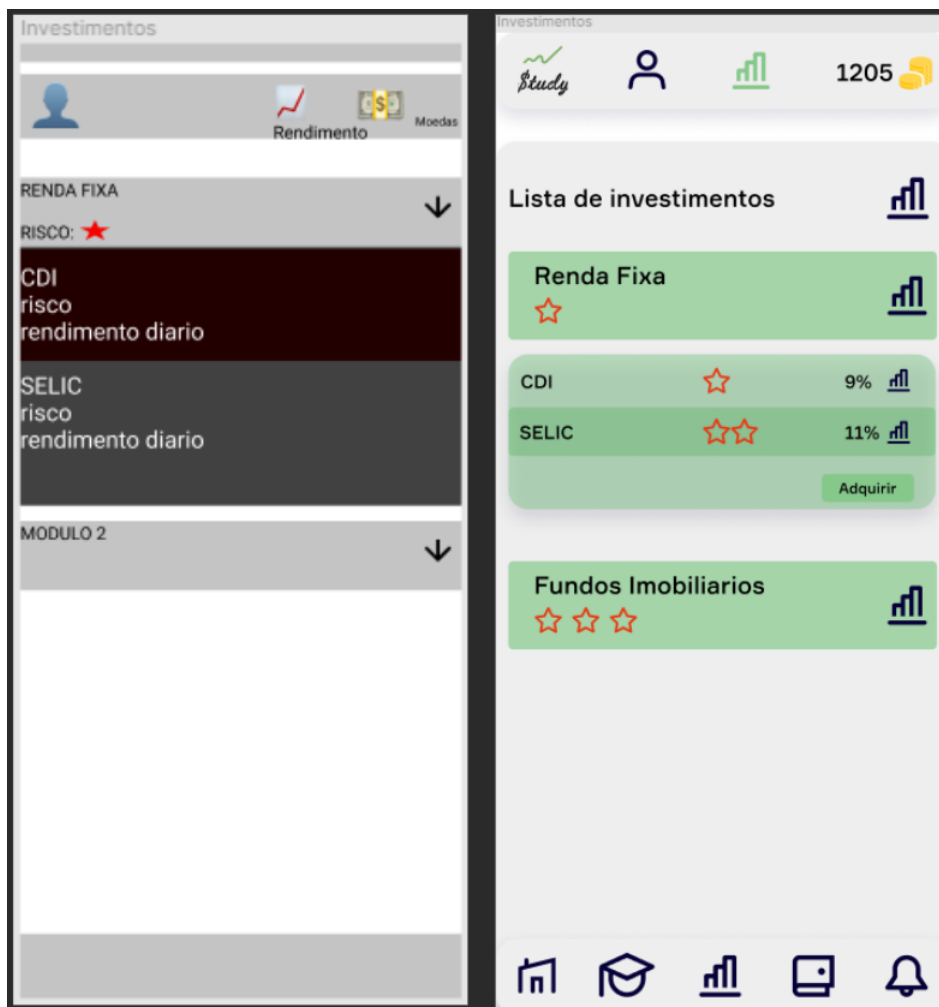


Figura 13 – Tela de investimento

4.1.7 Tela da loja virtual

A Figura 14 é o protótipo da tela da loja virtual utilizada para que o usuário possa liberar um novo investimento ou adquirir um bônus de moedas. Os requisitos associados a essa tela são os: **RF29**, **RF30**, **RF31**.

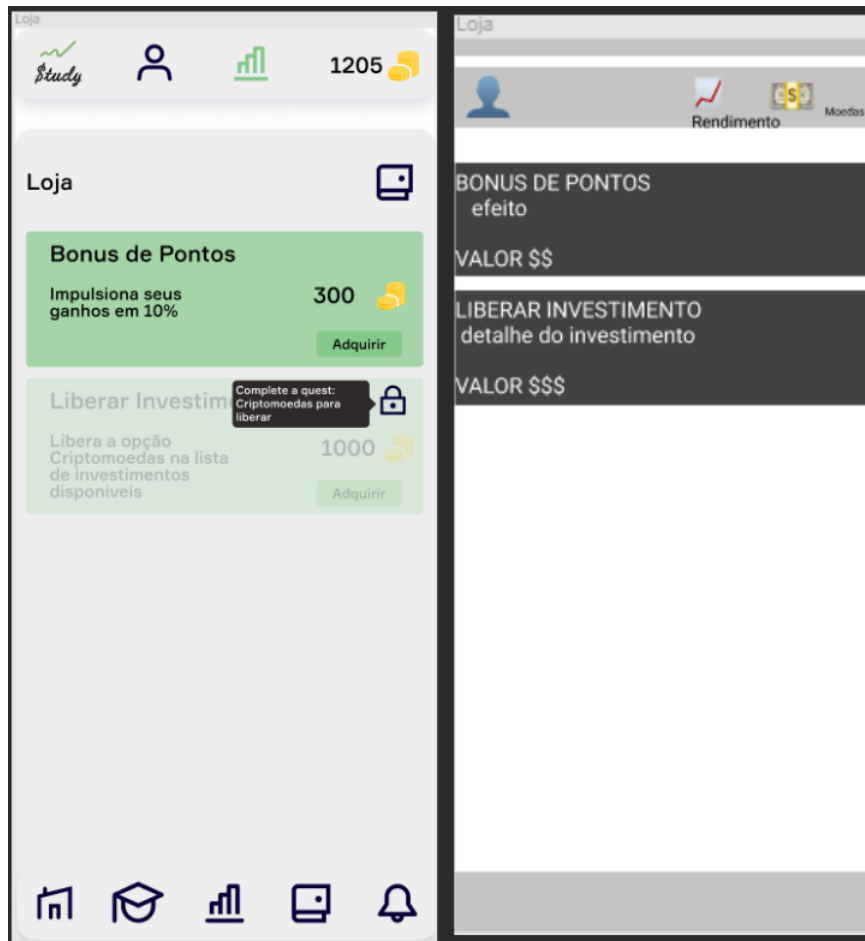


Figura 14 – Tela da loja virtual

4.1.8 Tela de notícias

A Figura 15 é o protótipo da tela de notícias utilizada para que o usuário possa acompanhar notícias reais e/ou irreais sobre eventos que possam impactar seus investimentos. Os requisitos associados a essa tela são os: **RF38**, **RF27**, **RF33**.



Figura 15 – Tela de notícias

4.1.9 Tela de perguntas e respostas

A Figura 16 é o protótipo da tela de perguntas e respostas utilizada para que o usuário possa realizar as missões solicitadas na aplicação e também realizar seus estudos a respeito da educação financeira. Os requisitos associados a essa tela são os: **RF13, RF15**.

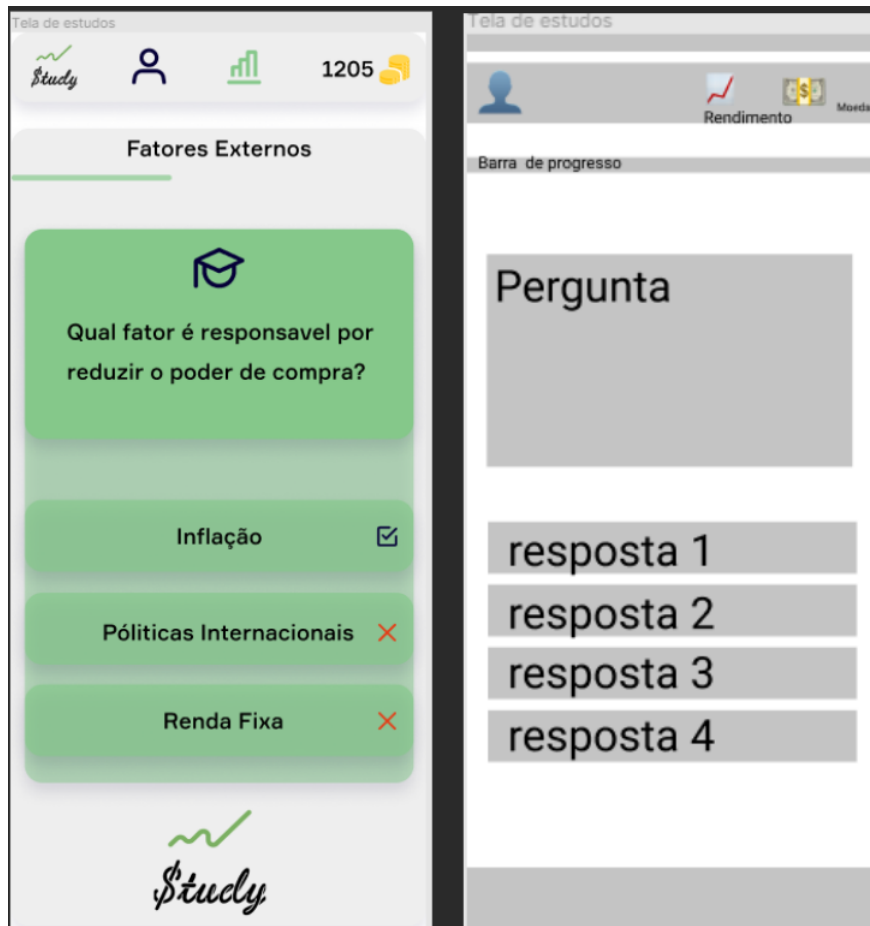


Figura 16 – Tela de perguntas e respostas

4.1.10 Tela das missões

A Figura 17 é o protótipo da Tela das missões utilizada para que o usuário possa selecionar, visualizar, e desistir das missões propostas no *%study*. Os requisitos associados a essa tela são os: **RF09, RF10, RF12, RF13, RF14, RF15**.

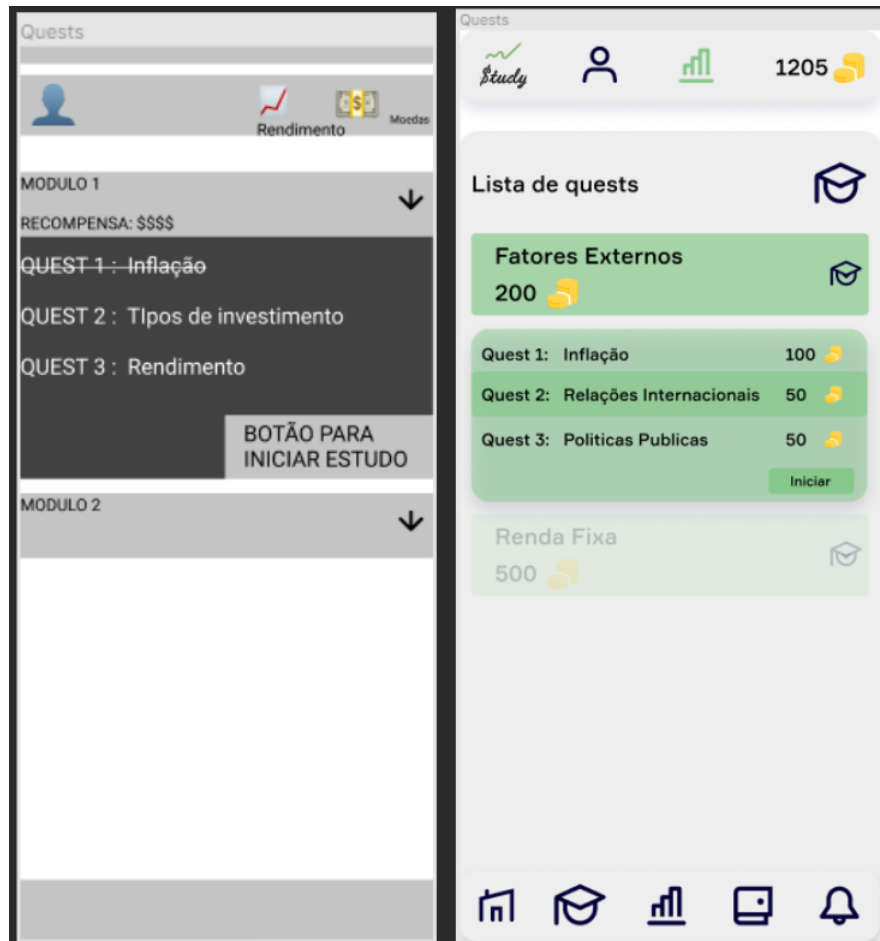


Figura 17 – Tela das missões

4.2 Modelagem de dados

4.2.1 Modelo entidade-relacionamento

A figura 18 representa o modelo entidade-relacionamento geral bem como as entidades, atributos e relacionamentos necessários para atender as necessidades do *Study* e seus requisitos.

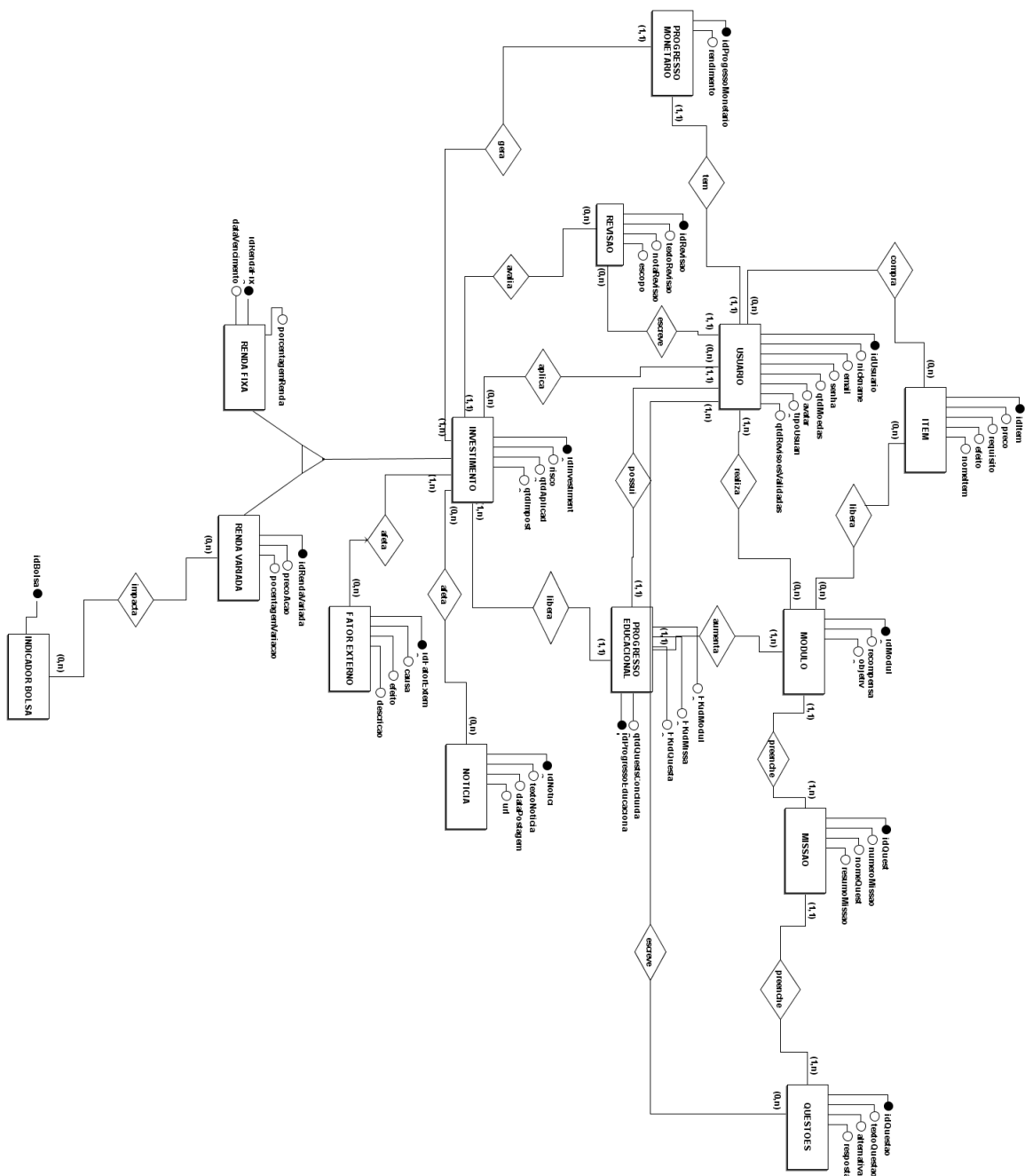


Figura 18 – Modelo entidade-relacionamento do *Study*

4.3 Relação entre investimento e fatores externos

Os investimentos são especializados em Renda fixa ou Renda variada, podendo ser afetados por Notícias e/ou Fatores externos. No caso de um investimento ser Renda Variada ele é impactado pelo Indicador da bolsa.

A figura 19 representa a relação dos Investimentos com os fatores externos.

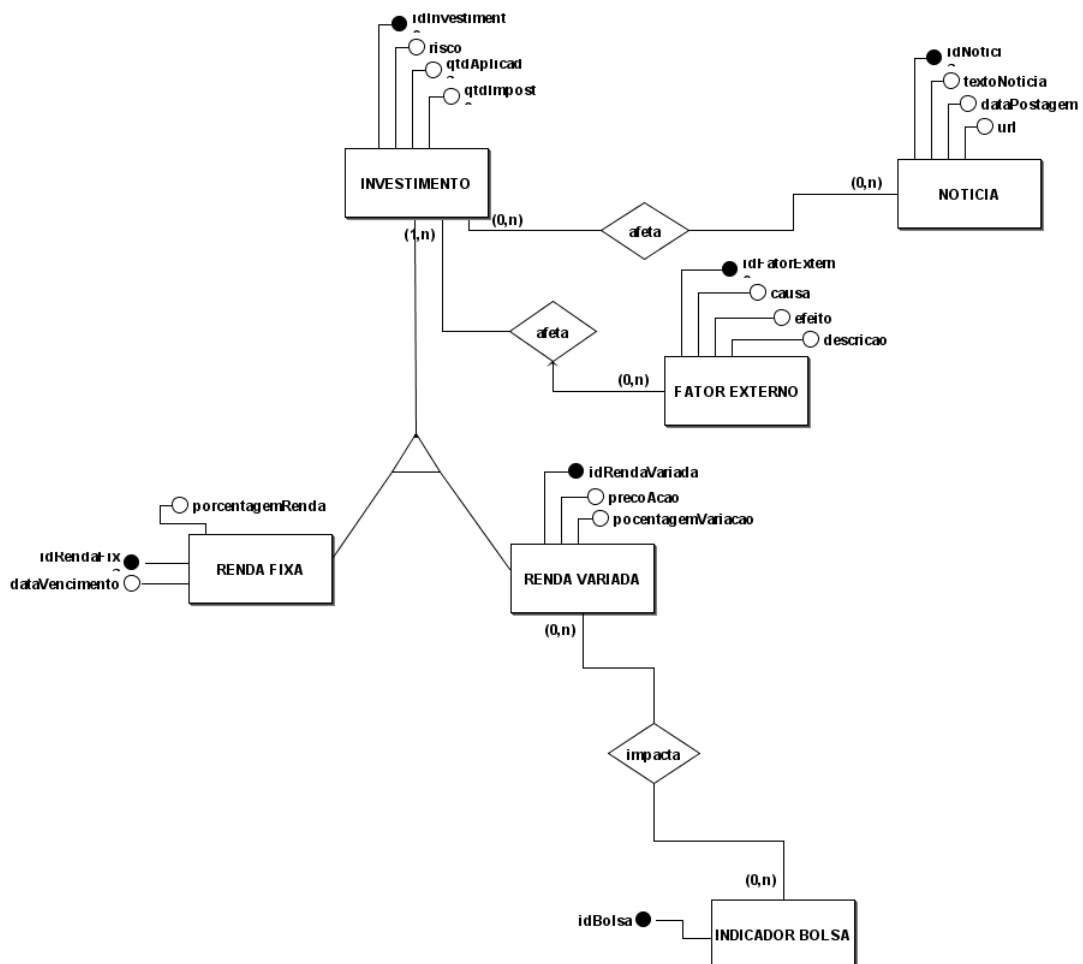


Figura 19 – Relação dos Investimentos com fatores externos

4.4 Relação entre loja, modulo e usuário

O Usuário realiza um Modulo de missões e esse Modulo realizado pode liberar um ou vários itens podendo ser comprado com moedas pelo Usuário.

A figura 20 mostra a relação entre loja e seus itens, modulo das missões e usuário.

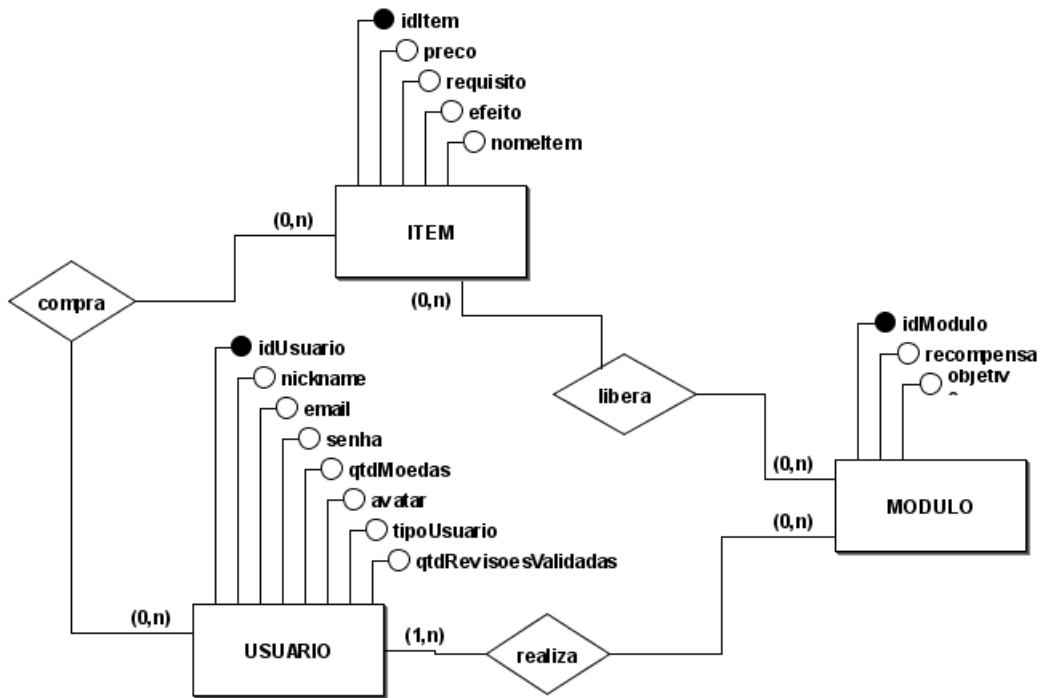


Figura 20 – Relação entre loja, modulo e usuário

4.5 Relação entre questões, missão e modulo

As Questões preenchem a Missão com várias Questões fazendo com que uma Missão seja um agrupamento de Questões.

Seguindo a mesma ideia da relação de Questão e Missão, o Modulo é preenchido por várias Missões, sendo assim um Modulo é um agrupamento de Missões.

A figura 21 mostra a relação entre questões, missão e modulo.

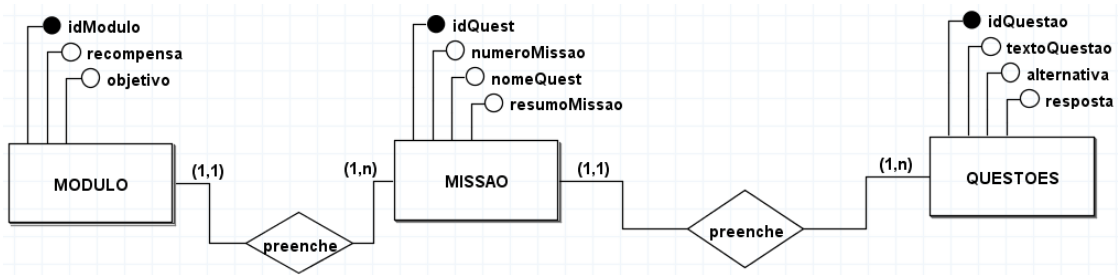


Figura 21 – Relação entre questões, missão e modulo

4.6 Relação entre Usuário e Investimento

O Usuário pode aplicar suas moedas em vários investimentos com a finalidade de aumentar a sua quantidade de moedas finais, podem os investimentos estão sujeitos a fatores externos que podem amplificar seus ganhos ou reduzi-los.

A figura 22 mostra a relação entre Usuário e Investimento.

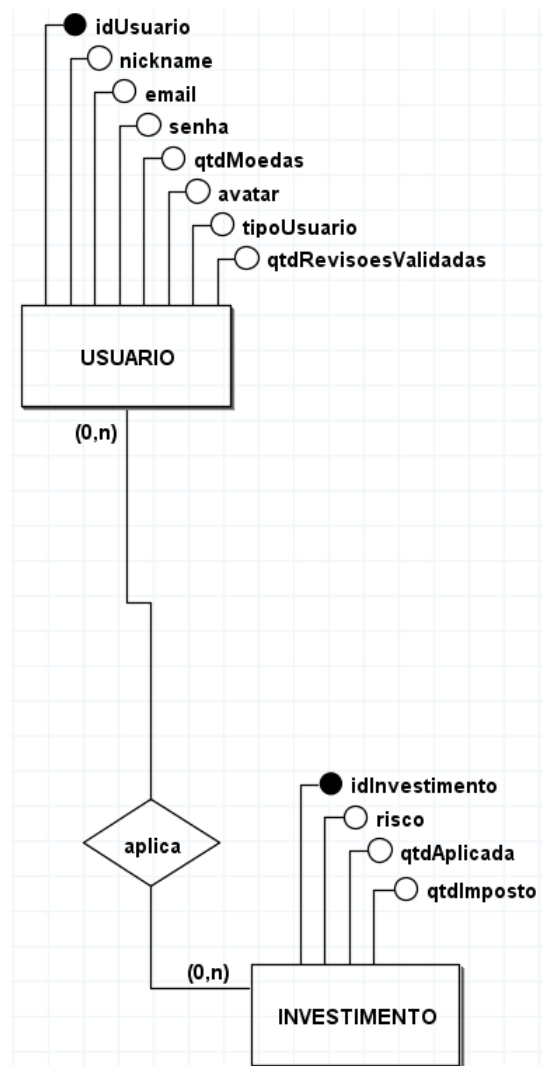


Figura 22 – Relação entre Usuário e Investimento

4.7 Relação entre Usuário, Progresso monetário e Progresso educacional

O Usuário possui Progresso educacional que mantém o rasto das Missões e Módulos realizados por aquele Usuário.

Seguindo a mesma ideia do Progresso educacional, o Progresso monetário mantém o rasto dos rendimentos de todos os investimentos do Usuário utilizando a valorização ou desvalorização do investimento para calcular o Progresso monetário.

A figura 23 mostra a relação entre Usuário, Progresso monetário e Progresso educacional.

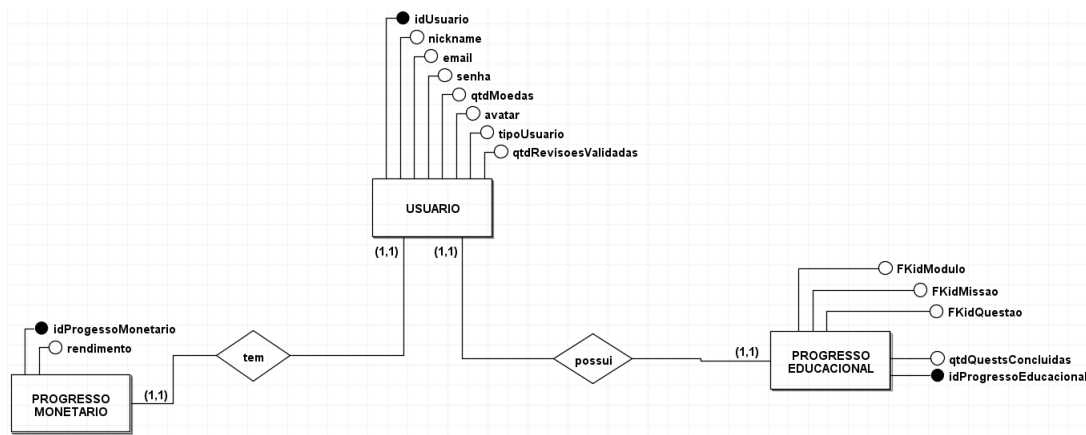


Figura 23 – Relação entre Usuário, Progresso monetário e Progresso educacional

5 Implementação do *\$study*

5.1 Arquitetura

A arquitetura do *\$study* foi implementada de maneira monolítica. A arquitetura monolítica é um modelo de *software* construído como uma grande unidade unificada a qual contém a si mesma, em uma base de código na qual une todas as regras de negócios (HARRIS, 2022).

O *front-end* é responsável por construir a interface gráfica onde o usuário realiza as interações visuais como navegar entre as telas e selecionar um investimento. O *front-end* também realiza as requisições para o *back-end* possibilitando a visualização dos dados em um banco de dados com a aplicação (SOUTO, 2022).

O *front-end* do *\$study* foi feito utilizando um *framework* chamado *React-native*, ele é responsável por controlar os elementos visuais na tela da aplicação e com um único código compilar uma aplicação para *IOS* ou *Android*, na linguagem nativa do sistema operacional utilizado (PATERSKA, 2021).

O *back-end* é responsável por controlar e tratar os dados expostos ou salvos no banco de dados bem como aplicar as regras de negócios e validações definidas no produto, esses dados são requisitados pelo *front-end* através de um protocolo chamado *http* o qual determina como é realizada uma requisição para o servidor e como ele responde a requisição (SOUTO, 2022).

O *back-end* do *\$study* foi feito utilizando um *framework* chamado *NodeJS* construído na linguagem de programação *Javascript* permitindo-se criar um *back-end* com o protocolo *http* possibilitando a comunicação com o *front-end* (MOZILLA, 2022).

No *\$study* o *back-end* é responsável por realizar a autenticação do usuário salvando os dados cadastrados no banco de dados e verificando esses dados no momento do *login*, realizar o controle da pontuação do usuário salvando a sua quantidade ou realizando o acréscimo e decréscimo dos pontos e realizar o controle dos investimentos do usuário sendo responsável por determinar a rentabilidade dos investimentos e salvar todos os investimentos adquiridos. O *back-end* também é o responsável por armazenar as questões, seus módulos bem como suas recompensas e respostas.

O banco de dados do *\$study* foi feito de forma relacional, ou seja, os dados são armazenados em tabelas as quais se relacionam por meio de chaves primárias únicas (ORACLE, 2022). Para a criação do banco de dados foi utilizado um gerenciador chamado *postgreSQL* o qual é responsável por criar, editar, modificar e deletar dados na base de

dados (JULIANO, 2014).

Para o versionamento do *Study* foi utilizado o *git* na plataforma *github*. O *git* é um sistema de controle de versão o qual é responsável por manter o histórico das versões do *software* bem como suas alterações (ATLASSIAN, 2022). O *github* é uma plataforma *online* onde fica salvo o projeto bem como o histórico de versões.

A tabela 5 mostra todas as ferramentas utilizadas para a produção e planejamento do *Study* e suas versões.

Tabela 5 – Ferramentas utilizadas e suas versões

| Ferramenta | Função | Versão |
|---------------------|--|--------|
| <i>Draw.io</i> | Criação de diagramas | - |
| <i>NodeJS</i> | Implementação do <i>back-end</i> | 16.17 |
| <i>React Native</i> | Implementação do <i>front-end</i> e interfaces gráficas | 17.0.2 |
| <i>Postgresql</i> | Gerenciamento do banco de dados | 14.05 |
| <i>Git</i> | Realizar o versionamento do <i>software</i> | 2.37.3 |
| <i>Heroku</i> | Plataforma responsável por hospedar o <i>back-end</i> do <i>Study</i> na nuvem | - |

A figura 24 é a representação visual da arquitetura implementada para o *Study*.

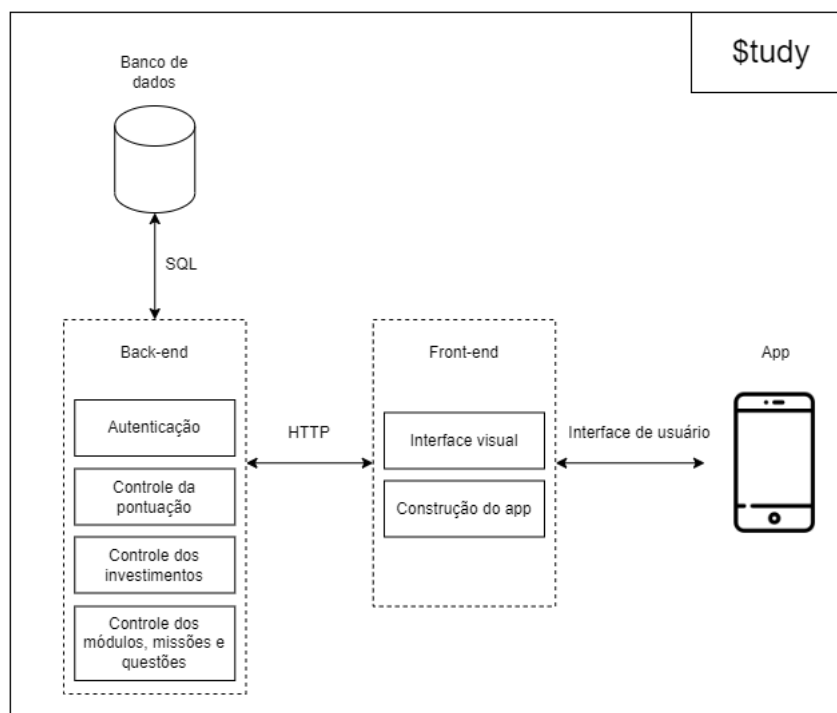


Figura 24 – Representação arquitetural do *Study*

5.2 Processo de *software*

O desenvolvimento do *Study* deu-se início após definido um mínimo produto viável. A definição do mínimo produto viável foi determinada pela análise da interdependência das técnicas de gamificação.

Um mínimo produto viável é um produto com funcionalidades e recursos suficientes para atrair clientes iniciais e validar o produto enquanto o mesmo está no início do desenvolvimento. Em um projeto ágil, o mínimo produto viável permite a equipe de desenvolvimento receber *feedbacks* dos usuários possibilitando a melhora do produto no seu estágio inicial (PRODUCTPLAN, 2022).

O mínimo produto viável do *Study* tem a finalidade de validar a ideia central do produto bem como suas funcionalidades principais.

A tabela 6 mostra as técnicas de gamificação implementadas no *Study*, seus requisitos, bem como a quantidade de dependentes.

Tabela 6 – Requisitos para o mínimo produto viável

| Quantidade de dependentes | Dependentes de | Core Drive | Requisitos |
|---------------------------|-------------------------|------------|---|
| 14 | Pontuação | 2 | Possuir moeda virtual, Mostrar total de moedas |
| 9 | Barra de progresso | 2 | Possuir indicativo de rendimento atual, Detalhar rendimento |
| 4 | Lista de missão | 2 | Detalhar missão específica, Listar missões,Mostrar recompensa de conclusão, Escolher qual missão realizar, Desistir da missão |
| 3 | Narrativa | 1 | Possuir tela de contextualização,Possuir protagonista na tela de contextualização |
| 3 | Escolhas significativas | 3 | Possuir fatores externos lúdicos que afetem o rendimento dos investimentos, Calcular Acréscimo/Decréscimo dos pontos |
| 0 | Percepção de escolha | 3 | Possuir tela de investimentos, Selecionar investimento, Possuir níveis de risco para os investimentos, Escolher quantidade de moedas investidas, Cancelar investimento, Possuir lista com investimentos aplicados |

Após a definição do mínimo produto viável foram definidas as *sprints* para a implementação do projeto. As *sprints* tinham a duração de uma semana, com a primeira se iniciando no dia 9 de junho de 2022 e a última no dia 18 de agosto de 2022.

Na definição das tarefas das *sprints* além dos requisitos presentes no mínimo produto viável foram listadas tarefas de configuração do ambiente de trabalho, configuração do projeto e a configuração da plataforma responsável por hospedar o *back-end* do *Study* na nuvem. No decorrer das *sprints* caso alguma tarefa não fosse concluída, a mesma era passada para a próxima *sprint*.

A tabela 7 mostra todas as tarefas executadas e requisitos implementados no *Study*. Também é possível notar tarefas as quais não foram finalizadas em sua *sprint* inicial por conta disso aparecem em mais de uma *sprint*.

Tabela 7 – Tarefas realizadas por *sprint*

| Período | Tarefas da <i>sprint</i> |
|---------------|---|
| 09/06 - 15/06 | Iniciar ambiente do <i>front-end</i> , Iniciar ambiente do <i>back-end</i> , Cronograma das <i>sprints</i> , Backlog das <i>sprints</i> , Criar componente padrão |
| 16/06 - 22/06 | Realizar modelagem dos dados no <i>back-end</i> , Criar a migração de dados no <i>back-end</i> , Criar tela de <i>login</i> , Criar tela de cadastro, Criar tela de <i>loading</i> |
| 23/06 - 29/06 | Realizar modelagem dos dados no <i>back-end</i> , Criar a migração de dados no <i>back-end</i> , Possuir um sistema de <i>login</i> , Possuir um sistema de <i>login</i> , Possuir um sistema de cadastro, Criar tela de <i>home</i> |
| 30/06 - 06/07 | Possuir tela de contextualização, Possuir protagonista na tela de contextualização, Possuir um sistema de cadastro, Migração de dados para criação das missões, módulos e questões |
| 07/07 - 13/07 | Possuir moeda virtual, Mostrar total de moedas, Possuir tela de investimentos, Possuir níveis de risco para os investimentos |
| 14/07 - 20/07 | Selecionar investimento, Escolher quantidade de moedas investidas, Cancelar investimento, Possuir indicativo de rendimento atual, Detalhar rendimento, Possuir moeda virtual, Mostrar total de moedas |
| 21/07 - 27/07 | Listar missões, Mostrar recompensa de conclusão, Tela da lista de investimentos aplicados, Escolher qual missão realizar, Realizar missão, Calcular Acréscimo/Decréscimo dos pontos, Desistir da missão, Possuir moeda virtual, Mostrar total de moedas |
| 28/07 - 03/08 | Listar missões, Mostrar recompensa de conclusão, Tela da lista de investimentos aplicados, Escolher qual missão realizar, Realizar missão, Calcular Acréscimo/Decréscimo dos pontos, Desistir da missão, Possuir moeda virtual, Mostrar total de moedas |
| 04/08 - 10/08 | Realizar missão, Calcular Acréscimo/Decréscimo dos pontos |
| 11/08 - 17/08 | <i>Deploy</i> do <i>back-end</i> , Gerar instalador para testes em <i>IOS</i> e <i>Android</i> , Realizar missão |
| 18/08 - 24/08 | <i>Deploy</i> do <i>back-end</i> , Gerar instalador para testes em <i>IOS</i> e <i>Android</i> , Realizar missão |

5.3 Versão final do \$tudy

O presente código está disponível na plataforma *GitHub* nos seguintes repositórios: github.com/rafaelteodosio/study-front-end e github.com/rafaelteodosio/study-back-end

5.3.1 Tela de login

A figura 25 é a tela de login, utilizada para autenticação do usuário no *\$tudy*. O requisito associado a essa tela é o **RF41**.

Na tela de login é possível notar os campos de *e-mail* e senha onde o usuário pode digita suas credenciais cadastradas, um botão para criação de conta para redirecionar o usuário para a tela de cadastro e o botão de *login* o qual redireciona o usuário para a tela inicial caso suas credenciais estejam corretas.

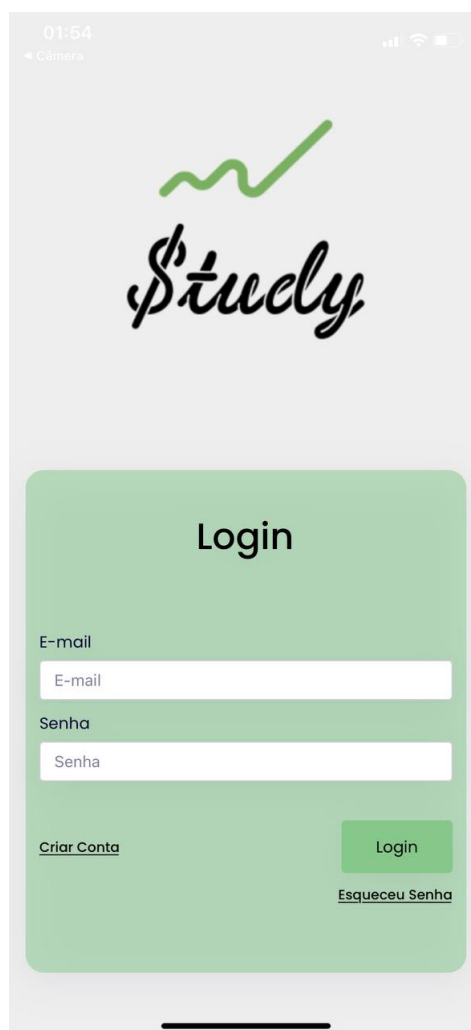
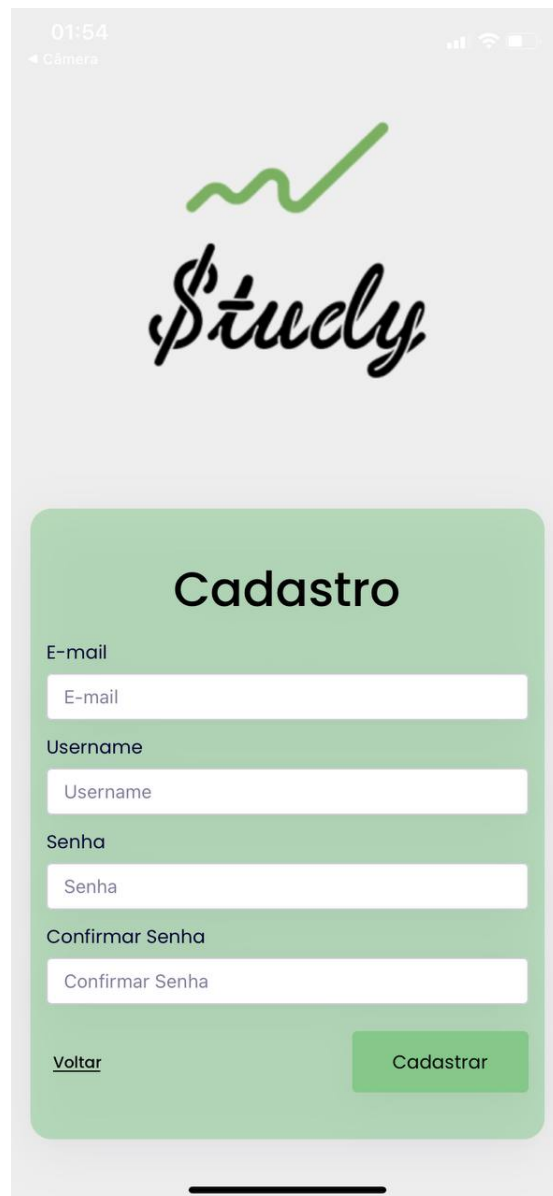


Figura 25 – Tela de login

5.3.2 Tela de cadastro

A figura 26 é a tela de cadastro, utilizada para o registro do usuário no *Study*. O requisito associado a essa tela é o **RF42**.

Na tela de cadastro é possível notar os campos onde o usuário digitar os dados que serão armazenados para ser feito o *login* na aplicação, um botão para voltar para a tela de *login* e o botão para realizar o cadastro dos dados.



A imagem mostra a tela de cadastro do aplicativo 'Study'. No topo, há o status bar com o horário 01:54 e ícones de câmera, sinal de rede e bateria. Abaixo, o logo 'Study' em uma fonte cursiva preta, precedido por uma linha decorativa verde. O formulário de cadastro é um retângulo verde com o título 'Cadastro' em negrito. Ele contém quatro campos de entrada brancos: 'E-mail', 'Username', 'Senha' e 'Confirmar Senha'. Na base do formulário, há um link 'Voltar' em azul e um botão verde 'Cadastrar'.

Figura 26 – Tela de cadastro

5.3.3 Tela do menu principal e tutorial

A figura 27 é a tela de menu principal e tutorial, utilizada para a contextualização, navegação, e explicação das funcionalidades para o usuário no *\$study*. Os requisitos associados a essa tela são os: **RF01**, **RF02**, **RF03**, **RF16**, **RF17**, **RF18**, **RF19**, **RF20**.

Na tela inicial é possível notar uma barra superior chamada de *header* onde o primeiro ícone é o logo do *\$study*, o segundo ícone é a representação de uma pessoa ele é responsável por permitir a escolha da imagem de perfil do usuário, porém essa funcionalidade não faz parte do mínimo produto viável. O terceiro ícone representado pelas barras verticais leva o usuário para a tela com uma lista de todos os investimentos aplicados, porem essa tela não foi totalmente implementada, o último ícone representa a quantidade total de moedas do usuário.

Abaixo do *header* são encontrados os botões para a navegação entre as telas de *Quests*, Investimentos, Loja e Notícias, porem as telas da Loja e Notícias não fazem parte do mínimo produto viável. Nessa tela pode-se notar na parte mais abaixo da tela uma área com texto onde são explicadas as funcionalidades básicas do *\$study* e uma seta para avançar durante a explicação.

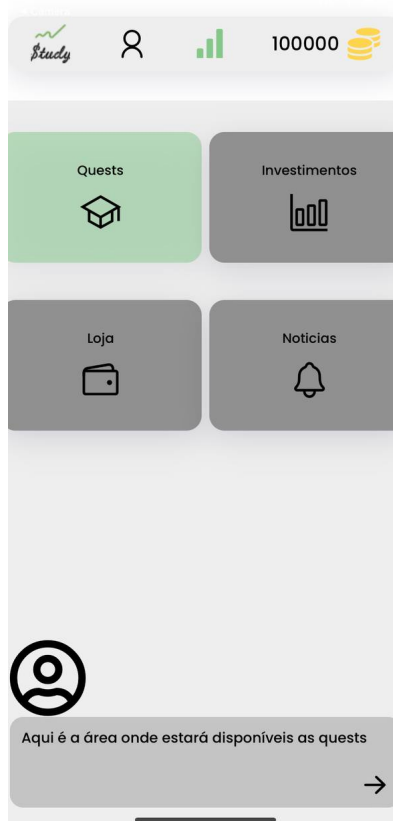


Figura 27 – Tela do menu principal e tutorial

5.3.4 Tela de investimento

A figura 28 é a tela de investimento utilizada para informar qual o preço e risco de um investimento e também para adquirir um novo investimento para o usuário no *Study*. Os requisitos associados a essa tela são os: **RF21**, **RF22**, **RF23**, **RF24**.

Nessa tela é possível encontrar o *header* citado na tela do menu principal, abaixo do *header* fica a lista dos investimentos disponíveis para o usuário adquirir utilizando as moedas, os investimentos são separados entre renda fixa e variável, ao pressionar os botões adquirir ou comprar é possível inserir a quantidade de moedas gastas. O risco de um investimento é representado pelo ícone com formato de uma estrela. Na lista de investimentos também é possível notar o nome do investimento ou ação e a sua rentabilidade ao lado de um ícone com barras verticais.



Figura 28 – Tela de investimento

5.3.5 Tela de perguntas e respostas

A figura 29 é a tela de perguntas e respostas utilizada para que o usuário possa realizar as missões solicitadas na aplicação e também realizar seus estudos a respeito da educação financeira. Os requisitos associados a essa tela são os: **RF13**, **RF14**, **RF15**.

Nessa tela é possível encontrar o *header* citado na tela do menu principal, abaixo do *header* existe um botão para retornar a tela anterior, também é possível notar um texto referente a pergunta realizada para o usuário, abaixo da pergunta são as respostas as quais somente uma delas é a resposta correta.

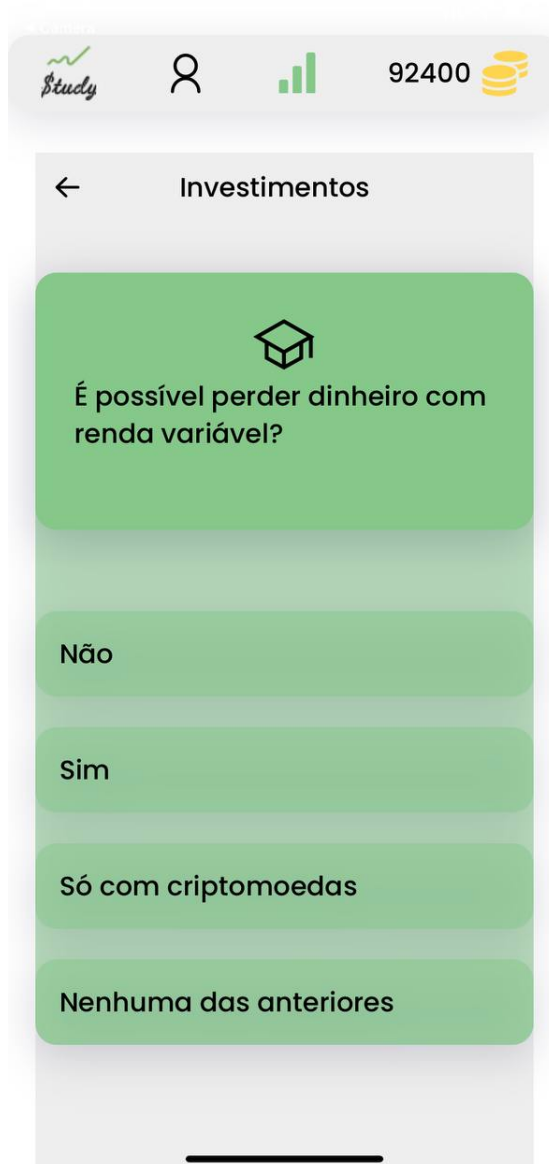


Figura 29 – Tela de perguntas e respostas

5.3.6 Tela das missões

A figura 30 é a tela das missões utilizada para que o usuário possa selecionar, visualizar, e desistir das missões propostas no %*tudy*. Os requisitos associados a essa tela são os: **RF09, RF10, RF12, RF13, RF14, RF15**.

Nessa tela é possível encontrar o *header*, abaixo dele existe uma lista com módulos de Renda Fixa e Renda Variável, ao selecionar um dos módulos a lista das missões disponíveis é mostrada, com os valores de recompensa ao concluir as missões, ao final da lista de missões existe o botão para iniciar as missões. Ao final dessa tela é possível notar uma barra chamada de *footer* nessa barra encontram-se os botões utilizados para a navegação na aplicação. O primeiro ícone representado por uma casa redireciona o usuário para a página do menu principal, ao lado dele existe o botão representado por um ícone com o formato de um capelo o qual redireciona o usuário para tela de missões, o ícone com as barras verticais redireciona o usuário para a tela de investimentos, o ícone representado por uma carteira redireciona o usuário para a tela da loja, porem ela não faz parte do mínimo produto viável e o último ícone redireciona para a tela com as notícias porem ela não faz parte do mínimo produto viável.



Figura 30 – Tela das missões

6 Conclusão e trabalhos futuros

Neste trabalho foram utilizados os conhecimentos adquiridos durante a graduação no curso de Engenharia de *Software* na Universidade de Brasília. As disciplinas da graduação mais presentes são as disciplinas de Requisitos de *software* presente em toda a elicitação de requisitos do *Study*, a disciplina de Métodos de desenvolvimento de *software* presente na metodologia de *software* aplicada e a disciplina de Arquitetura e desenho de *software* presente na decisão arquitetural da aplicação, bem como as tecnologias utilizadas para a sua implementação. Além dos conceitos presentes na graduação foi realizada uma revisão da literatura dos conceitos de educação financeira e gamificação. O *Study* foi desenvolvido com essa base teórica.

Observando o processo de desenvolvimento planejado e implementado tenho na minha percepção que o processo de desenvolvimento do *Study* teve um bom desempenho, considerando que seu planejamento e implementação foi realizado por uma pessoa com habilidades específicas.

O *Study* é um aplicativo gamificado para o ensino da educação financeira que apesar de não ter sido aplicado, possui potencial para impactar positivamente o mercado de *software* na área da educação financeira.

6.1 Trabalhos futuros

O *Study* é um aplicativo desenvolvido em um curto espaço de tempo, portanto sua implementação teve o escopo reduzido. O *Study* pode ter melhorias como:

- **Criar usuário para adição de questões:** A criação de questões são feitas diretamente no banco de dados, impossibilitando ao administrador realizar essa criação direto na aplicação;
- **Implementar a funcionalidade "esqueci minha senha":** Permitir ao usuário recuperar sua senha, sem a necessidade da intervenção de terceiros;
- **Implementar as funcionalidades de Loja e Notícias:** As funcionalidades de Loja e Notícias trazem uma profundidade ao aplicativo possibilitando diferentes interações;
- **Melhorar a interface de usuário e a experiência do usuário:** Visto que a interface gráfica implementada é simples, um aperfeiçoamento da mesma trará ao usuário uma experiência de uso mais agradável.

Referências

- ATLASSIAN. *O que é Git*. 2022. Acessado em: 24/08/2022. Disponível em: <<https://www.atlassian.com/br/git/tutorials/what-is-git>>. Citado na página 76.
- BCB. *Entenda o juro*. 2022. Acessado em: 13/04/2022. Disponível em: <<https://www.bcb.gov.br/cidadaniafinanceira/entendajuro>>. Citado na página 24.
- BCB. *O que é inflação*. 2022. Acessado em: 13/04/2022. Disponível em: <<https://www.bcb.gov.br/controleinflacao/oqueinflacao>>. Citado 3 vezes nas páginas 11, 23 e 24.
- BRASIL, B. do. *Guia de Renda Fixa*. 2022. Acessado em: 13/04/2022. Disponível em: <<https://www.bb.com.br/docs/pub/voce/dwn/rendafixa5.pdf>>. Citado na página 25.
- CAIUSCA, A. *Cálculo relacionado à matemática financeira*. 2019. Acessado em: 13/04/2022. Disponível em: <<https://www.educamaisbrasil.com.br/enem/matematica/juros-simples>>. Citado na página 24.
- CAMBRIDGE. *Methodology*. 2022. Acessado em: 25/04/2022. Disponível em: <<https://dictionary.cambridge.org/pt/dicionario/ingles/methodology>>. Citado na página 36.
- CHOU, Y. *Actionable Gamification: Beyond Points, Badges, and Leaderboards*. Createspace Independent Publishing Platform, 2015. ISBN 9781511744041. Disponível em: <<https://books.google.com.br/books?id=jFWQrgEACAAJ>>. Citado 9 vezes nas páginas 13, 27, 28, 29, 30, 31, 32, 33 e 41.
- CRAWFORD chris. *My Definition of 'Game'*. 1991. Acessado em: 13/04/2022. Disponível em: <<http://www.erasmatazz.com/library/the-journal-of-computer/jcgd-volume-4/my-definition-of-game.html>>. Citado 2 vezes nas páginas 11 e 26.
- DETERDING, S. et al. Gamification: Toward a definition. In: VANCOUVER, CANADÁ. *CHI 2011 gamification workshop proceedings*. [S.l.], 2011. v. 12, p. 12–15. Citado na página 26.
- DICIO. *Significado de Elicitar*. 2009. Acessado em: 25/04/2022. Disponível em: <<https://www.dicio.com.br/elicitare/>>. Citado 2 vezes nas páginas 33 e 35.
- DOMÍNGUEZ, A. et al. Gamifying learning experiences: Practical implications and outcomes. *Computers Education*, v. 63, p. 380–392, 2013. ISSN 0360-1315. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360131513000031>>. Citado na página 26.
- FARDO, M. L. A gamificação aplicada em ambientes de aprendizagem. *RENOTE*, v. 11, n. 1, jul. 2013. Disponível em: <<https://seer.ufrgs.br/index.php/renote/article/view/41629>>. Citado na página 21.
- FORTES, G. *Qual a diferença entre renda fixa e renda variável?* 2021. Acessado em: 13/04/2022. Disponível em: <<https://conteudos.xpi.com.br/aprenda-a-investir/relatorios/renda-fixa-e-variavel/>>. Citado na página 25.

- GOGUEN, J.; LINDE, C. Techniques for requirements elicitation. In: *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering*. [S.l.: s.n.], 1993. p. 152–164. Citado na página 35.
- GRAÇAS, A. G. *O que são despesas fixas e como economizar com elas?* 2021. Acessado em: 13/04/2022. Disponível em: <<https://neon.com.br/aprenda/economizar-dinheiro/despesas-fixas/>>. Citado na página 25.
- HARRIS, C. *Microservices vs. monolithic architecture*. 2022. Acessado em: 24/08/2022. Disponível em: <<https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith>>. Citado na página 75.
- IED. *5 tipos de paleta de cores que todo designer deve conhecer*. 2021. Acessado em: 25/04/2022. Disponível em: <<https://ied.edu.br/100porcentodesign/paleta-de-cores>>. Citado na página 57.
- JULIANO. *Gerenciamento de Banco de Dados: Análise Comparativa de SGBD'S*. 2014. Acessado em: 24/08/2022. Disponível em: <<https://www.devmedia.com.br/gerenciamento-de-banco-de-dados-analise-comparativa-de-sgbd-s/30788>>. Citado na página 76.
- KIM, B. *Understanding Gamification*. American Library Association, 2015. (Library technology reports : expert guides to library systems and services). ISBN 9780838959534. Disponível em: <<https://books.google.com.br/books?id=vF8orgEACAAJ>>. Citado na página 26.
- LOGICEARTH. *OCTALYSIS DRIVERS OF HUMAN EMOTION*. 2022. Acessado em: 13/04/2022. Disponível em: <<https://www.logicearth.com/octalysis-infographic/>>. Citado 2 vezes nas páginas 11 e 27.
- MALL, R. *Fundamentals of software engineering*. [S.l.]: PHI Learning Pvt. Ltd., 2018. Citado na página 33.
- MIT. *Brainstorming Guidelines*. 2022. Acessado em: 25/04/2022. Disponível em: <<https://hr.mit.edu/learning-topics/meetings/articles/brainstorming>>. Citado na página 34.
- MOREIRA, E. 5 princípios da educação financeira para adotar como hábito. 2020. Acessado em: 13/04/2022. Disponível em: <<https://edumoreira.com.br/5-principios-da-educacao-financeira/>>. Citado na página 23.
- MOZILLA. *Introdução Express/Node*. 2022. Acessado em: 24/08/2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction#introduzindo_o_express>. Citado na página 75.
- OCDE. *Recommendation on Principles and Good Practices for Financial Education and Awareness*. Paris, França, 2005. 7 p. Citado 2 vezes nas páginas 21 e 23.
- OLIVEIRA, R. Prince2: A técnica de priorização moscow. 01 2014. Citado na página 35.
- ORACLE. *O que é um banco de dados relacional (RDBMS)?* 2022. Acessado em: 24/08/2022. Disponível em: <<https://www.oracle.com/br/database/what-is-a-relational-database/>>. Citado na página 75.

- PATERSKA, P. *Front-end, Back-end e Full Stack*. 2021. Acessado em: 24/08/2022. Disponível em: <<https://www.elpassion.com/blog/what-is-react-native-and-when-to-use-it>>. Citado na página 75.
- PRADO, N. A. e E. V. Técnicas de elicitação de requisitos no desenvolvimento de software: uma revisão sistemática da literatura. *AtoZ: novas práticas em informação e conhecimento*, v. 10, n. 1, p. 39–49, 2021. ISSN 2237-826X. Disponível em: <<https://revistas.ufpr.br/atoz/article/view/77393>>. Citado na página 34.
- PRODUCTPLAN. *Minimum Viable Product (MVP)*. 2022. Acessado em: 24/08/2022. Disponível em: <<https://www.productplan.com/glossary/minimum-viable-product>>. Citado na página 77.
- PUCPR. *Juros simples e compostos: saiba o que é, fórmula e como calcular*. 2021. Acessado em: 13/04/2022. Disponível em: <<https://ead.pucpr.br/blog/juros-simples-compostos-formula/>>. Citado na página 25.
- RICO. *Ativo Financeiro: O que é, Características e Classificações!* 2021. Acessado em: 13/04/2022. Disponível em: <<https://riconnect.rico.com.vc/blog/ativo-financeiro>>. Citado na página 26.
- SOMMERVILLE, I. *Software Engineering*. 9. ed. Harlow, England: Addison-Wesley, 2010. ISBN 978-0-13-703515-1. Citado 6 vezes nas páginas 11, 33, 34, 36, 37 e 39.
- SOUTO, M. *Front-end, Back-end e Full Stack*. 2022. Acessado em: 24/08/2022. Disponível em: <<https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>>. Citado na página 75.
- STIELER, E. C. Definição de taxa de juros. 2009. Acessado em: 13/04/2022. Disponível em: <http://www2.unemat.br/eugenio/taxas_de_juros.html>. Citado na página 24.
- TECNICON. *Metodologia Scrum para a gestão de processos ágeis na indústria*. 2019. Acessado em: 25/04/2022. Disponível em: <https://www.tecnicon.com.br/blog/411-Metodologia_Scrum_para_a_gestao_de_processos_ageis_na_industria>. Citado 2 vezes nas páginas 11 e 37.
- TIWARI, S.; RATHORE, S. A methodology for the selection of requirement elicitation techniques. 09 2017. Citado 2 vezes nas páginas 34 e 35.
- VALENTE, M. T. D. O. *Engenharia de software moderna : princípios e práticas para desenvolvimento de software com produtividade*. Universidade Federal de Minas Gerais, 2020. Disponível em: <<https://engsoftmoderna.info/>>. Citado 4 vezes nas páginas 11, 33, 38 e 39.
- WIEGERS, K.; BEATTY, J. *Software Requirements*. Microsoft Press, 2013. (Best practices). ISBN 9780735679665. Disponível em: <<https://books.google.com.br/books?id=40IDmAEACAAJ>>. Citado na página 34.
- XP. *O que é Renda Fixa e por que investir?* 2022. Acessado em: 13/04/2022. Disponível em: <<https://www.xpi.com.br/investimentos/renda-fixa/>>. Citado na página 25.
- YOUNG, D. Software development methodologies. *White paper*, 08 2013. Citado na página 36.