

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
ETEC DA ZONA LESTE
Mtec Desenvolvimento De Sistemas**

Letícia Borges da Silva

Mariana Akemi Arashiro Santos Feitosa

Yasmin Pereira Lima Oliveira

**TEDDYMATH: IoT para Auxílio em Matemática para Crianças do
Ensino Fundamental I**

São Paulo

2024

Letícia Borges da Silva

Mariana Akemi Arashiro Santos Feitosa

Yasmin Pereira Lima Oliveira

**TEDDYMATH: IoT para Auxílio em Matemática para Crianças do
Ensino Fundamental I**

Trabalho de Conclusão de Curso apresentado
ao Curso Técnico em Desenvolvimento de
Sistemas da Etec Zona Leste, orientado pelo
Prof. Esp. Jeferson Roberto de Lima, como
requisito parcial para obtenção do título de
técnico em Desenvolvimento de Sistemas.

São Paulo

2024

Aos nossos pais, que são a base de todos os nossos sonhos. Desde a infância, vocês nos ensinaram a sonhar alto e a acreditar em nossas capacidades. Foram nossos primeiros professores, nossos maiores incentivadores e nossas inspirações constantes. Obrigada por nos guiarem com amor e sabedoria, e por nos mostrarem que, com determinação e dedicação, podemos alcançar qualquer objetivo. Este trabalho é fruto de tudo que aprendemos com vocês ao longo dos anos.

AGRADECIMENTOS

Assim como grande parte das conquistas de nossas vidas, a realização deste Trabalho de Conclusão de Curso (TCC) representa o fim de um importante ciclo, que só foi possível graças a todas as pessoas que sempre nos apoiaram e incentivaram desde o início. Gostaríamos de expressar nossa mais profunda gratidão a todas essas pessoas, cujo apoio e incentivo constantes tornaram possível a concretização deste projeto. É com grande respeito e gratidão que fazemos os seguintes agradecimentos:

Primeiramente, agradecemos a Deus, por nos conceder saúde, sabedoria e perseverança para concluir essa etapa.

Às nossas famílias, que sempre foram nossa base e nosso alicerce e principalmente aos nossos pais, por todo amor, incentivo e por acreditarem em nosso potencial, mesmo nos momentos mais difíceis vocês estiveram presentes.

Aos nossos amigos, que estiveram ao nosso lado durante essa jornada, compartilhando momentos de alegria e de dificuldades. Vocês foram essenciais para que mantivéssemos a motivação e a confiança necessárias para seguir em frente.

Aos nossos Coordenadores, Jeferson Roberto de Lima e Rogério Bezerra, por sua paciência, orientação precisa e encorajamento constante. A dedicação e disponibilidade de vocês foram fundamentais para a elaboração deste trabalho. Agradecemos por cada conselho, crítica construtiva e por nos guiar durante todo o processo. Vocês fazem com que esse curso seja cada dia melhor!

Aos professores que contribuíram para nossa formação acadêmica ao longo desses anos. Cada aula, discussão e orientação foram um aprendizado para nossa vida e nossa carreira profissional. Vocês foram peças fundamentais na construção do nosso conhecimento.

Por fim, agradecemos a todos que, de alguma forma, contribuíram para a realização deste TCC. Este trabalho é resultado do esforço conjunto e da colaboração de muitas pessoas. A todos, nosso mais sincero, obrigada!

"Brincar é a maneira natural da criança aprender. Através do jogo, ela explora, experimenta e descobre o mundo ao seu redor."

Maria Montessori

RESUMO

O presente estudo tem como objetivo a implementação de um aplicativo móvel integrado a um brinquedo interativo, com o propósito de fornecer suporte a crianças com dificuldades de aprendizagem em matemática. A atual pesquisa investiga as metodologias empregadas para a criação de uma experiência de aprendizagem envolvente, eficaz, criativa e personalizada, adequando-se ao atual contexto tecnológico. Com o intuito de alcançar esse objetivo, foram realizados pesquisas e estudos sobre a aplicação e o uso da Internet das Coisas (IoT) em conjunto com o ambiente mobile, incluindo a análise de requisitos, elaboração de diagramas e a prototipagem de interfaces do sistema, além da incorporação de componentes eletrônicos, como um alto-falante com arquivos de áudio integrados ao aplicativo. Com uma abordagem interdisciplinar, pretende-se explorar as potencialidades dessa ferramenta para lidar com os desafios da educação de matemática para crianças do Ensino Fundamental I (faixa etária entre 6 e 8 anos) os resultados não se limitam a diminuir as dificuldades de aprendizagem, mas também estimulam o interesse e a participação ativa dos alunos no processo educacional, abrangendo não somente a matemática, mas também o conteúdo programático.

Palavras-chave: aplicativo móvel; brinquedo pedagógico; aprendizagem de matemática; ensino fundamental I;

ABSTRACT

This study aims to implement a mobile application integrated with an interactive toy, with the purpose of providing support to children with learning difficulties in mathematics. To do so, it's necessary investigates the methodologies used to create an engaging, effective, creative, and personalized learning experience, adapting to the current technological context. With an interdisciplinary approach, the main result is to explore the potential of this tool to deal with the challenges of mathematics education for elementary school children (aged between 6 and 8). The results are not limited to reducing learning difficulties, but also stimulating students' interest and active participation in the educational process, covering not only mathematics, but also the syllabus.

Keywords: mobile application; pedagogical toy; math learning; elementary school.

LISTA DE ILUSTRAÇÕES

Figura 1 - Elaboração de um formulário de contato utilizando HTML.....	16
Figura 2 - Demostraçāo do resultado obtido aps a codificaçāo em HTML	19
Figura 3 - Exemplo de codificaçāo em CSS	20
Figura 4 - Continuaçāo do Exemplo de codificaçāo em CSS	21
Figura 5 - Incorporando o CSS em página HTML	23
Figura 6 - Visualizaçāo da página HTML estilizada com CSS.....	24
Figura 7 - Linguagens de programação mais populares do GitHub.	25
Figura 8 - Exemplo de codificaçāo em JavaScript utilizando funções	26
Figura 9 - Resultado da codificaçāo em JavaScript	27
Figura 10 - Implementando o JavaScript em uma página HTML	27
Figura 11 - Codificaçāo de um formulário em React Native.	29
Figura 12 - Resultado da codificaçāo de um formulário em React Native.....	32
Figura 13 - Uso do StyleSheet para estilizaçāo do código React Native.....	33
Figura 14 - Infográfico apresentando os serviços encontrados no Firebase.	36
Figura 15 - Cloud Firestore Logo.....	37
Figura 16 - Exemplo de Diagrama de Caso de Uso	39
Figura 17 - Exemplo de Diagrama de Sequênciा.....	41
Figura 18 - Exemplo de Diagrama de Atividade	42
Figura 19 - Exemplo de Diagrama de Máquina de Estado	43
Figura 20 - Exemplo de prototipaçāo inicial de uma tela de login e cadastro, utilizando wireframes de baixa fidelidade	44
Figura 21 - Exemplo de prototipaçāo inicial de uma tela de login e cadastro, utilizando wireframes de Alta fidelidade	45
Figura 22 - Codificaçāo C++ para ESP32	46
Figura 23- Resultado da codificaçāo em C++	48
Figura 24 - ESP32 - Dual Core.....	50
Figura 25 - Jumper Macho Macho.....	51
Figura 26 - Resistores	52
Figura 27 - Alto Falante – 2" YD50.....	53
Figura 28 – Push Button.....	55
Figura 29 - Exemplo de Prototipaçāo de Modelagem 3D.....	56
Figura 30 - Diagrama de Caso de Uso do Sistema TeddyMath.....	57

Figura 31- Diagrama de Sequência: Cadastrar Criança.....	70
Figura 32 - Diagrama de Sequência: Cadastrar Responsável	71
Figura 33 - Diagrama de Sequência: Criar Perfil	72
Figura 34 - Diagrama de Sequência: Editar Perfil	73
Figura 35 - Diagrama de Sequência: Excluir Perfil.....	74
Figura 36 - Diagrama de Sequência: Fazer Login.....	75
Figura 37 - Diagrama de Sequência: Gerenciar Pontuação.....	76
Figura 38 - Diagrama de Sequência: Monitorar Progresso.	77
Figura 39 - Diagrama de Sequência: Notificar Desempenho.	78
Figura 40 - Diagrama de Sequência: Parear Dispositivo.....	79
Figura 41 - Diagrama de Atividade: Cadastrar Criança.....	80
Figura 42 - Diagrama de Atividade: Cadastrar Responsável.	81
Figura 43 - Diagrama de Atividade: Criar Perfil	82
Figura 44 - Diagrama de Atividade: Editar Perfil	83
Figura 45 - Diagrama de Atividade: Excluir Perfil.	84
Figura 46 - Diagrama de Atividade: Fazer Login	84
Figura 47 - Diagrama de Atividade: Gerenciar Pontuação	85
Figura 48 - Diagrama de Atividade: Jogar	86
Figura 49 - Diagrama de Atividade: Monitorar Progresso.	87
Figura 50 - Diagrama de Atividade: Notificar Desempenho.	88
Figura 51 - Diagrama de Atividade: Parear Dispositivo.....	89
Figura 52 - Diagrama de Máquina de Estados: Criar Perfil.	90
Figura 53 - Diagrama de Máquina de Estados: Editar Perfil.	91
Figura 54 - Diagrama de Máquina de Estados: Excluir Perfil.	92
Figura 55 - Diagrama de Máquina de Estados: Fazer Login.	93
Figura 56 - Diagrama de Máquina de Estados: Gerenciar Pontuação.	94
Figura 57 - Diagrama de Máquina de Estados: Monitorar Progresso.....	95
Figura 58 - Diagrama de Máquina de Estados: Notificar Desempenho.....	96
Figura 59 - Diagrama de Máquina de Estados: Parear Dispositivo	97
Figura 60 - Diagrama de conexão entre os componentes.....	98
Figura 61 - Wireframe Baixa Fidelidade do Site: Página Inicial, Vantagens e Como Funciona.....	99
Figura 62 - Wireframe Alta Fidelidade do Site: Página Inicial, Vantagens e Como Funciona.....	100

Figura 63 - Wireframe Baixa Fidelidade do Site: Desenvolvedoras e Contato	101
Figura 64 - Wireframe Alta Fidelidade do Site: Desenvolvedoras e Contato.....	102
Figura 65 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Inicial.....	103
Figura 66 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Login.....	104
Figura 67 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Cadastro.	105
Figura 68 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Perfil.....	106
Figura 69 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Perfil Nome e Gênero	107
Figura 70 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Perfil Idade.....	108
Figura 71 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Perfil Completo.109	
Figura 72 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Menu.....	110
Figura 73 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Splash.....	111
Figura 74 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Colorê.	112
Figura 75 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Jogo Caça Formas.	113
Figura 76 - Wireframe de Baixa e Alta Fidelidade da Tela Caça-Formas (Continuação).	113
Figura 77 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Caça Formas (Concluído).....	114
Figura 78 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela TeddyQuiz.....	115
Figura 79 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Parabéns.	116
Figura 80 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Tente Novamente	117
Figura 81 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Tabuada.....	118
Figura 82 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Acesso Área do Responsável.....	119
Figura 83 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Área do Responsável.....	120
Figura 84 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Progresso Criança.	121
Figura 85 - Modelagem 3D Case do Circuito Eletrônico.	122
Figura 86 - Case do Circuito Eletrônico.....	122
Figura 87 - TeddyMath Brinquedo IoT.....	123
Figura 88 - Circuito do Dispositivo.....	124

LISTA DE QUADROS

Quadro 1 - Exemplo de Documentação de Caso de Uso.....	40
Quadro 2 - Descrição do caso de uso : Selecionar jogo	59
Quadro 3 - Descrição do caso de uso: Jogar	60
Quadro 4 - Descrição de caso de uso: Cadastrar criança	60
Quadro 5 - Descrição do caso de uso: Manter Responsável	61
Quadro 6 - Descrição de caso de uso: Cadastrar responsável	64
Quadro 7 - Descrição do caso de uso: Monitorar Progresso.....	65
Quadro 8 - Descrição do caso de uso: Gerenciar Pontuação	66
Quadro 9 - Descrição do caso de uso: Parear Dispositivo	67

LISTA DE ABREVIATURAS E SIGLAS

Application Programming Interface (API)
Cascading Style Sheets (CSS)
Extensible Markup Language (XML)
HyperText Markup Language (HTML)
Internet of Things (IoT)
iPhone Operating System (iOS)
JavaScript (JS)
Light Emitting Diode (LED)
Node Package Manager (NPM)
Not Only SQL (NoSQL)
Programação Orientada a Objetos (POO)
Radio Frequency Identification (RFID)
Software Development Kits (SDKs)
Structured Query Language (SQL)
Unified Modeling Language (UML)
Universal Serial Bus (USB)
User Interface (UI)
Wireless Fidelity (WI-FI)
World Wide Web Consortium (W3C)

SUMÁRIO

1 INTRODUÇÃO.....	13
2 REFERENCIAL TEÓRICO	14
2.1 Dificuldade no aprendizado de matemática entre as crianças	14
2.2 HTML.....	15
2.3 CSS	19
2.4 JavaScript.....	24
2.5 React.....	28
2.6 React Native	28
2.7 StyleSheet.....	32
2.8 Node.js e NPM	35
2.9 Banco de dados	35
2.10 Firebase	36
2.11 Firestore	36
2.12 UML	37
2.12.1 Diagrama de Casos de Uso	38
2.12.2 Documentação de Caso de Uso	39
2.12.3 Diagrama de Sequência	40
2.12.4 Diagrama de Atividade	41
2.12.5 Diagrama de Máquina de Estados.....	42
2.13 Wireframe	43
2.14 Figma	45
2.15 C++	46
2.16 IoT	48
2.17 Componentes	49

2.17.1	ESP32	49
2.17.2	Jumpers.....	50
2.17.3	Resistores	51
2.17.4	Alto Falante	52
2.17.5	Módulo MP3 <i>DFPlayer Mini</i>	53
2.17.6	Push Button	54
2.18	Impressão 3D	55
2.19	Modelagem 3D.....	55
3	DESENVOLVIMENTO	56
3.1	Diagrama de Casos de Uso.....	56
3.1.1	Documentação dos Casos de Uso	57
3.2	Diagrama de Sequência	70
3.3	Diagrama de Atividade	79
3.4	Diagrama de Máquina de Estados.....	89
3.5	Montagem do circuito do dispositivo	98
3.6	Prototipagem das telas da aplicação	98
3.7	Modelagem 3D para o Hardware.....	121
3.8	TeddyMath: Brinquedo IoT.....	123
4	CONSIDERAÇÕES FINAIS.....	124
	REFERÊNCIAS.....	127

1 INTRODUÇÃO

De acordo com dados do QEdu (2021), o Índice de Desenvolvimento da Educação Básica (IDEB) revela uma preocupante falta de proficiência em matemática nos anos iniciais do ensino fundamental, com apenas 37% dos alunos alcançando um nível satisfatório de aprendizagem na disciplina.

As dificuldades de aprendizagem dos alunos vão além da falta de material e explicação dos professores, envolvendo condições como dislexia, discalculia e TDAH, afetando a compreensão dos conteúdos educacionais. (DUTRA, 2019). Os problemas enfrentados pelos alunos da educação básica em matemática são variados, englobando aspectos lógicos, operacionais, perceptivos e interpretativos (ANDRADE; COLARES; COSTA, 2018).

De acordo com os Parâmetros Curriculares Nacionais (PCNs) referentes ao Ensino Fundamental:

A constatação da sua importância apoia-se no fato de que a Matemática desempenha papel decisivo, pois permite resolver problemas da vida cotidiana, tem muitas aplicações no mundo do trabalho e funciona como instrumento essencial para a construção de conhecimentos em outras áreas curriculares. Do mesmo modo, interfere fortemente na formação de capacidades intelectuais, na estruturação do pensamento e na agilização do raciocínio dedutivo do aluno. (BRASIL, 1997).

Nos próximos anos, a tecnologia terá um grande impacto na aprendizagem de matemática. Os métodos tradicionais de ensino como livros, papel utilizam apenas o homem como transmissor de conhecimentos. (D'AMBRÓSIO, 1986). Há uma necessidade de fazer uma mesclagem entre o ensino tradicional e o não tradicional, implementando softwares como jogos pedagógicos, tornando a aprendizagem mais atraente e divertida. (SILVA, 2009).

A utilização de brinquedos interativos no ensino matemático é uma opção eficaz para enfrentar esses desafios, pois a integração da tecnologia à educação proporciona uma abordagem mais lúdica e atrativa no aprendizado das crianças. (TEIXEIRA; VITAL,

2022). Tais ferramentas podem ajudar no desenvolvimento do raciocínio lógico, a resolução de problemas e o pensamento crítico, tornando o ensino mais eficaz e divertido. Ademais, a aplicação desses brinquedos pode contribuir para um aumento significativo da motivação e do interesse dos alunos, facilitando a assimilação dos conceitos matemáticos de forma natural e intuitiva.

Com base nas dificuldades encontradas por uma proporção significativa de crianças do ensino fundamental na aprendizagem de matemática básica, a hipótese baseada na premissa de que a aprendizagem efetiva ocorre quando os alunos estão engajados e motivados apoia a implementação de um aplicativo móvel, conectado a um brinquedo interativo com jogos relacionados ao conteúdo de matemática básica, pode despertar o interesse das crianças pela matemática, o que consequentemente facilitará a compreensão e fixação de conceitos fundamentais.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta todos os fundamentos teóricos e as tecnologias empregadas para a criação e o desenvolvimento do sistema TeddyMath.

2.1 Dificuldade no aprendizado de matemática entre as crianças

A aprendizagem de matemática básica entre as crianças do Ensino Fundamental I apresenta diversos desafios com uma grande proporção de estudantes enfrentando dificuldades, como resolução de cálculos, multiplicação e tabuadas sendo os mais frequentes.

De acordo com Viana (2024), no Ceará, 48,4% dos alunos do 5º ano do Ensino Fundamental atingiram um “nível apropriado” de aprendizado em matemática. Esses dados foram considerados críticos e preocupantes. A falta de concentração do aluno, dificuldade de interpretação e o raciocínio lógico não aperfeiçoado são razões que afetam à aprendizagem de matemática. (COUTINHO; EBERHARDT, 2011).

A Base Nacional Comum Curricular (BNCC) justifica a ideia de:

[...] o ensino fundamental deve ter compromisso com o desenvolvimento do letramento matemático, definido como as competências e habilidades. de raciocinar, representar, comunicar e argumentar matematicamente, de modo a favorecer o estabelecimento de conjecturas, a formulação e a resolução de problemas em uma variedade de contextos, utilizando conceitos, procedimentos, fatos e ferramentas matemáticas. (BRASIL, 2017, p. 264).

Diante dos principais motivos que geram essas dificuldades, é crucial adotar uma abordagem inovadora, acessível e eficiente para despertar o interesse e facilitar a compreensão dos conceitos matemáticos desde o início da educação formal.

Nesse sentido, a Base Nacional Comum Curricular (BNCC) afirma que:

O conhecimento matemático é necessário para todos os alunos da Educação Básica, seja por sua grande aplicação na sociedade contemporânea, seja pelas suas potencialidades na formação de cidadãos críticos, cientes de suas responsabilidades sociais. (BRASIL, 2017, p. 263).

É essencial que o ensino de matemática seja contextualizado, tornando-o mais envolvente e significativo para os estudantes, ou que facilite a superação de barreiras iniciais no entendimento da disciplina. (TORQUATO; SILVA, 2024). Portanto, a integração de uma aplicação móvel com jogos pode ser valiosa e proporcionar um grande desenvolvimento nos conhecimentos matemáticos, aumentando o interesse e a motivação. (ANTUNES, 2020). A implementação de um *software* com um brinquedo interativo promove a ludicidade e auxilia no desenvolvimento da criança, incluindo a fala, memória, atenção e relações sociais. (BRANDT; MORETTI, 2016).

2.2 HTML

Como apontado por Eis e Ferreira (2012), o HTML, cuja abreviação contém o significado de *HyperText Markup Language*, é uma linguagem operada para publicações de conteúdo na web.

De acordo com Ferreira (2018) uma página pode conter diversos itens, tais como imagens, textos, parágrafos, títulos e quebra linhas. Portanto para criação desses elementos, existe uma *tag* designada para cada um deles.

Figura 1 - Elaboração de um formulário de contato utilizando HTML.

```

1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="style.css">
7      <title>Forms</title>
8  </head>
9  <body>
10     <div class="container">
11
12         <form method="post">
13             <div class="titulo">
14                 <h1>Contato</h1>
15             </div>
16             <div class="campo-input">
17                 <label for="text">Nome</label>
18                 <br>
19                 <input type="text" id="text" name="text" placeholder="Digite seu nome" required><br><br>
20             </div>
21             <div class="campo-input">
22                 <label for="email">Email:</label>
23                 <br>
24                 <input type="text" id="email" name="email" placeholder="Digite seu email" required><br><br>
25             </div>
26
27             <div class="campo-input">
28                 <label for="text">Mensagem</label>
29                 <br>
30                 <textarea placeholder=""></textarea>
31             </div>
32
33                 <button type="submit">Enviar</button>
34             </form>
35     </div>
36 </body>
37 </html>
```

Fonte: De autoria própria, 2024.

É de suma importância ressaltar que as *tags* consistem em um comando inserido entre os símbolos `<` e `>`. Tipicamente, elas começam com `<...>` e são encerradas com `</...>`. Portanto, nem todas exigem ser fechadas. (COSTA, 2007).

Para uma compreensão mais clara, as *tags* do código presentes na figura acima são:

- *Doctype*: “`<!DOCTYPE html>`” conhecido por não ser uma *tag*, e sim uma instrução especial. Ela informa para o navegador qual versão do HTML será utilizado para renderizar a página;
- *Html*: “`<html></html>`” é fundamental na estrutura do HTML, servindo como base principal para todos os elementos do seu documento, marcando inicialmente no código e no final do conteúdo;
- *Head*: “`<head></head>`” delimita o cabeçalho do documento, ela não possui um valor significativo, mas, em suma, é capaz de transmitir ao navegador diversas informações auspíciosas e imprescindível para uma boa apresentação do seu documento;

- **Meta:** “<meta></meta>” permite a inserção de metadados em seu documento Html. Utilizando o atributo “charset=“ UTF-8”, assegura-se a compatibilidade do código com os caracteres de padrão latino-americano;
- **Title:** “<title></title>” determina o título da página, que apresentará em sua janela, aba ou guia. Esta tag é primordial para otimização de mecanismos de busca (SEO);
- **Body:** “<body></body>” representa o corpo da página HTML. Em resumo, é dentro dessa tag onde todos os elementos visíveis do seu site são inseridos;
- **Div:** “<div></div>” define uma seção da página, atuando como um container para conteúdo de fluxo, proporcionando uma organização mais estruturada do conteúdo.
- **Form:** “<form></form>” estabelece um bloco destinado a agrupar tags de formulário e seus respectivos dados, facilitando a coleta e o envio de conjunto de dados para outro documento.
- **H1:** “<h1></h1>” empregada para definir o cabeçalho principal de uma seção ou página web. O texto contido entre as tags, é apresentado em um tamanho ampliado, indicando a importância e centralidade do título.
- **Label:** “<label></label>” é empregada para associar um rotulo descriptivo a elementos de formulário, como campos de entrada e seleção. Assegura uma identificação clara dos campos, otimizando a usabilidade e acessibilidade do formulário.
- **Br:** “
” operada para inserir uma quebra de linha, permitindo a interrupção do conteúdo, assim, perpetrando uma nova linha em seguida.
- **Input:** “<input></input>” utilizada para estabelecer controles interativos em formulários web, facilitando a inserção de dados. A interpretação e sua funcionalidade podem divergir substancialmente com base no valor especificado em seu atributo.
- **Textarea:** “<textarea></textearea>” atua como um campo e edição extensível, proporcionando uma caixa de texto adequada para inserção de textos longos.
- **Button:** “<button></button>” é empregado em páginas web com o propósito da criação de um botão interativo para executar deliberadas ações.

É válido destacar que as *tags* possuem atributos que funcionam como modificadores dos elementos, especificando suas propriedades e são inseridos na *tag* de abertura (CALDEIRA, 2015).

- *Lang="..."* tem como sua funcionalidade a tipificação o idioma do conteúdo da *tag*.
- *Id="..."* utilizado para fornecer uma identificação única a um elemento individual em uma página.
- *Class="..."* aplica-se para atribuição de uma classe a um elemento específico.
- *Placeholder="..."* serve para apresentar uma orientação ou uma sugestão ao usuário, indicando o tipo de informação que deve ser inserida em um campo de entrada.
- *Type="..."* empregado para especificar o tipo de entrada, informando que se trata de um campo destinado à inserção de texto.
- *Content="..."* determina a largura da *viewport* do dispositivo e o nível inicial de zoom da página.
- *Href="..."* especifica o local do arquivo CSS externo que será associado a página *web*.
- *Name="..."* refere-se ao identificador do campo que será transmitido no formulário.
- *Required="..."* denota que a inserção de informações no campo é indispensável.

A codificação apresentada resultará na criação de um formulário HTML básico, conforme ilustrado na figura a seguir:

Figura 2 - Demostraçao do resultado obtido aps a codificaçao em HTML.

The image shows a contact form with a black header containing the word "Contato". Below the header, there are three input fields: "Nome" (Name) with placeholder "Digite seu nome" (Type your name), "Email:" (Email) with placeholder "Digite seu email" (Type your email), and "Mensagem" (Message) with a text area for input. At the bottom of the form is a "Enviar" (Send) button.

Fonte: De autoria prpria, 2024.

2.3 CSS

De acordo com Jobstraibizer (2008), a sigla CSS significa *Cascading Style Sheets*, que em português se torna Folhas de Estilo Cascata, uma linguagem de estilo e/ou formatação para construção de *layouts* de páginas ou sites.

Conforme Scheidt (2015), para utilizar o CSS é necessário seguir uma sintaxe de três regras: o seletor, indicando o que será estilizado, a propriedade definindo o que será modificado e o valor da propriedade aps a modificação.

Figura 3 - Exemplo de codificação em CSS

```
1  *{
2      margin:0;
3      padding: 0;
4      box-sizing: border-box;
5
6  }
7
8 body{
9     font-family:Verdana, Geneva, Tahoma, sans-serif;
10    color:rgb(23, 20, 20);
11    display: flex;
12    height: 100vh;
13
14    background-color: brown;
15    justify-content: center;
16    align-items: center;
17 }
18 form{
19     background-color: beige;
20     padding: 50px;
21     border-radius: 30px;
22     display: flex;
23     flex-direction: column;
24     gap: 25px;
25
26 }
27
28 h1{
29     font-size: 24px;
30     text-align: center;
31
32
33 }
34
```

Fonte: De autoria própria, 2024.

A seguir encontra-se a continuação da codificação em CSS:

Figura 4 - Continuação do Exemplo de codificação em CSS

```

35 button{
36     padding: 15px;
37     align-self: center;
38     width: 160px;
39     border-radius: 30px;
40     background-image: linear-gradient(#5e2129, #6e3139);
41     border:solid 1px #5e2129;
42     border-right-width: 4px;
43     border: bottom width 16px ;
44     color:white;
45     font-weight: 700;
46     text-transform: uppercase;
47
48
49 }
50
51 input{
52     padding: 10px 20px;
53     border-radius: 15px;
54     border:none;
55
56
57 }
58
59
60 .campo-input {
61     display: flex;
62     flex-direction: column;
63 }
64
65
66 .titulo {
67     display: flex;
68     flex-direction: column
69 }
70
71
72 textarea {
73     width: 400px;
74     height: 120px;
75     max-width: 400px;
76     border: none;
77     border-radius: 30px;
78 }
79
80

```

Fonte: De autoria própria, 2024.

Para melhor entendimento do código acima em CSS, as *tags* presentes nele são:

- **`*`**: O seletor universal, diz que todos os elementos serão afetados pelas definições de estilos colocadas dentre as chaves.
- **`{}`**: As chaves são usadas para delimitar blocos de declarações.
- **`margin`**: Define o tamanho das quatro margens.
- **`padding`**: Define o espaço em torno de um elemento em todos os lados.

- *box sizing: border box*: Determina que não seja realizado o cálculo da dimensão de um elemento somando as bordas e margens com altura e largura.
- *body*: Define que todos os elementos presentes no corpo do documento HTML, sejam afetados pelas definições de estilos colocadas dentre as chaves.
- *font - family*: Define a família de fontes do texto apontado.
- *color*: Define a cor do texto apontado.
- *display: flex*: Essa propriedade torna o elemento em um *flex container*, para melhor manipulação de todos os elementos filhos.
- *height*: Define a altura de um elemento.
- *background - color*: Define a cor de fundo do elemento.
- *justify - content: center*: Propriedade para alinhar elementos em um container flexível.
- *gap*: Define o espaçamento de um grid ou *flexbox*.
- *form*: Define que todos os elementos presentes no formulário, sejam afetados pelas definições de estilos colocadas dentre as chaves.
- *border - radius*: Propriedade para arredondar as bordas de um elemento.
- *flex - direction: column*: Define que define que os elementos sejam organizados como colunas, de cima para baixo.
- *font - size*: Define o tamanho da fonte apontada.
- *button*: Define que todos os elementos, presentes no botão, sejam afetados pelas definições de estilos colocadas dentre as chaves.
- *align - self: center*: Propriedade para “auto alinhamento” para o centro.
- *width*: Define a largura do elemento apontado.
- *border: solid*: Torna a linha da borda pontilhada.
- *border - right - width*: Define a largura da linha da borda direita do elemento apontado.
- *border*: Define todas as propriedades da borda de uma vez só.
- *background - image*: Propriedade para definição de uma imagem apontada.
- *font - weight*: Define o peso da fonte apontada.
- *texto - transform: uppercase*: Propriedade para mudar as letras para maiúsculas de um elemento apontado.
- *Input*: Define que todos os elementos, presentes no *input*, sejam afetados pelas definições de estilos colocadas dentre as chaves.

- `.campo` - *input*: Define que todos os elementos, presentes no campo *input*, sejam afetados pelas definições de estilos colocadas dentre as chaves.
- `título`: Define que todos os elementos, presentes no título, sejam afetados pelas definições de estilos colocadas dentre as chaves.

Para a codificação em CSS funcionar, é necessário indicá-la dentro do arquivo HTML. Na tag “`<head>`”, inclua a seguinte linha. (ARAÚJO, 2023).

Figura 5 - Incorporando o CSS em página HTML

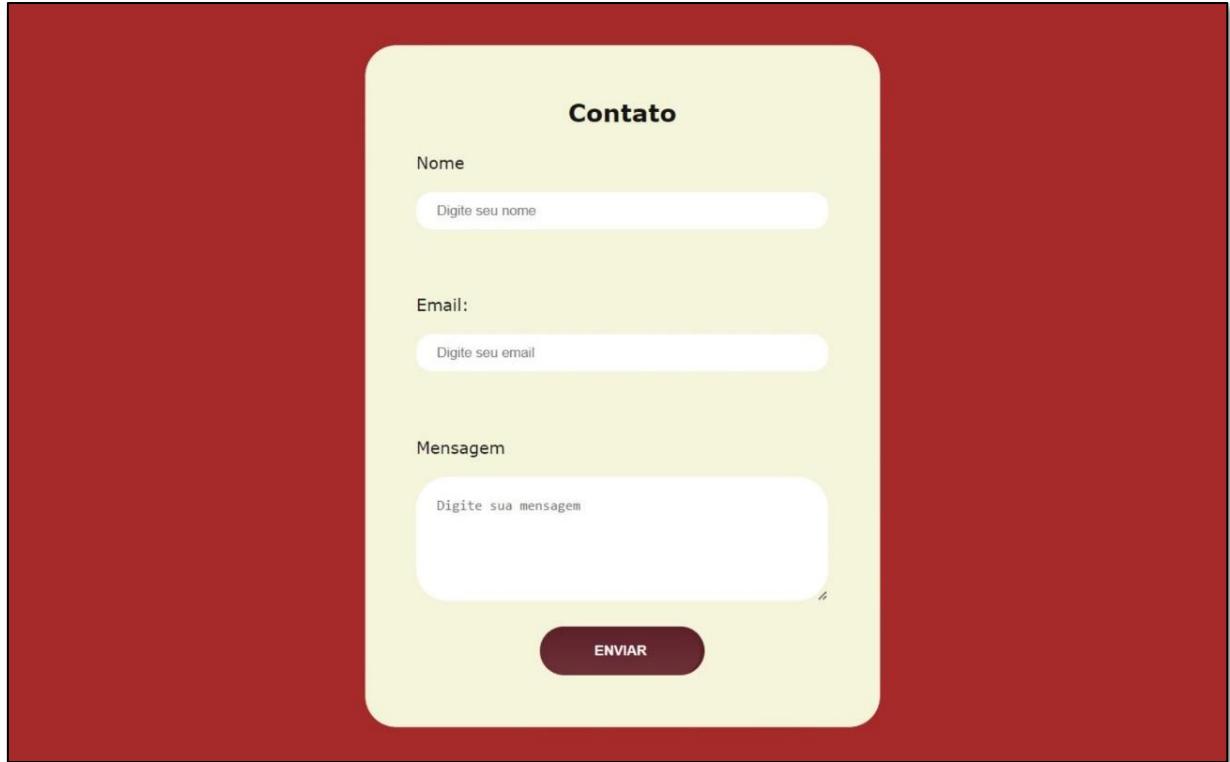
6

```
<link rel="stylesheet" href="style.css">
```

Fonte: De autoria própria, 2024.

Abaixo, segue o resultado da combinação do formulário simples codificado em HTML, com estilização em CSS:

Figura 6 - Visualização da página HTML estilizada com CSS



Fonte: De autoria própria, 2024.

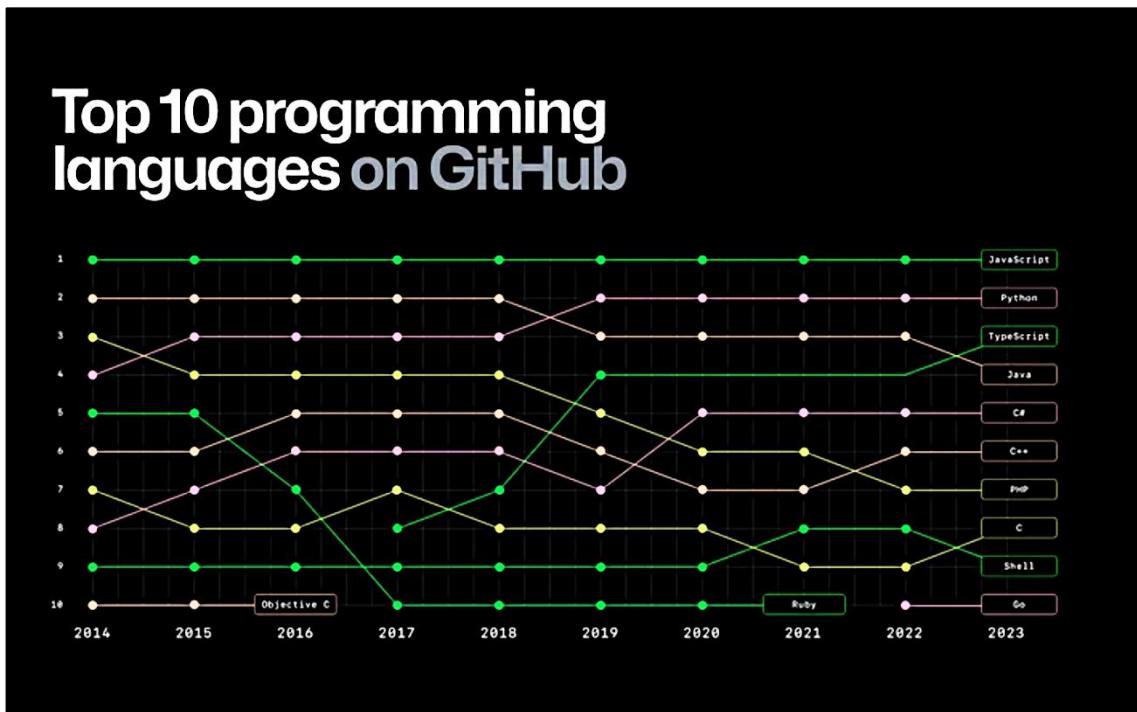
Em conclusão Maia (2011) aponta que, em dezembro de 2005 foi desenvolvido o CSS3 pela W3C (*World Wide Web Consortium*). Essa nova versão possui funcionalidades para o suporte de novos módulos de cor.

2.4 JavaScript

De acordo com sua própria documentação (MDN WEB DOCS, 2023), o JavaScript, conhecido como JS, é uma linguagem de *script* orientada a objetos, em que é utilizada na implementação de funcionalidades complexas dentro de um site.

Dada como a linguagem mais popular entre os programadores *web* da plataforma GitHub (ARRUDA, 2023), o JavaScript é considerado uma linguagem de fácil integração e utilização de recursos avançados.

Figura 7 - Linguagens de programação mais populares do GitHub.



Fonte: TecMundo, 2023.

Na figura acima, é possível ver que o JavaScript está liderando o *ranking* das melhores linguagens para programação, assim mostrando a quanto desenvolvida essa linguagem se tornou.

Em concordância com Flanagan (2012), a maioria dos códigos em JavaScript é composta por funções que são estabelecidas através de um bloco de código que pode ser definido e executado várias vezes.

Figura 8 - Exemplo de codificação em JavaScript utilizando funções

```

1  const form = document.getElementById('form');
2  const campos = document.querySelectorAll('#form input, #form textarea');
3  const spans = document.querySelectorAll('.span-required');
4  const emailRegex = /^w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
5
6  form.addEventListener('submit', (event) => {
7      event.preventDefault();
8      nameValidate();
9      emailValidate();
10     messageValidate();
11 });
12
13 function setError(element, index) {
14     element.style.border = '2px solid #e63636';
15     spans[index].style.display = 'block';
16 }
17
18 function removeError(element, index) {
19     element.style.border = '';
20     spans[index].style.display = 'none';
21 }
22
23 function nameValidate() {
24     const nameInput = campos[0];
25     if (nameInput.value.length < 3) {
26         setError(nameInput, 0);
27     } else {
28         removeError(nameInput, 0);
29     }
30 }
31
32 function emailValidate() {
33     const emailInput = campos[1];
34     if (!emailRegex.test(emailInput.value)) {
35         setError(emailInput, 1);
36     } else {
37         removeError(emailInput, 1);
38     }
39 }
40
41 function messageValidate() {
42     const messageInput = campos[2];
43     if (messageInput.value.length < 10) {
44         setError(messageInput, 2);
45     } else {
46         removeError(messageInput, 2);
47     }
48 }
49

```

Fonte: De autoria própria, 2024.

Na figura acima as funções de validação como “nameValidate”, “emailValidate” e “messageValidate” são usadas para verificar se os dados recebidos do usuário são válidos de acordo com os critérios determinados dentro da função como no exemplo “nameValidate”, se o nome não possuir no mínimo três caracteres, uma mensagem de erro pode ser exibida ao usuário e assim para as demais funções, como na figura abaixo:

Figura 9 - Resultado da codificação em JavaScript

The screenshot shows a contact form titled "Contato". It has three fields: "Nome" (Name) containing "An", "Email:" containing "ana", and "Mensagem" (Message) containing "Formulári". Below each field is a red error message: "Nome deve ter no mínimo 3 caracteres" (Name must have at least 3 characters) for the name field, "Digite um email válido" (Enter a valid email) for the email field, and "A mensagem deve ter pelo menos 10 caracteres." (The message must have at least 10 characters) for the message field. A large red button labeled "ENVIAR" (SEND) is at the bottom.

Fonte: De autoria própria, 2024.

Para a codificação em JavaScript acima funcionar, é necessário indicá-lo dentro do arquivo HTML. Dentro da tag “<head>”, inclua a seguinte linha. (ZAKAS, 2010).

Figura 10 - Implementando o JavaScript em uma página HTML

10

<script src="script.js"></script>

Fonte: De autoria própria, 2024.

2.5 React

Conforme sua respectiva documentação (React, 2024), o React é uma biblioteca JavaScript especializada na criação de interfaces de usuários (UI).

Essa biblioteca está entre as mais populares de JavaScript existentes na área de criação de aplicações *web*. É conhecido por ser usado no desenvolvimento de grandes plataformas como Facebook e Instagram. (SILVA, 2021).

2.6 React Native

O React Native é um *framework* de código aberto em JavaScript baseado no React.js. Esse componente melhora a experiência do usuário em aplicativos móveis e facilita sua integração com *smartphones*. (BRAGA, 2019).

Como citado por Galvão (2018), o React Native foi inicialmente desenvolvido pelo Facebook como um anúncio em 2015, que utilizou essa tecnologia para criar aplicativos na plataforma iOS e posteriormente foi lançado no Android.

Para criar interfaces usando React Native, utiliza-se exclusivamente JavaScript XML (JSX), uma extensão JavaScript semelhante ao HTML. (LEITÃO, 2019).

Figura 11 - Codificação de um formulário em React Native.

```

1 import React, { useState } from 'react';
2 import { StyleSheet, Text, View, TextInput, Button } from 'react-native';
3
4 export default function App() {
5   const [name, setName] = useState('');
6   const [email, setEmail] = useState('');
7   const [message, setMessage] = useState('');
8
9   const handleSubmit = () => {
10     console.log('Nome:', name);
11     console.log('Email:', email);
12     console.log('Mensagem:', message);
13   };
14
15   return (
16     <View style={styles.container}>
17       <Text style={styles.title}>Contato</Text>
18       <TextInput
19         style={styles.input}
20         placeholder="Nome"
21         value={name}
22         onChangeText={setName}
23       />
24       <TextInput
25         style={styles.input}
26         placeholder="Email"
27         value={email}
28         onChangeText={setEmail}
29         keyboardType="email-address"
30       />
31       <TextInput
32         style={[styles.input, styles.textArea]}
33         placeholder="Mensagem"
34         value={message}
35         onChangeText={setMessage}
36         multiline
37       />
38       <Button title="Enviar" onPress={handleSubmit} color="#5b2b29" />
39     </View>
40   );
41 }

```

Fonte: De autoria própria, 2024.

Para melhor entendimento do código acima em React Native, as *tags* presentes nele são:

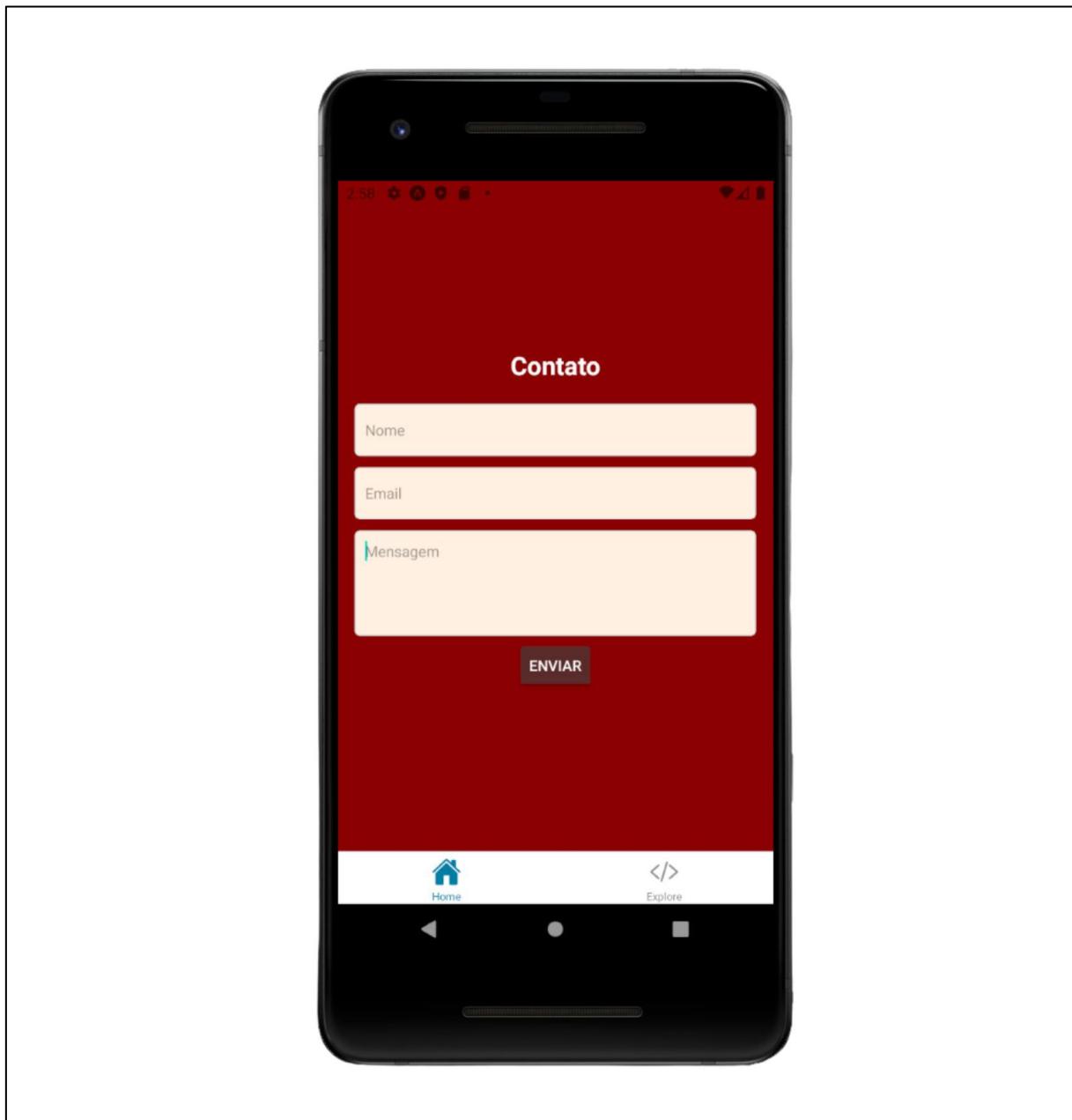
- *import React, {useState } from 'react'*: A linha está importando o React com a função *useState* do React, utilizada para gerenciar o estado de cada componente;
- *.import { StyleSheet, Text, View, TextInput, Button } from 'react-native'*: Nesta linha estão sendo importados alguns componentes do React Native que serão utilizados no aplicativo, sendo eles: *StyleSheet, Text, View, TextInput e Button*;
- *export default function App ()*: Esta função define o componente principal do aplicativo;
- *const [name, setName] = useState("")*; *const [email, setEmail] useState("")*; *const [message, setMessage] = useState("")*: Aqui estão sendo definidos os três estados usando a função *useState* do React. Cada estado é uma variável, sendo elas: *name*, *email* e *message*, as quais, mantém um valor e uma função para a sua atualização de valor;
- *const handleSubmit = () => {console.log ('Nome:', name); console.log ('Email:', email); console.log ('Mensagem:', message)}*: A função *hadleSubmit* foi configurada para ser chamada quando o botão de envio for pressionado, imprimindo os valores dos estados (*nome*, *email* e mensagem). Essa função é chamada quando o botão de envio é pressionado. Ela imprime os valores dos estados (*name*, *email* e *message*) no console;
- *return (*
<View style={styles.container}>
<Text style={styles.title}>Contato</Text>: Nesta parte se inicia o retorno do componente, sendo uma estrutura “*View*” contendo um título (“*contato*”);
- *<TextInput*
style=[styles.input]
placeholder="Nome"
value={name}

onChangeText={setName}/>: Este agrupamento apresenta uma entrada de texto para o nome, utilizando o seu respectivo estado e o atualizando quando esse texto ser alterado;

- <TextInput
style=[styles.input]
placeholder="Email"
value=[email] onChangeText={setEmail}
keyboardType="email-address"/>: Este agrupamento muito semelhante ao anterior atribui as mesmas condições para o estado “email”. Além disso é utilizado o atributo “keyboardType” é definido como “email-addres” exibindo o teclado para inserir os endereços de e-mail;
- <TextInput
style= {[styles.input, styles.textArea]}
placeholder="Mensagem"
value={message}
onChangeText={setMessage}
multiline />: O componente “TextInput” é uma área de texto multilinha para mensagem, ou seja, este componente pode ser usado como o valor de estado “mensagem” e também se atualiza quando for alterado;
- *Button title="Enviar" onPress={handleSubmit} color="#5b2b29" />*: O botão “Enviar” foi alterado para chamar a função “*handleSubmit*” quando for pressionado.

Abaixo, apresenta-se o resultado da codificação em React Native realizada:

Figura 12 - Resultado da codificação de um formulário em React Native.



Fonte: De autoria própria, 2024.

2.7 StyleSheet

O StyleSheet é uma API utilizada para estilizar componentes dentro do React Native. A classe StyleSheet possibilita a criação do objeto StyleSheet que define regras como estilização de texto e outros estilos. (ADOBE, 2018).

A imagem abaixo demonstra o código em StyleSheet CSS para a aplicação em React Native, onde cada *tag* presente no código, compõe um elemento para a formação do formulário citado anteriormente.

Figura 13 - Uso do StyleSheet para estilização do código React Native.

```
43 const styles = StyleSheet.create({
44   container: {
45     flex: 1,
46     backgroundColor: '#8B0000',
47     alignItems: 'center',
48     justifyContent: 'center',
49     padding: 16,
50   },
51   title: {
52     fontSize: 24,
53     fontWeight: 'bold',
54     marginBottom: 20,
55     color: '#ffffff',
56   },
57 },
58
59   button: {
60     backgroundColor: '#5b2b29',
61     padding: 15,
62     borderRadius: 5,
63     alignItems: 'center',
64   },
65
66   input: {
67     width: '100%',
68     padding: 10,
69     marginBottom: 10,
70     backgroundColor: '#FFF0E1',
71     borderColor: '#ccc',
72     borderWidth: 1,
73     borderRadius: 5,
74   },
75   textArea: {
76     height: 100,
77     textAlignVertical: 'top',
78   },
79 });
```

Fonte: De autoria própria, 2024.

Para melhor entendimento do código acima em StyleSheet, as *tags* CSS presentes nele são:

- `const styles = StyleSheet.create({`: Aqui se inicia a definição dos estilos em StyleSheet;
- `create.container: {`
`flex: 1,`
`backgroundColor: '#880000',`
`alignItems: 'center',`
`justifyContent: 'center',`
`padding: 16}`: Aqui estão localizados os estilos do *container* principal do aplicativo;
- `title: {`
`fontSize: 24,`
`fontWeight: 'bold',`
`marginBottom: 20,`
`color: '#ffffff'}`: Aqui estão os estilos do título "Contato";
- `input: {`
`width: 100%',`
`padding: 10,`
`marginBottom: 10,`
`backgroundColor: '#FFF8E1',`
`borderColor: '#ccc',`
`borderWidth: 1,`
`borderRadius: 5}`: Aqui estão os estilos para as estradas de texto “*name*” e “*email*” e área de entrada “*message*”;
- `textArea: {`
`height: 100,`
`textAlignVertical: 'top'}`: Aqui estão os estilos para a área de texto.

2.8 Node.js e NPM

De acordo com sua própria documentação (NODE.JS, 2024), o Node.js é um ambiente de execução JavaScript de código aberto, que oferece a criação de servidores, aplicativos *web*, ferramentas e *scripts*.

Esta plataforma foi desenvolvida em 2009 por Ryan Dahl e outros colaboradores. Ela utiliza uma arquitetura não bloqueante, permitindo que operações ocorram simultaneamente sem interromper o progresso das outras. (PEREIRA, 2014).

A arquitetura não bloqueante permite a execução paralela e eficiente de várias ações simultaneamente, em contraste com a arquitetura bloqueante, onde cada ação aguarda a conclusão da anterior. (NETO, 2017).

O Node.js é assíncrono, orientado a eventos no desenvolvimento de redes escaláveis, ou seja, sistemas que podem lidar com um número crescente de usuários, carga de trabalho e/ou volume de dados sem prejudicar sua funcionalidade. (AGUIAR, 2015).

Como ressalta Nunes (2018), dentre suas vantagens, ele apresenta leveza e multiplataforma, o que permite aos desenvolvedores rodarem seus projetos em servidores abertos e com um sistema operacional de escolha própria.

O NPM é um gerenciador de pacotes e é utilizado para instalar, atualizar e desinstalar dependências dentro de um projeto por meio de uma linha de comando executada no terminal. (SOUZA, 2020).

2.9 Banco de dados

Um banco de dados é conceituado como uma entidade destinada à coleta e organização de dados. (NIELD, 2016).

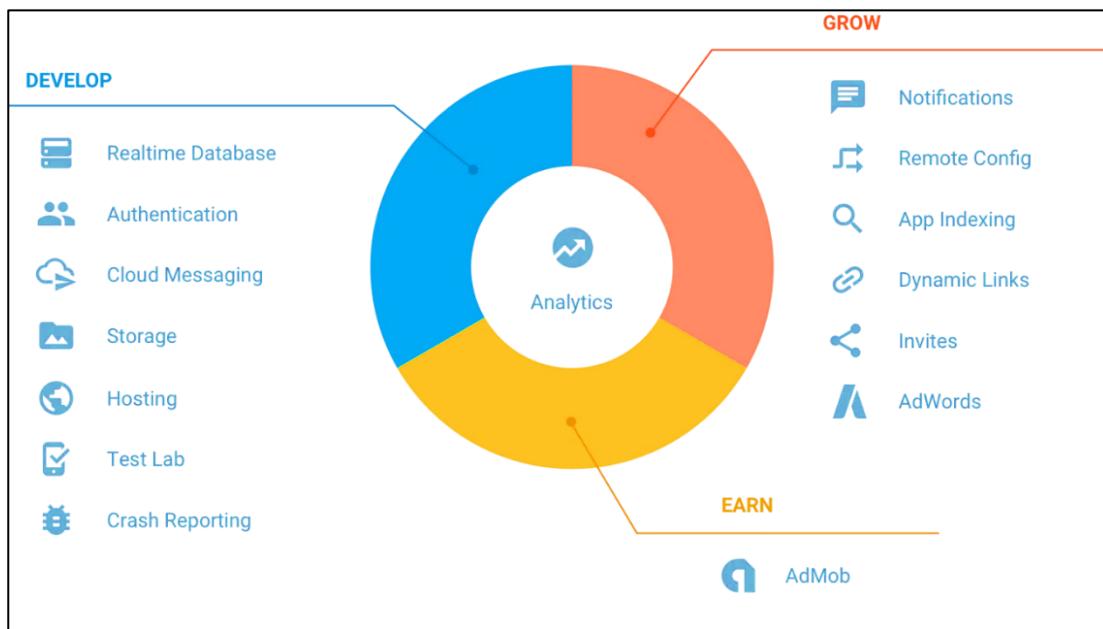
Em seus estudos, Taylor (2016) ressalta que um banco de dados comprehende tanto dados quanto metadados. Os metadados são informações que descrevem a estrutura dos dados dentro de um banco.

2.10 Firebase

A plataforma Firebase apresenta uma documentação detalhada e oferece conjuntos de desenvolvimento de software (SDKs) abrangentes, destinados à criação e distribuição de aplicativos. (FIREBASE, 2023).

Esta proporciona uma ampla gama de serviços, abrangendo desde a hospedagem de bancos de dados NoSQL até a integração com plataformas de autenticação social, como Google, Facebook, Twitter e Github. (NASCIMENTO et al., 2023).

Figura 14 - Infográfico apresentando os serviços encontrados no Firebase.



Fonte: Treinaweb, 2017.

2.11 Firestore

O conceito de NoSQL surgiu em 1998 para descrever bancos de dados sem interface SQL. Posteriormente, passou a representar soluções alternativas ao Modelo Relacional, tornando-se uma abreviação para "Not Only SQL". (BRITO, 2010).

Como banco de dados NoSQL, o Cloud Firestore facilita o desenvolvimento de aplicativos, oferecendo uma modelagem flexível dos dados. Representa as relações entre objetos e dados de diversas maneiras. (GOOGLE CLOUD, 2024).

Figura 15 - Cloud Firestore Logo.



Fonte: Medium, 2019.

2.12 UML

De acordo com Fowler (2005), a UML (*Unified Modeling Language*) é um conjunto de notações gráficas respaldada por metamodelo unificado. Seu intuito primordial consiste auxiliar na descrição do projeto de sistema de *software*.

A UML é uma linguagem destinada a estabelecer padrões na modelagem orientada a objetos, representando adequadamente qualquer sistema. Esta abordagem engloba todas as perspectivas de implementação de *software*. (Devmedia, 2011).

Conforme afirmado por Melo (2011), a programação orientada a objetos (POO) foi largamente incorporada, consolidando um novo paradigma no desenvolvimento de sistemas e inaugurando novas perspectivas para sua aplicação.

Possui padrões contemporâneos compreendendo 13 tipos diversos de diagramas, os quais são utilizados para representar graficamente um sistema, destacando seus principais atores, papéis, ações, artefatos ou classes. (NOLETO, 2022).

Os principais diagramas da UML são essenciais no planejamento e na especificação dos aspectos conceituais de sistemas, abrangendo todas as fases da engenharia envolvida na construção desses sistemas. (PEREIRA, 2011).

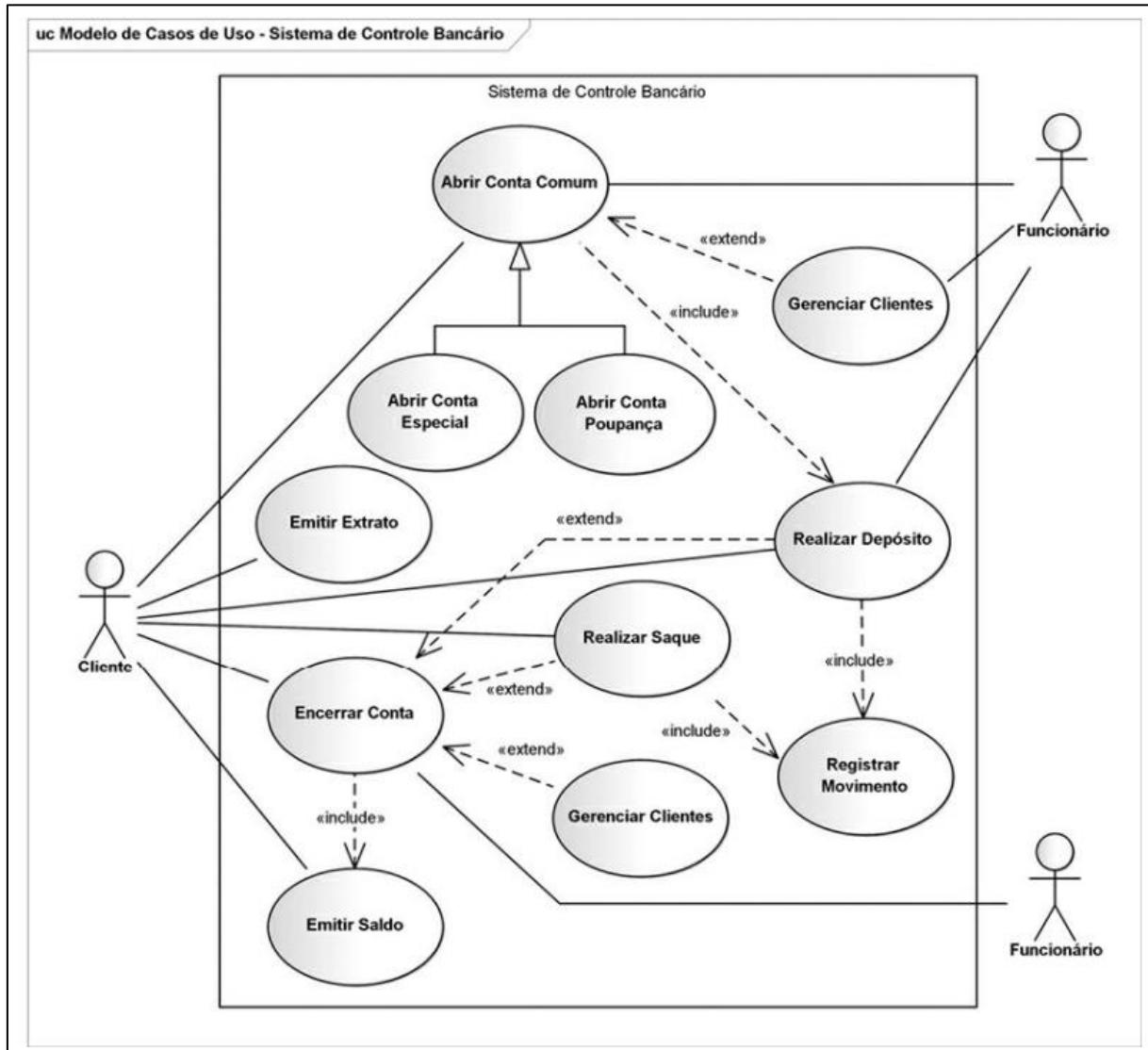
Os diagramas da UML oferecem diferentes perspectivas do sistema, desde visões gerais até detalhes específicos, abordando interações externas e estruturas internas. Cada diagrama contribui para a compreensão global do sistema. (GUEDES, 2018).

O desenvolvimento do sistema se baseia em três modelos: funcional, objeto e dinâmico. Esses modelos são representados por diagramas estruturais e comportamentais.

2.12.1 Diagrama de Casos de Uso

Silva, Martins e Diniz (2017) afirmam que o diagrama de casos de uso modela a visão estática do sistema, ilustrando relacionamentos de dependência, generalização e associação, esclarecendo funcionalidades do sistema e quem as utilizará.

Figura 16 - Exemplo de Diagrama de Caso de Uso



Fonte: Guedes, 2018.

2.12.2 Documentação de Caso de Uso

De acordo com Oliveira (2015), a documentação de caso de uso tem como objetivo documentar de forma detalhada todas as funcionalidades do sistema, incluindo suas interações dentro do *software* e os passos essenciais para situações alternativas.

Quadro 1 - Exemplo de Documentação de Caso de Uso

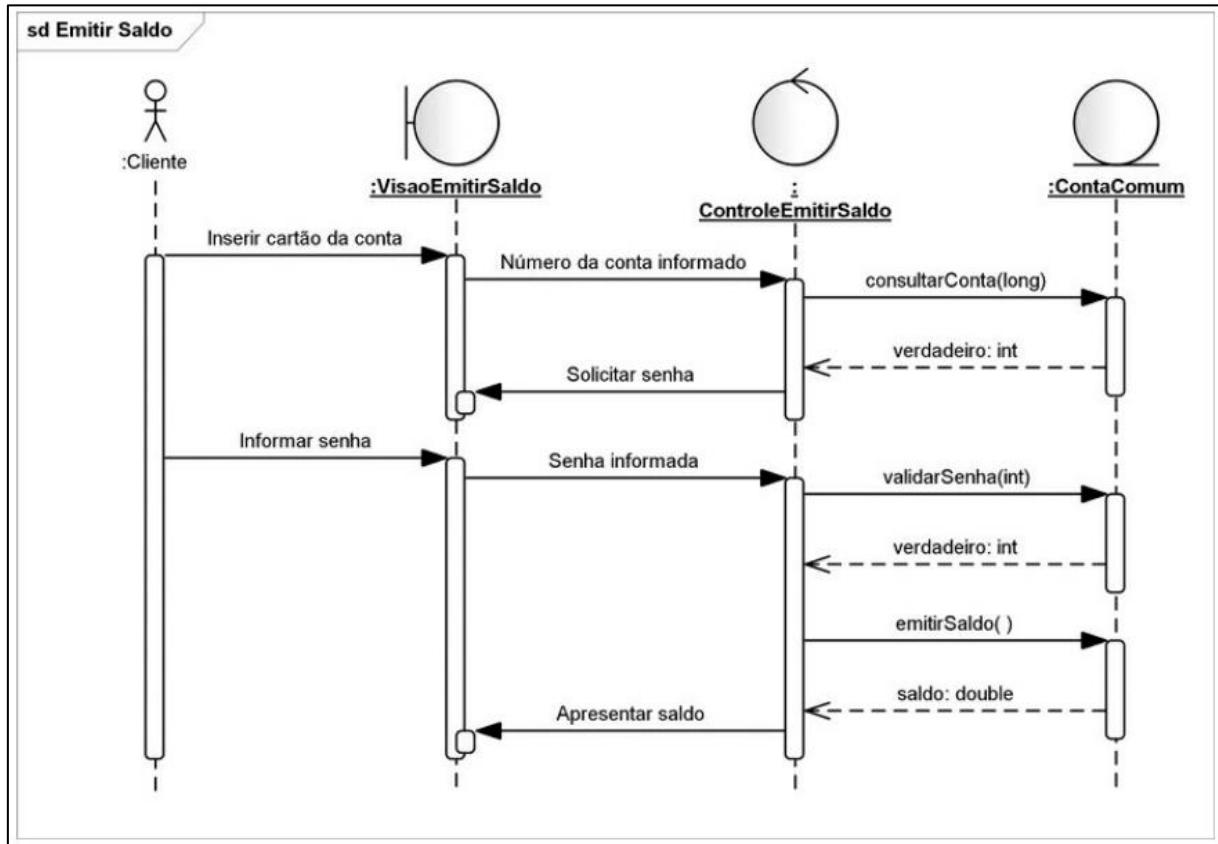
Nome do Caso de Uso		UC06 – Emitir Saldo
Ator Principal	Cliente	
Atores Secundários		
Resumo	Descreve os passos necessários para um cliente obter o saldo referente a uma determinada conta	
Pré-condições		
Pós-condições		
Ações do Ator		Cenário Principal
1. Informar o número da conta		Ações do Sistema
	2. Verificar a existência da conta	
	3. Solicitar a senha da conta	
4. Informar a senha		
	5. Verificar se a senha está correta	
	6. Emitir o saldo	
Restrições/Validações	1. A conta precisa existir e estar ativa 2. A senha precisa estar correta	
Cenário de Exceção I – Conta não encontrada		
Ações do Ator		Ações do Sistema
	1. Comunicar ao cliente que o número da conta informada não foi encontrado	
Cenário de Exceção II – Senha inválida		
Ações do Ator		Ações do Sistema
	1. Comunicar ao cliente que a senha fornecida não combina com a senha da conta	

Fonte: Guedes, 2018.

2.12.3 Diagrama de Sequência

Miguel (2019) enfatiza que o diagrama de sequência fornece uma representação visual da ordem das ações que ocorrem dentro de uma funcionalidade específica do sistema, com base no diagrama de casos de uso.

Figura 17 - Exemplo de Diagrama de Sequência

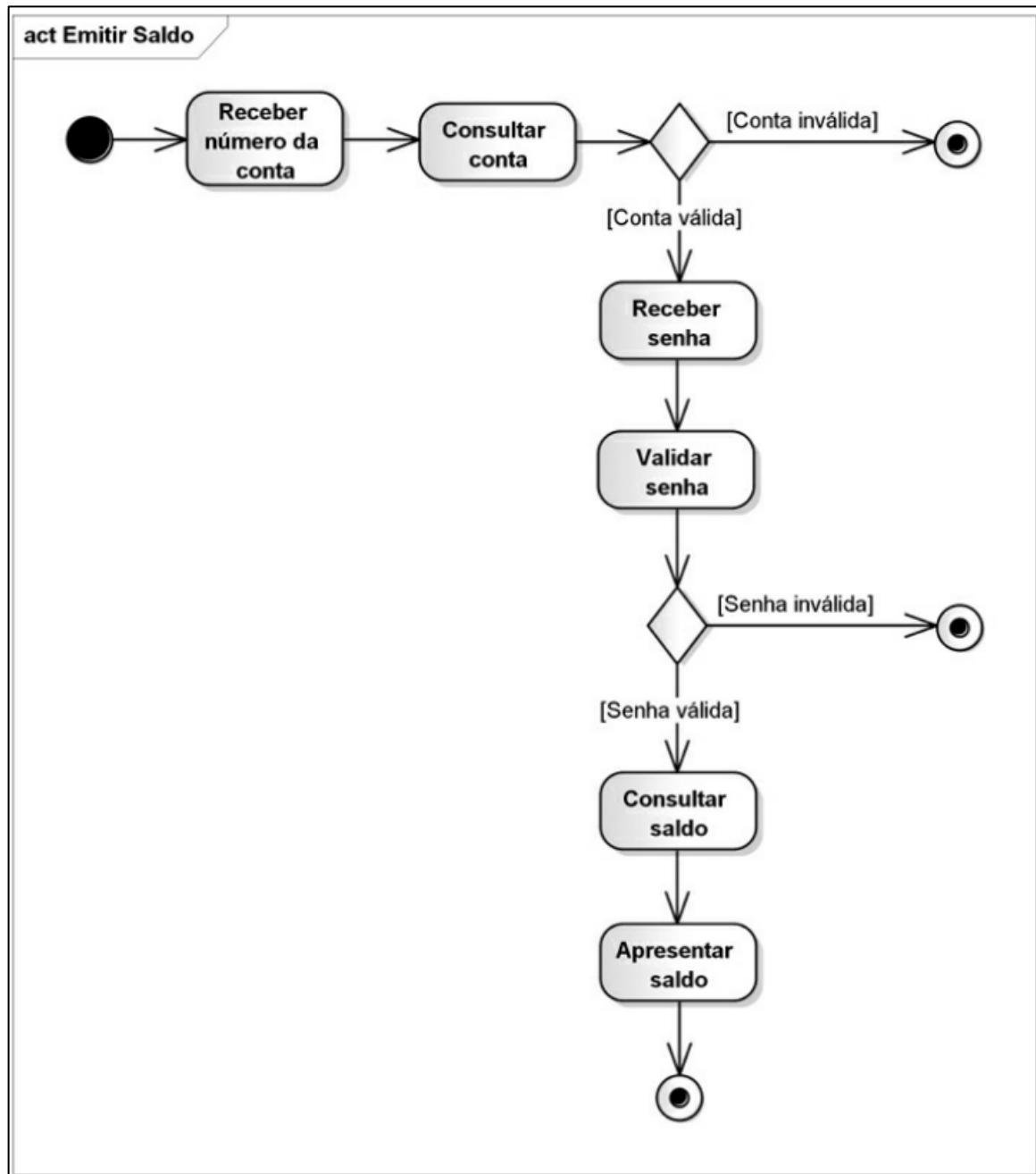


Fonte: Guedes, 2018.

2.12.4 Diagrama de Atividade

Ventura (2016) destaca que o diagrama de atividade oferece uma representação gráfica do funcionamento do software, incluindo a execução de partes específicas e a interação do sistema com o contexto do negócio em que está inserido.

Figura 18 - Exemplo de Diagrama de Atividade

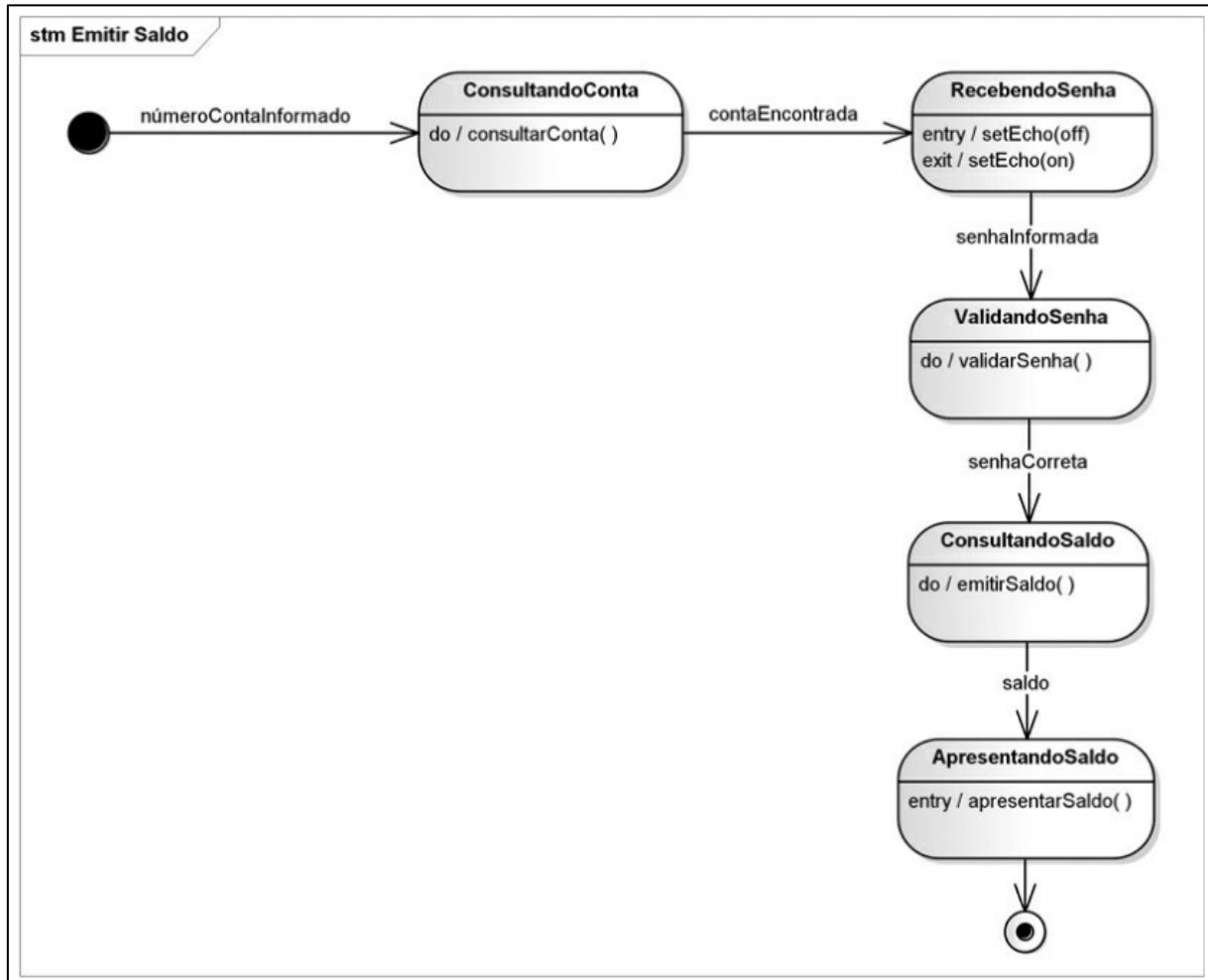


Fonte: Guedes, 2018.

2.12.5 Diagrama de Máquina de Estados

Conforme assegurado pela IBM (2021), o diagrama de máquina de estados é uma representação gráfica que ilustra a sequência de estados de um objeto, os eventos que provocam a transição entre esses estados e as ações.

Figura 19 - Exemplo de Diagrama de Máquina de Estado



Fonte: Guedes, 2018.

2.13 Wireframe

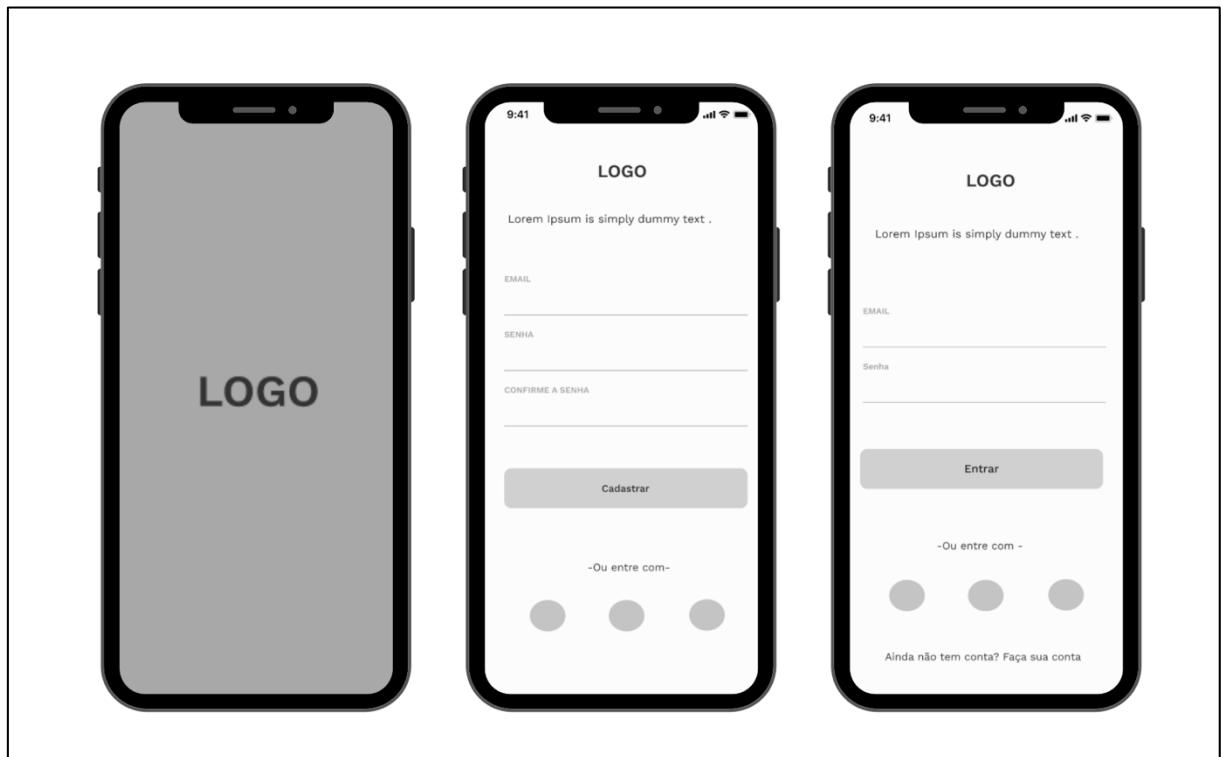
Em *web design*, o *wireframe* é uma representação visual da estrutura de uma tela, usada para o desenvolvimento da página antes de acrescentar o *design* visual e o conteúdo. (LUCIDCHART, 2024).

Elaborar *wireframes* antes de entregar um projeto, evita gastos e atrasos e torna sua alteração mais fácil e rápida, além de trazer mais tranquilidade e segurança ao cliente. (ROCKCONTENT, 2019).

Os tipos de *wireframes* mais comuns são: *wireframes* de baixa fidelidade, comuns na criação de esboços sucintos e os de alta fidelidade para uma representação visual mais completa e detalhada do projeto. (MIRO, 2024).

Nesta representação foram utilizados apenas os *wireframes* de baixa fidelidade e de alta fidelidade. Abaixo, encontra-se seus respectivos exemplos.

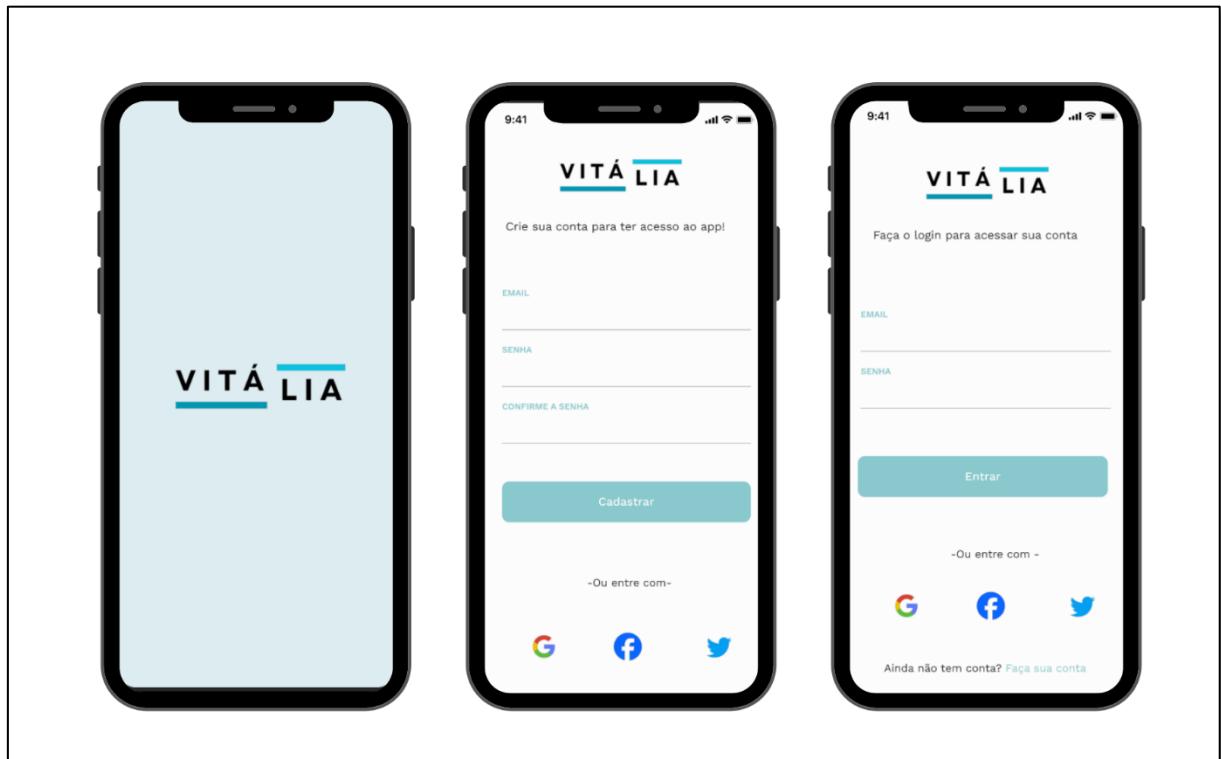
Figura 20 - Exemplo de prototipação inicial de uma tela de login e cadastro, utilizando wireframes de baixa fidelidade



Fonte: De autoria própria, 2024.

Abaixo encontra-se os *wireframes* de alta fidelidade.

Figura 21 - Exemplo de prototipação inicial de uma tela de login e cadastro, utilizando wireframes de Alta fidelidade



Fonte: De autoria própria, 2024.

2.14 Figma

Para construção dos *wireframes* desse projeto, será utilizado o *software* de desenvolvimento Figma.

O Figma é uma ferramenta desenvolvida pela empresa Figma Inc, e é utilizada para a construção de interfaces, com o objetivo de tornar a prática de design mais acessível a qualquer navegador. (LOPES, 2022).

A principal vantagem dessa ferramenta é o seu caráter multiplataforma, que permite que qualquer sistema operacional ou computador acesse o Figma. (HOSTINGER, 2023).

Outra vantagem de utilizar o Figma é o trabalho colaborativo, sendo possível convidar outros usuários para colaborar em um único arquivo, visualizando e editando em tempo real. (AWARI, 2023).

2.15 C++

A linguagem C++ é uma linguagem de programação orientada a objetos, que possui mecanismo classe/objeto, baseada em C e em Simula, criada por Bjarne Stroustrup, em 1980. (COUTO; VIRTUOSO; MARTINS, 2011).

Ricarte (1995) menciona que sistemas em linguagem C tinham muitas linhas de código, o que dificultava a compreensão. Em 1980, Bjarne Stroustrup criou "C com classes", que se tornou C++ em 1983.

Esta linguagem de programação é de elevado nível, apresentando maior flexibilidade para lidar com interfaces *hardware-software* com eficiência, economia e poder em comparação com a linguagem C. (WIENER; PINSON, 1991).

Figura 22 - Codificação C++ para ESP32

```

1 #define LEDPIN 2
2
3 void setup(){
4
5   pinMode(LEDPIN, OUTPUT);
6 }
7
8 void loop (){
9
10   digitalWrite(LEDPIN, HIGH);
11   delay(100);
12
13   digitalWrite(LEDPIN, LOW);
14   delay(100);
15
16
17 }
18

```

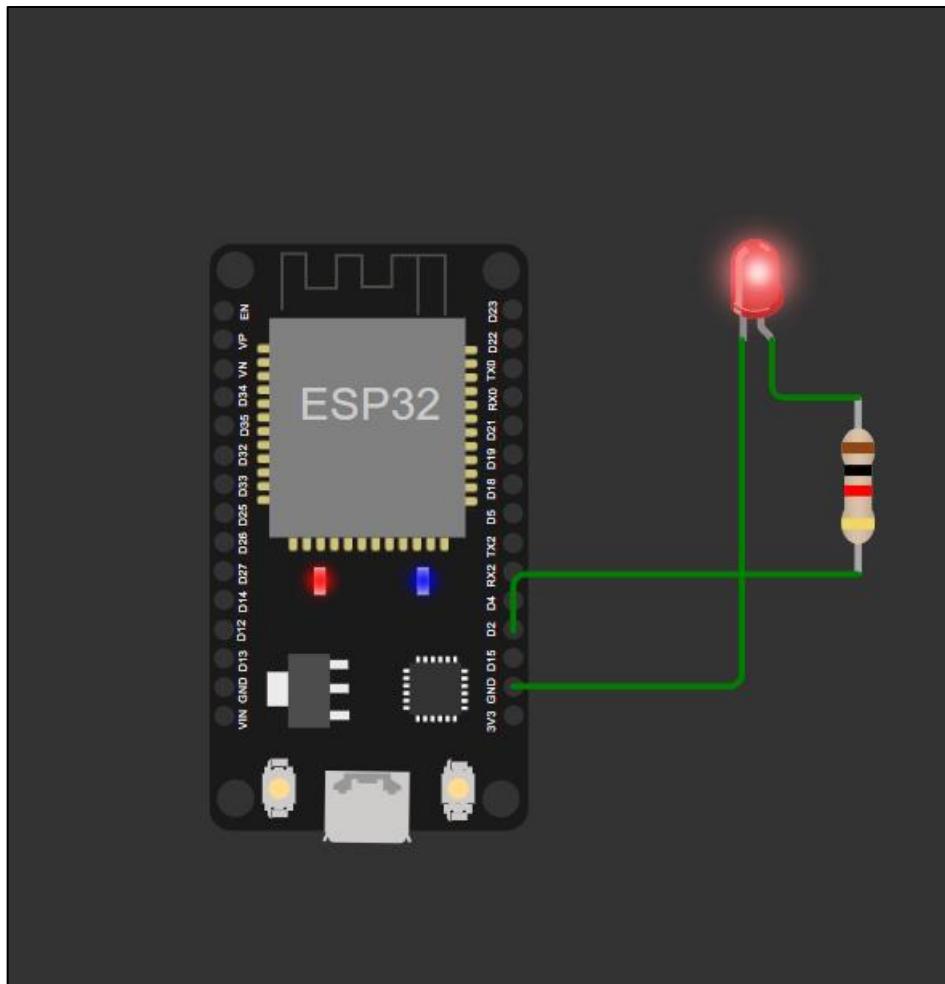
Fonte: De autoria própria, 2024.

Para melhor entendimento do código acima em C++, as *tags* presentes nele são:

- `#define` - é uma diretiva de pré-processador utilizada para definir uma variável dentro do código. Sendo utilizado para constantes simbólicas, macros de função simples ou para ativar/desativar partes do código com base em condições. Neste código, ela está sendo utilizada para definir que o led “LEDPIN” seja substituído pelo número 2 sempre que encontrado no código.
- `void setup ()` - Uma função que permite configuração do ambiente associadas a entradas e saídas de dados, em relação às portas analógicas e digitais.
- `pinMode` - Esta função é utilizada para configurar um pino de entrada ou saída de dados, além de ser possível a alteração do estado do pino.
- `pinMode (LEDPIN, OUTPUT)` - Neste código o pino associado ao led definido como “LEDPIN” está sendo configurado como uma saída.
- `void loop ()` - A função executa em loop, ou seja, repetitivamente os comandos e estruturas que estiverem em seu bloco de comando.
- `digitalWrite ()` - Esta função permite a configuração de retorno em um pino que foi definido como saída digital.
- `digitalWrite (LEDPIN, LOW)` - Neste código o pino associado ao led definido como “LEDPIN” está sendo configurado para que o led esteja desligado.
- `delay (100)` - Esta função é utilizada para a execução de uma pausa no programa, que neste código foi programada para 100 milissegundos.
- `digitalWrite (LEDPIN, HIGH)` - Neste código o pino associado ao led definido como “LEDPIN” está sendo configurado para o que o led esteja ligado

A imagem a seguir demonstra como o código acima é executado, acendendo um LED e apagando com o ESP32 e um resistor.

Figura 23- Resultado da codificação em C++



Fonte: De autoria própria, 2024.

2.16 IoT

De acordo com Santos (2018) o termo “Internet das Coisas” significa a conexão de dispositivos que podem detectar aspectos do mundo real, como a temperatura, e relatar dados ou agir sobre isso, se comunicando para melhorar a qualidade de vida.

O termo IoT surgiu em 1999 pelo pioneiro em tecnologia britânico, Kevin Ashton, em uma apresentação para executivos de uma empresa, sobre a utilização da tecnologia RFID que utiliza radiofrequência em produtos. (PAIXÃO, 2024).

Um dos benefícios dessa tecnologia é transformar objetos comuns, em dispositivos inteligentes, que passam a fornecer dados telemétricos e responder a comandos externos de forma conectada. (DURAES; FERREIRA; MANZAN, 2022).

Massola e Pinto (2018) afirmam que a utilização da “Internet das Coisas” pode ser aplicada em diversas áreas, como residências, transporte, saúde, entre outros, sempre trazendo uma vantagem econômica e tecnológica na área aplicada.

2.17 Componentes

Os componentes de sistemas constituem elementos fundamentais responsáveis por executar funções específicas dentro do sistema, garantindo assim sua operacionalidade adequada. (AERO ENGENHARIA, 2023).

2.17.1 ESP32

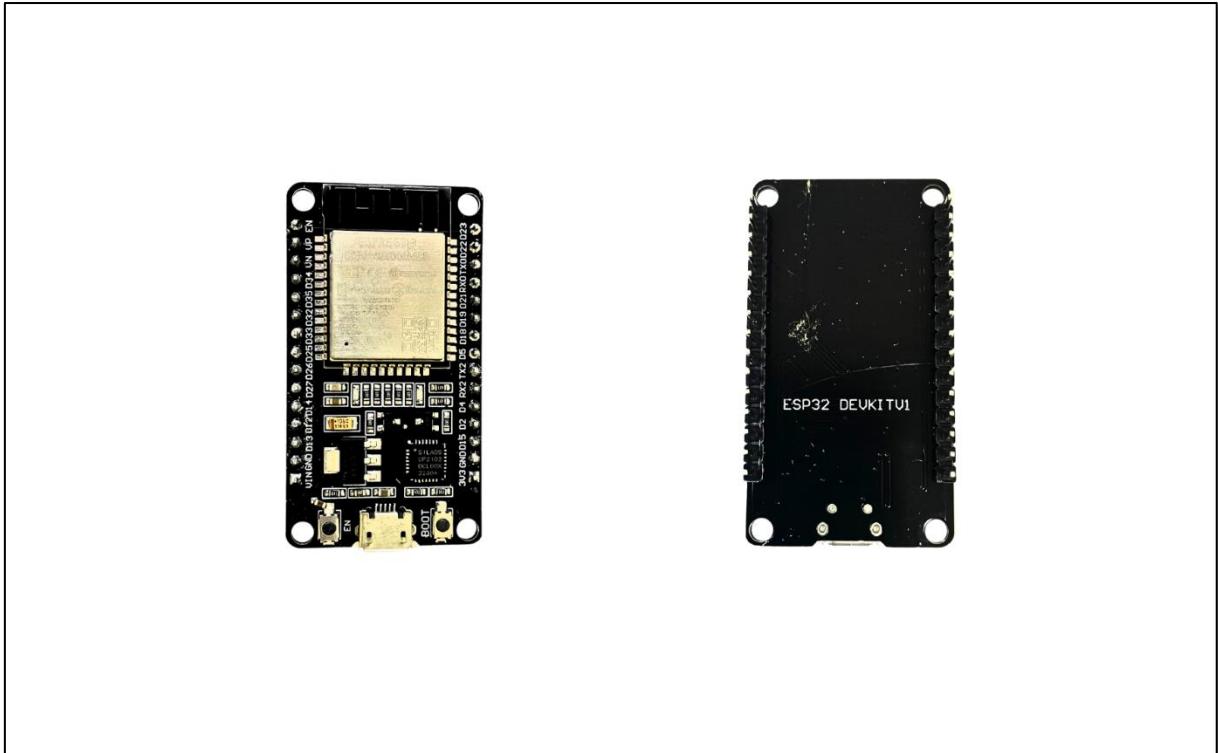
O ESP32 é uma placa de desenvolvimento de código aberto, muito semelhante ao Arduino Uno, porém esse microcontrolador detém a vantagem de utilização Wi-fi e Bluetooth nativo. (ELETRÔNICA ÔMEGA, 2021).

A utilização de sistemas embarcados, como microcontroladores com comunicação sem fio, para a coleta de dados e o controle de dispositivos, permite o desenvolvimento de soluções e automação de ambientes. (LIMA, 2022).

Esta placa apresenta inúmeras vantagens em relação ao Arduino, incluindo o grande poder de processamento, um bom armazenamento e a alteração do *clock* de processamento, no próprio código, para gastar menos energia. (TORELLA, 2020).

Neste projeto utilizaremos o microcontrolador ESP32:

Figura 24 - ESP32 - Dual Core



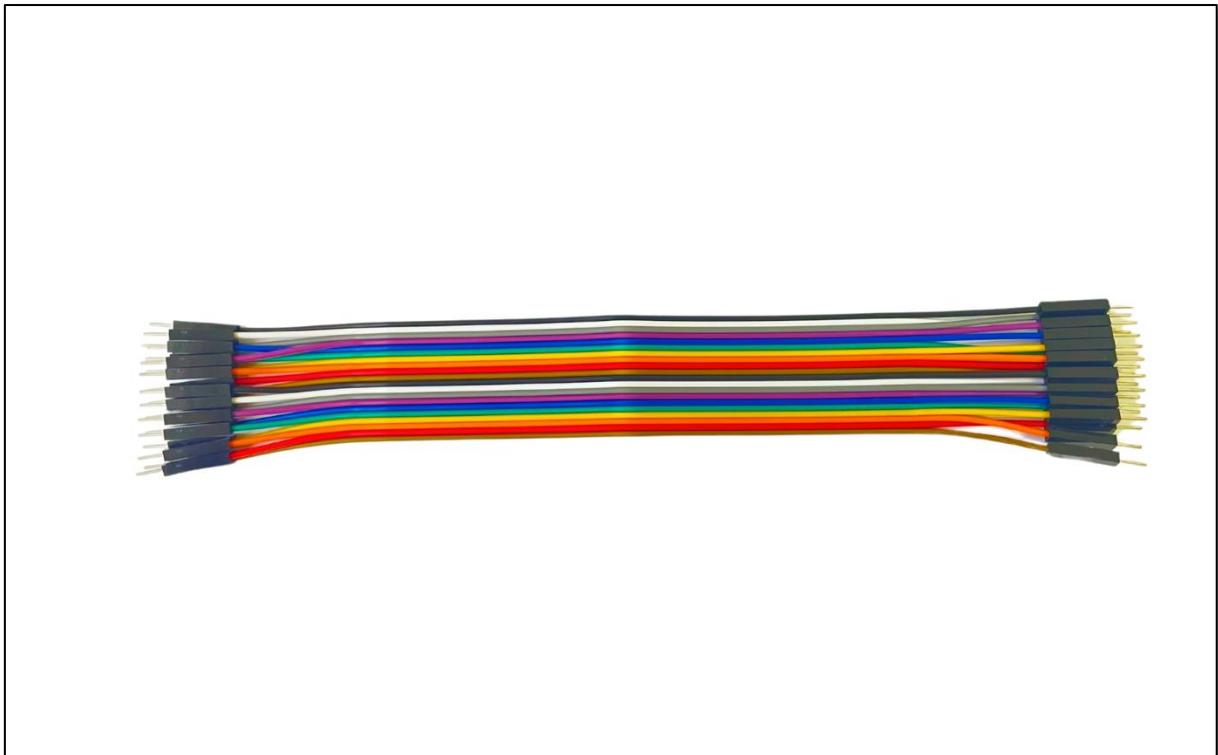
Fonte: De autoria própria, 2024.

2.17.2 Jumpers

Jumpers são fios, que podem ter diversas cores, tamanhos e espessuras, utilizados para fazer conexões entre os componentes, geralmente, entre o Arduino e a Protoboard. (MONK, 2014).

Nessa aplicação será utilizado os jumpers macho-macho devido à sua capacidade de conectar diretamente pontos específicos de interesse no circuito. Esta escolha permite uma interligação eficiente entre os diferentes elementos do circuito durante as fases de prototipagem e desenvolvimento.

Figura 25 - Jumper Macho Macho



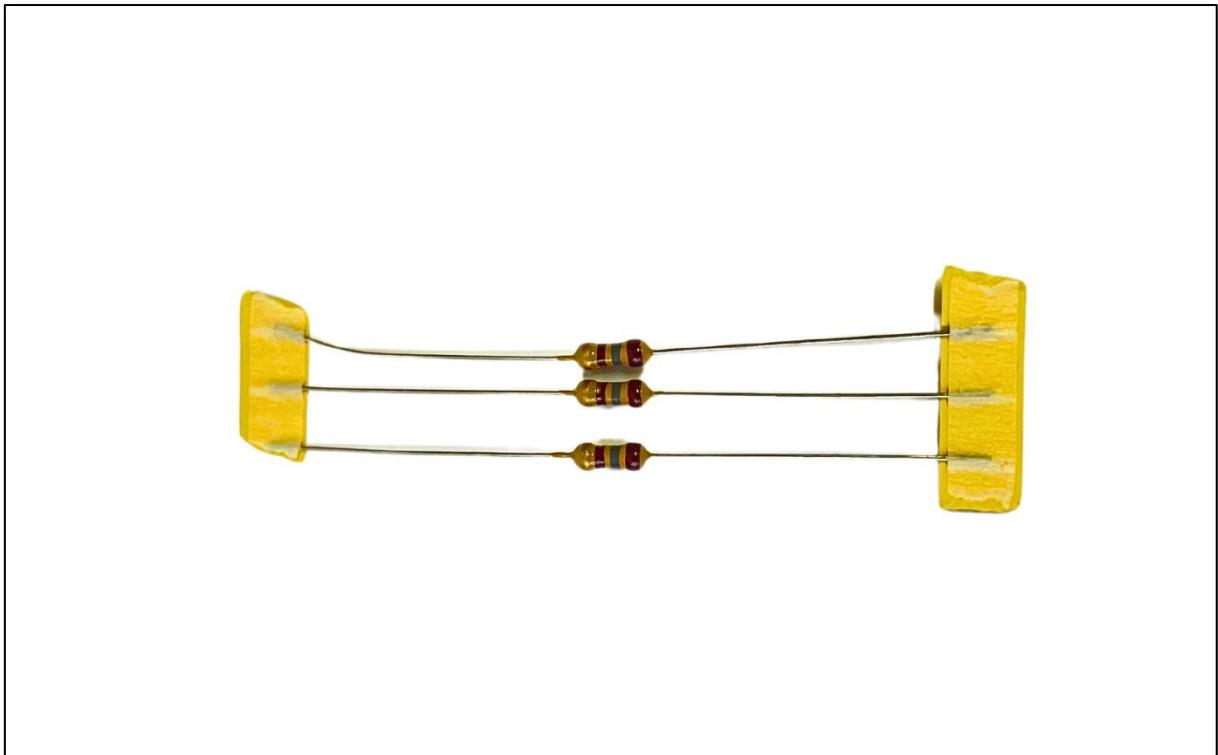
Fonte: De autoria própria, 2024.

2.17.3 Resistores

Os resistores são componentes utilizados em circuitos, para limitar a passagem da corrente elétrica entre componentes. (AMORIM; SCHEIBNER, 2018).

Será utilizado resistores de 180 ohms. Este valor foi selecionado com base em considerações técnicas específicas, visando garantir o funcionamento adequado e seguro do circuito envolvendo o LED e o ESP32.

Figura 26 - Resistores



Fonte: De autoria própria, 2024.

2.17.4 Alto Falante

O alto falante é usado para transformar energia elétrica em energia mecânica, com a capacidade de comprimir o ar, assim gerando um som. (REIS, 2021).

Figura 27 - Alto Falante – 2” YD50



Fonte: De autoria própria, 2024.

2.17.5 Módulo MP3 *DFPlayer Mini*

O Módulo MP3 *DFPlayer Mini* é utilizado no desenvolvimento de projetos que utilizam áudio e reproduzem arquivos em MP3 e WMV. A sua prototipação tem comunicação serial que facilita a integração com microcontroladores. (FERREIRA, 2023).

É compatível com as tensões 3.2V e 5,5V e possui um compartimento para armazenar um cartão microSD, organizando os áudios em pastas e permitindo uma reprodução mais flexível. (SANTOS, 2023).



2.17.6 Push Button

O Push Button ou botão de pressão é um pequeno interruptor que serve para controlar a passagem de corrente elétrica dentro do circuito. (ROSA; SANTOS; SOUZA, 2021).

Figura 28 – Push Button



Fonte: UsinalInfo, 2024.

2.18 Impressão 3D

A impressão 3D, também conhecida como fabricação aditiva, é uma tecnologia utilizada para criar um objeto a partir de um modelo digital, chamada modelagem 3D. (JÚNIOR; VAZ 2020).

A impressora 3D tem a mesma funcionalidade de uma impressora comum, porém ao invés de colocar tinta em cima de uma folha, ela cria objetos tridimensionais. (BAIÃO, 2012).

2.19 Modelagem 3D

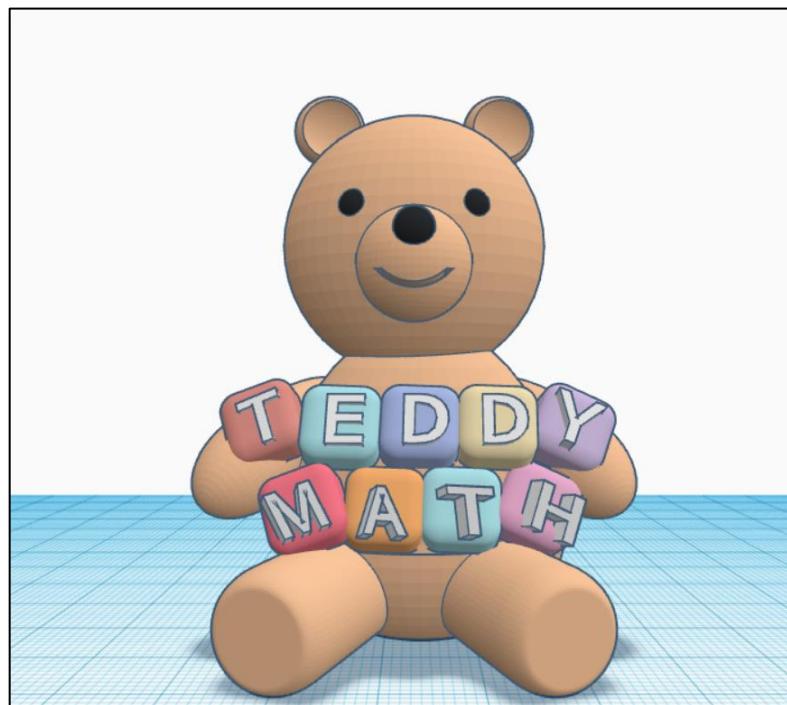
A modelagem 3D consiste na utilização de softwares para o desenvolvimento de representações matemáticas de um objeto ou forma tridimensional. (AUTODESK, 2024).

A tecnologia 3D está presente em diversas áreas, desde a indústria, para o desenvolvimento de produtos específicos e detalhados. (EBAC, 2023).

As técnicas de modelagem 3D também permitem a construção de personagens e cenários sofisticados, detalhados e realistas de jogos e filmes. (SAGA, 2020).

A seguir, apresentamos um exemplo de prototipação da logo TeddyMath através de modelagem 3D:

Figura 29 - Exemplo de Prototipação de Modelagem 3D.



Fonte: De autoria própria, 2024.

3 DESENVOLVIMENTO

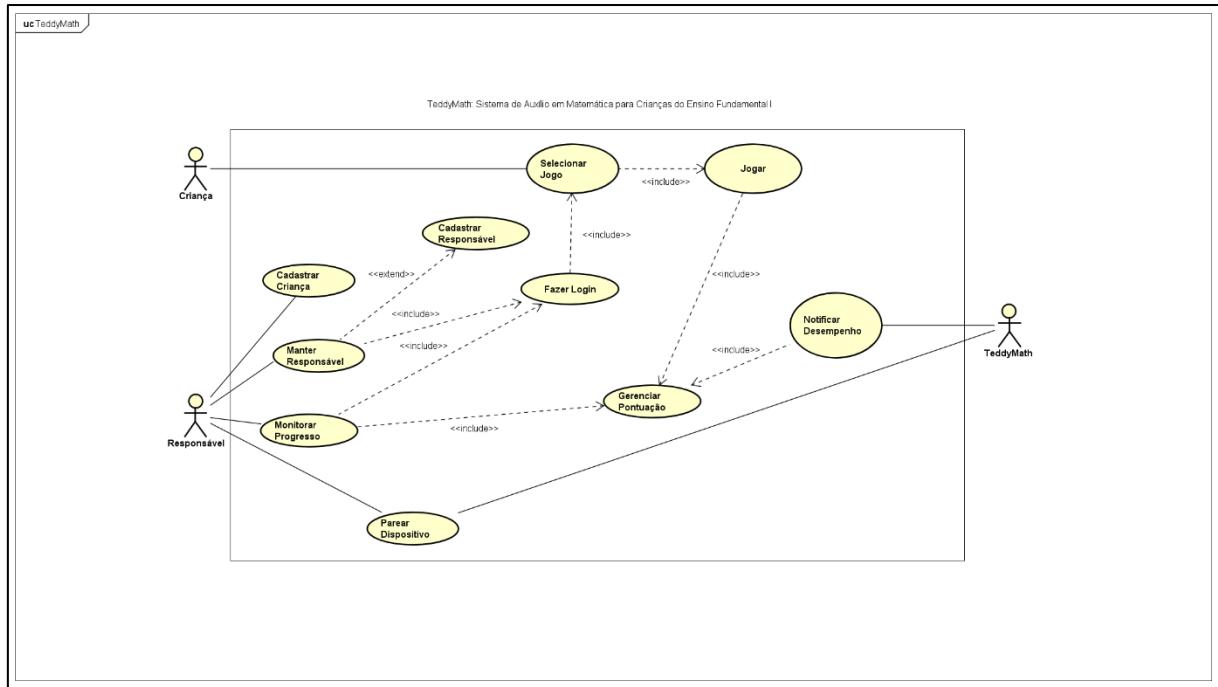
Este capítulo apresenta todo o desenvolvimento do sistema TeddyMath, incluindo a documentação UML, as imagens das telas do sistema e as explicações das documentações necessárias.

3.1 Diagrama de Casos de Uso

A seguir, apresenta-se a figura do Diagrama de Casos de Uso do sistema, com as interações entre os atores Criança, Responsável e TeddyMath que interagem com o

sistema e as ações que cada um deles podem executar. Neste diagrama, contém dez elipses de caso de uso.

Figura 30 - Diagrama de Caso de Uso do Sistema TeddyMath.



Fonte: De autoria própria, 2024.

3.1.1 Documentação dos Casos de Uso

De acordo com as normas e padrões dos módulos UML, a seguir temos a documentação de cada caso de uso. Os casos de usos são estão determinados como:

- Requisitos Funcionais, requisitos que determinam uma funcionalidade ativa no sistema.
- Requisitos Não Funcionais, requisitos não essenciais para o funcionamento do sistema, normalmente representados por características requeridas pelo usuário, sendo normalmente relacionadas à segurança, desempenho, visual, etc.
- Regras de Negócio, que como já dito, são regras e fundamentos essenciais para o funcionamento do sistema, afetando diretamente o cliente.

Abaixo apresenta os respectivos tópicos abordados referentes ao projeto:

Requisitos Funcionais da Criança:

- RF01: A criança poderá selecionar um jogo disponível no aplicativo;
- RF02: A poderá jogar qualquer jogo disponível no aplicativo;

Requisitos Funcionais do Responsável:

- RF01: O responsável poderá cadastrar uma criança no sistema;
- RF02: O responsável poderá monitorar o progresso da criança nos jogos;
- RF03: O responsável poderá cadastrar-se no sistema;
- RF04: O responsável poderá atualizar as informações do seu perfil no sistema;
- RF05: O responsável poderá visualizar seu perfil no sistema;
- RF06: O responsável poderá excluir sua conta;
- RF07: O responsável poderá fazer login na conta criada;
- RF08: O responsável poderá parear o dispositivo,inserindo seu código.

Requisitos Funcionais de TeddyMath:

- RF01: O ator TeddyMath deverá notificar a criança sobre seu desempenho nos jogos, após cada partida;

Requisitos Não-Funcionais do Sistema:

- RNF01: O website informativo deverá usar a linguagem de marcação HTML, linguagem de formatação e estilo CSS e a linguagem de script Javascript;
- RNF02: O aplicativo deverá usar a biblioteca React.js,o framework React Native,o ambiente de execução Javascript Node.js e o gerenciador de pacotes NPM;
- RNF03: Para o banco de dados, o aplicativo deverá usar a plataforma Firebase e o banco de dados Cloud Firestore;
- RNF04: O sistema deverá ser modelado usando UML;
- RNF05: O dispositivo deverá utilizar a linguagem C++;
- RNF06: O dispositivo será composto por um ESP32;
- RNF07: O dispositivo conterá um botão árcade para interatividade do dispositivo TeddyMath com o aplicativo;

- RNF08: O dispositivo conterá um Alto Falante para emitir os sons integrados ao aplicativo;
- RNF09: O dispositivo deverá conter conectividade via Bluetooth.

Regras de Negócio do Sistema:

- Somente o responsável pode cadastrar a criança no sistema;
- Somente o responsável pode monitorar o progresso da criança;
- O progresso da criança deve ser registrado a cada etapa dos jogos, a partir de sua pontuação;
- A comunicação entre o dispositivo TeddyMath e o sistema deve ser segura para evitar interceptações não autorizadas.

Quadros descritivos dos casos de uso:

Documentação dos Casos de Uso da Criança

Quadro 2 - Descrição do caso de uso: Selecionar jogo

Nome do Caso de Uso	Selecionar jogo
Ator Principal	Criança
Resumo	Este caso de uso descreve a ação de selecionar um jogo
Pré-condições	1. A criança deve estar logada no sistema
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Fazer login	
2. Acessar ao catálogo de jogos	
	2. Exibição dos jogos disponíveis no aplicativo
3. Selecionar jogo	

	4. Exibição do jogo selecionado
--	---------------------------------

Fonte: De autoria própria, 2024.

Quadro 3 - Descrição do caso de uso: Jogar

Nome do Caso de Uso	Jogar
Ator Principal	Criança
Resumo	Este caso de uso descreve a ação de jogar algum jogo disponível no aplicativo
Pré-condições	1. A criança deve estar cadastrada no sistema
Pós-condições	1. Após finalizar qualquer partida, o sistema salva sua pontuação automaticamente
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Selecionar jogo	
	2. Exibição do jogo selecionado
3. Jogar	
	4. Salvar pontuação

Fonte: De autoria própria, 2024.

Documentação dos Casos de Uso Do Responsável

Quadro 4 - Descrição de caso de uso: Cadastrar criança

Nome do Caso de Uso	Cadastrar criança
Ator Principal	Responsável

Resumo	Este caso de uso descreve o processo de cadastramento inicial da criança
Pós-condições	1. A criança terá acesso ao sistema e poderá acessar todos os jogos disponíveis
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessar área de cadastramento da criança	
	2. Solicitação dos dados necessários
3. Inserir dados necessários para cadastrar a criança	
	4. Realizar cadastro
Cenário Alternativo - Dados incorretos	
Ações do Ator	Ações do Sistema
	1. Exibir mensagem de erro
	2. Solicitar dados necessários novamente

Fonte: De autoria própria, 2024.

Quadro 5 - Descrição do caso de uso: Manter Responsável.

Nome do Caso de Uso	Manter Responsável
Ator Principal	Responsável
Resumo	Este caso de uso permite ao responsável gerenciar seu perfil, realizando operações de criação, leitura, atualização e exclusão dos dados associados.

Pré-condições	1. O responsável deve ter uma conta cadastrada no sistema
Pós-condições	1. O perfil do usuário é criado, atualizado ou excluído, conforme a operação escolhida
Cenário – Criar Conta	
Ações do Ator	Ações do Sistema
1. Acessar a aba perfil	
	2. Exibição da área de perfil
3. Selecionar opção criar perfil	
	4. Exibição do formulário para preenchimento de dados
5. Preenchimento do formulário	
6. Envio do formulário	
	7. Verificação dos dados informados
	8. Confirmação dos dados fornecidos
	9. Criar Perfil
	10. Exibição do perfil criado
Cenário – Ler Perfil	
Ações do Ator	Ações do Sistema
1. Acessar Perfil	
	2. Exibição da área de perfil
Cenário – Editar Perfil	
Ações do Ator	Ações do Sistema
1. Acessar Perfil	
	2. Exibição da área da perfil

3. Selecionar opção de atualização de dados	
	4. Exibição dos dados atuais
	5. Exibição do formulário para modificação dos dados
6. Inserir dados atualizados	
7. Envio dos dados	
	8. Verificação dos novos dados informados
	9. Confirmação da atualização com sucesso
	10. Atualizar dados
	11. Exibição dos dados atualizados
Cenário – Excluir Conta	
1. Acessar Perfil	
	2. Exibição da área do perfil
3. Selecionar opção de exclusão de perfil	
	4. Solicitação de confirmação de exclusão de perfil
5. Confirmar exclusão	
	6. Excluir Perfil
	7. Redirecionamento para área de login

Fonte: De autoria própria, 2024.

Quadro 6 - Descrição de caso de uso: Cadastrar responsável

Nome do Caso de Uso	Cadastrar Responsável
Autor Principal	Responsável
Resumo	Este caso de uso descreve o processo de cadastramento do responsável no sistema
Pós-condições	1. O responsável terá acesso ao seu perfil 2. O responsável poderá acessar a área de progresso da criança
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessar área de cadastro	
	2. Solicitação dos dados necessários
3. Inserir dados necessários para cadastro	
	4. Realizar cadastro
Cenário Alternativo - Dados incorretos	
Ações do Ator	Ações do Sistema
	1. Exibir mensagem de erro
	2. Solicitar dados necessários novamente

Fonte: De autoria própria, 2024.

Quadro 7 - Descrição do caso de uso: Monitorar Progresso

Nome do Caso de Uso	Monitorar Progresso
Autor Principal	Responsável
Resumo	Este caso de uso descreve o processo monitorar o progresso de desempenho da criança nos jogos
Pré-condições	1. O responsável deve estar logado na sua conta 2. A pontuação do jogo que a criança jogou deve ter sido salva
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessar área do responsável	
	2. Exibição da área do responsável
3. Selecionar opção de monitorar progresso da criança	
	4. Buscar pontuação em cada jogo que a criança jogou
	5. Exibir progresso da criança em cada jogo
6. Analisar progresso da criança	

Fonte: De autoria própria, 2024.

Quadro 8 - Descrição do caso de uso: Gerenciar Pontuação

Nome do Caso de Uso	Gerenciar Pontuação
Autor Principal	Responsável
Autor Secundário	Criança
Resumo	Este caso de uso descreve o processo automático de salvar a pontuação dos jogos que a criança jogou, após cada partida
Pré-condições	1. A criança deve ter jogado algum jogo
Cenário Principal	
Ações do Ator	Ações do Sistema
1. A criança encerra a partida de algum jogo	
	2. Calcular pontuação
	3. Salvar pontuação no banco de dados
	4. Exibir pontuação

Fonte: De autoria própria, 2024.

Quadro 9 - Descrição do caso de uso: Parear Dispositivo

Nome do Caso de Uso	Parear Dispositivo
Autor Principal	Responsável
Autor Secundário	TeddyMath
Resumo	Este caso de uso descreve o processo de parear o dispositivo e conectá-lo ao sistema.
Pós-condições	1. Após a conexão com o dispositivo a criança tem acesso ao jogos
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessar aplicativo	
	2. Exibição da tela para inserir o código do dispositivo
3. Inserir o código	
	4. Verificação do código informado
	5. Confirmar validação
	6. Exibir mensagem de sucesso
	6. Redirecionamento para o catálogo de jogos
Cenário Alternativo- Código Inválido	
Ações do Ator	Ações do Sistema
	1. Informar que o código informado não existe
	2. Negar acesso
	3. Solicitar código novamente

Fonte: De autoria própria, 2024.

Quadro 10 - Descrição do caso de uso: Fazer Login

Nome do Caso de Uso	Fazer Login
Autor Principal	Responsável
Resumo	Este caso de uso descreve o processo de login de uma conta criada
Pré-condições	1. O responsável deve ter um cadastro no sistema para efetuar o login.
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Acessar área de login	
	2. Exibir opções de login
3. Selecionar login com conta interna	
	4. Solicitação dos dados necessários
5. Inserir dados necessários para o login	
	6. Verificação dos dados informados
	7. Realizar login
Cenário Alternativo- Conta Externa	
Ações do Ator	Ações do Sistema
	1. Solicitação dos dados para login
2. Inserir dados de login	
	3. Verificação dos dados informados
	4. Realizar login
Cenário Alternativo- Dados incorretos	
Ações do Ator	Ações do Sistema
	1. Informar que os dados estão incorretos
	2. Negar login
	3. Solicitação dos dados para login novamente

Fonte: De autoria própria, 2024.

Documentação dos Casos de Uso do TeddyMath.

Quadro 11 - Descrição do caso de uso: Notificar Desempenho

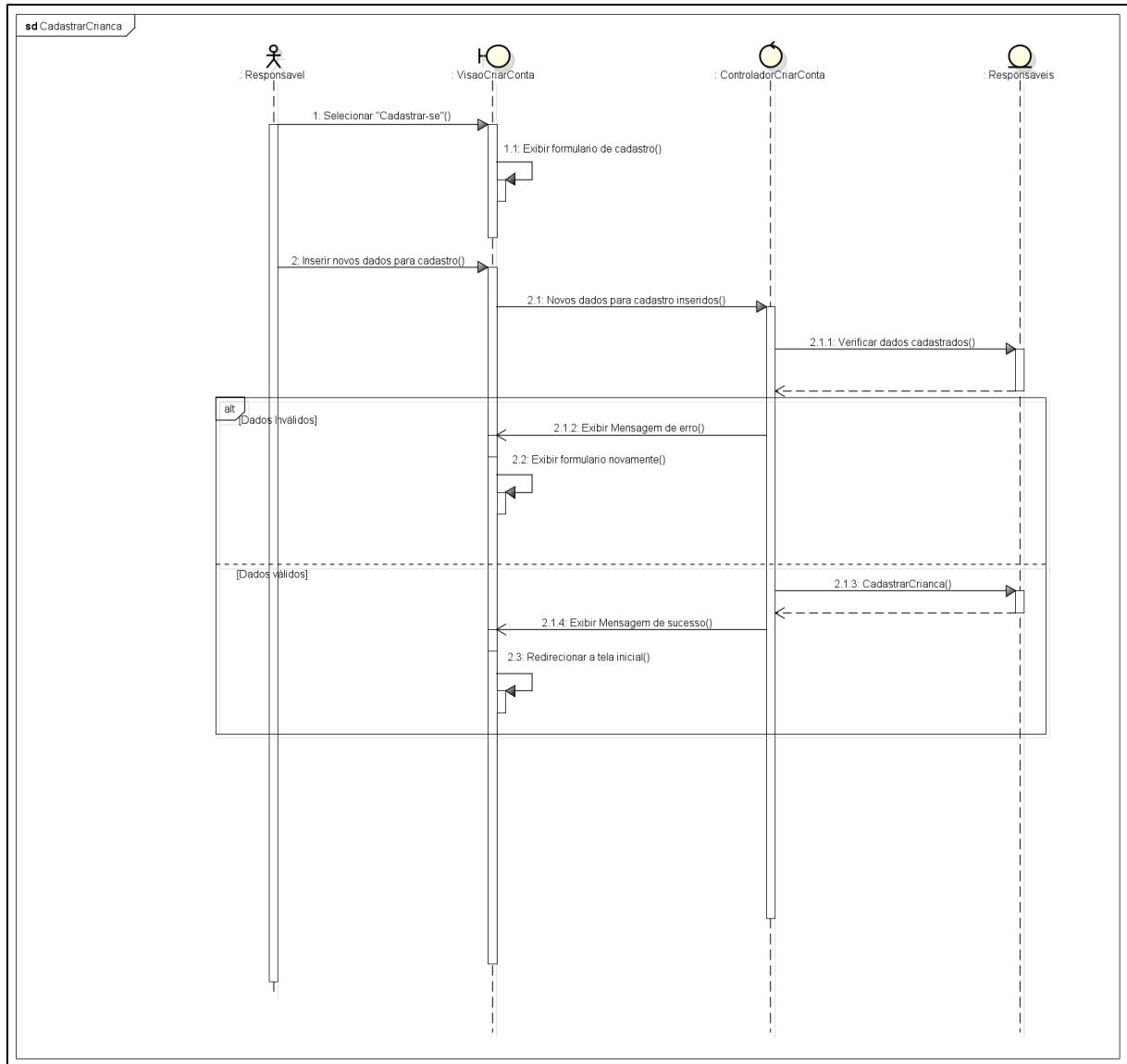
Nome do Caso de Uso	Notificar desempenho
Autor Principal	TeddyMath
Autor Secundário	Criança
Resumo	Este caso de uso descreve a ação do dispositivo TeddyMath de emitir um feedback em áudio à criança sobre seu desempenho em um jogo
Pré-condições	<ol style="list-style-type: none"> 1. A criança deve ter jogado algum jogo 2. A pontuação do jogo que a criança jogou deve ter sido salva
Cenário Principal	
Ações do Ator	Ações do Sistema
1. A criança encerra a partida de algum jogo	
	2. Buscar a pontuação da criança no jogo
	3. Emitir feedback em áudio
Cenário Alternativo - Pontuação Alta	
Ações do Ator	Ações do Sistema
	1. Emitir feedback em áudio sobre bom desempenho
Cenário Alternativo - Pontuação Baixa	
Ações do Ator	Ações do Sistema
	1. Emitir feedback em áudio sobre mal desempenho

Fonte: De autoria própria, 2024.

3.2 Diagrama de Sequência

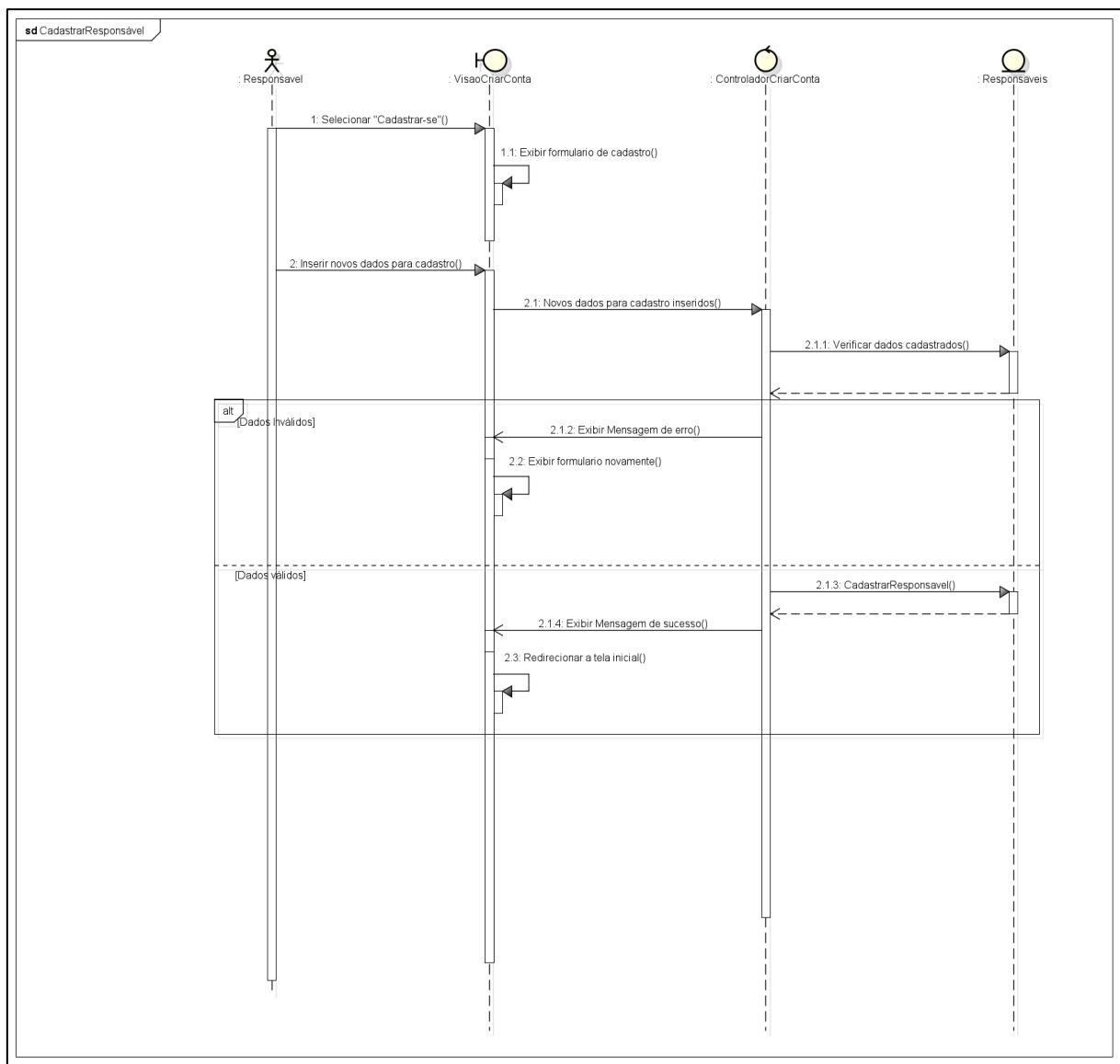
Adiante temos os Diagramas de Sequências, os quais são utilizados para demonstrar o comportamento detalhado de um cenário dentro do sistema, segundo sua ordem temporal e suas interações com todas as partes necessárias para que a função seja realizada com sucesso. Abaixo estão os diagramas sequência do projeto:

Figura 31- Diagrama de Sequência: Cadastrar Criança



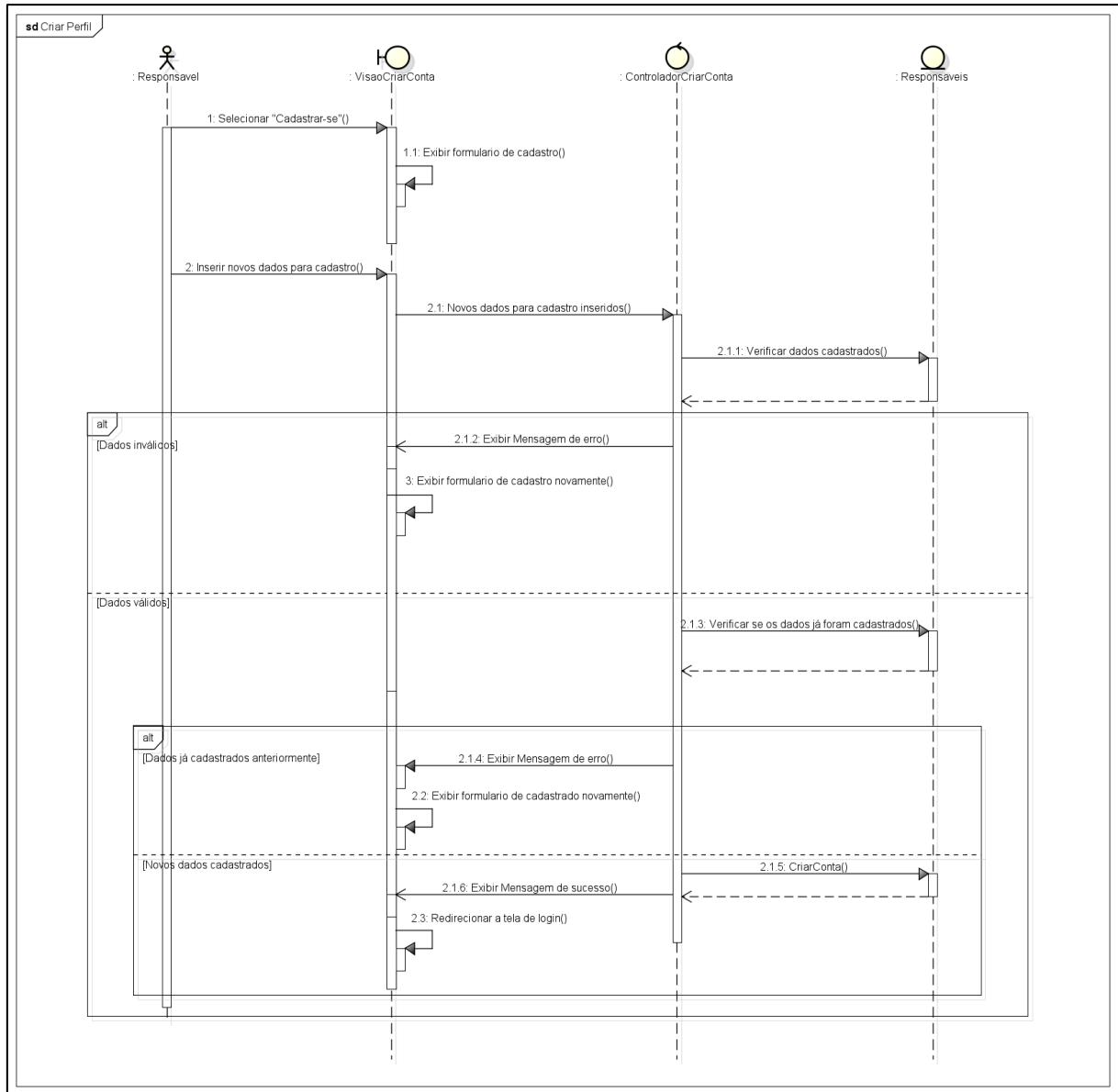
Fonte: De autoria própria, 2024.

Figura 32 - Diagrama de Sequência: Cadastrar Responsável



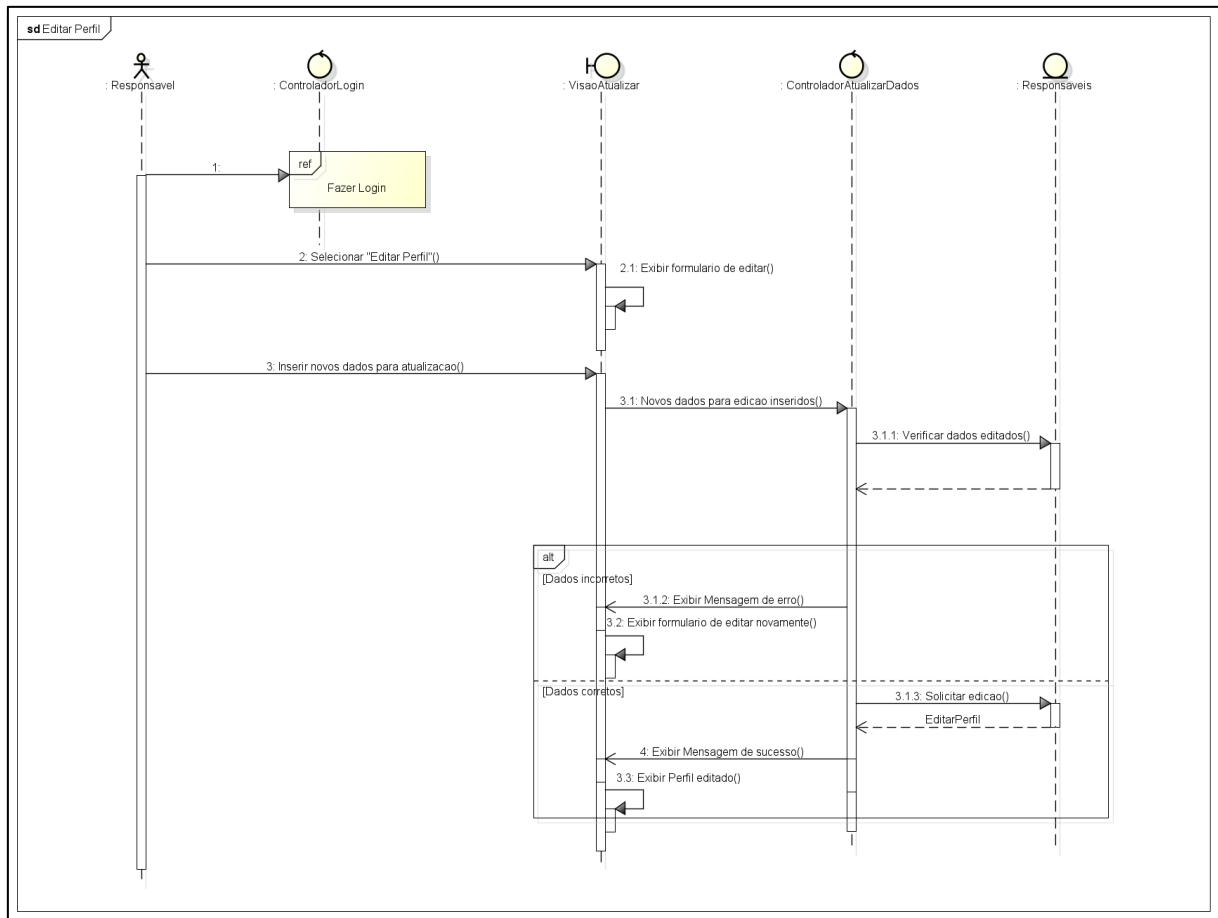
Fonte: De autoria própria, 2024.

Figura 33 - Diagrama de Sequência: Criar Perfil



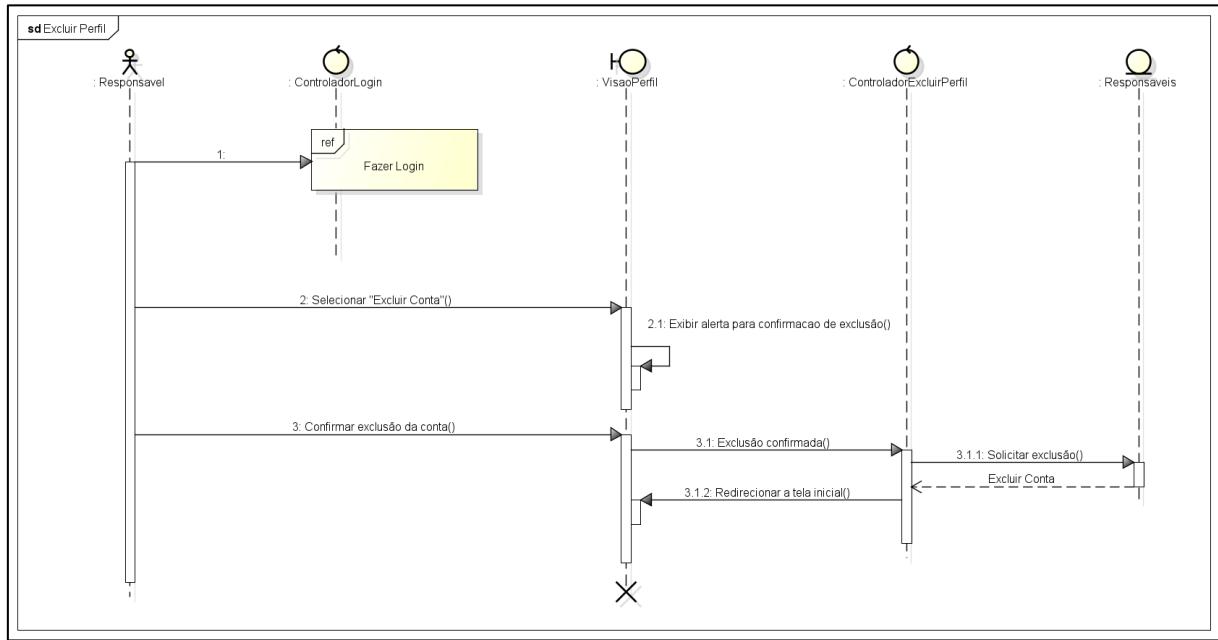
Fonte: De autoria própria, 2024.

Figura 34 - Diagrama de Sequência: Editar Perfil



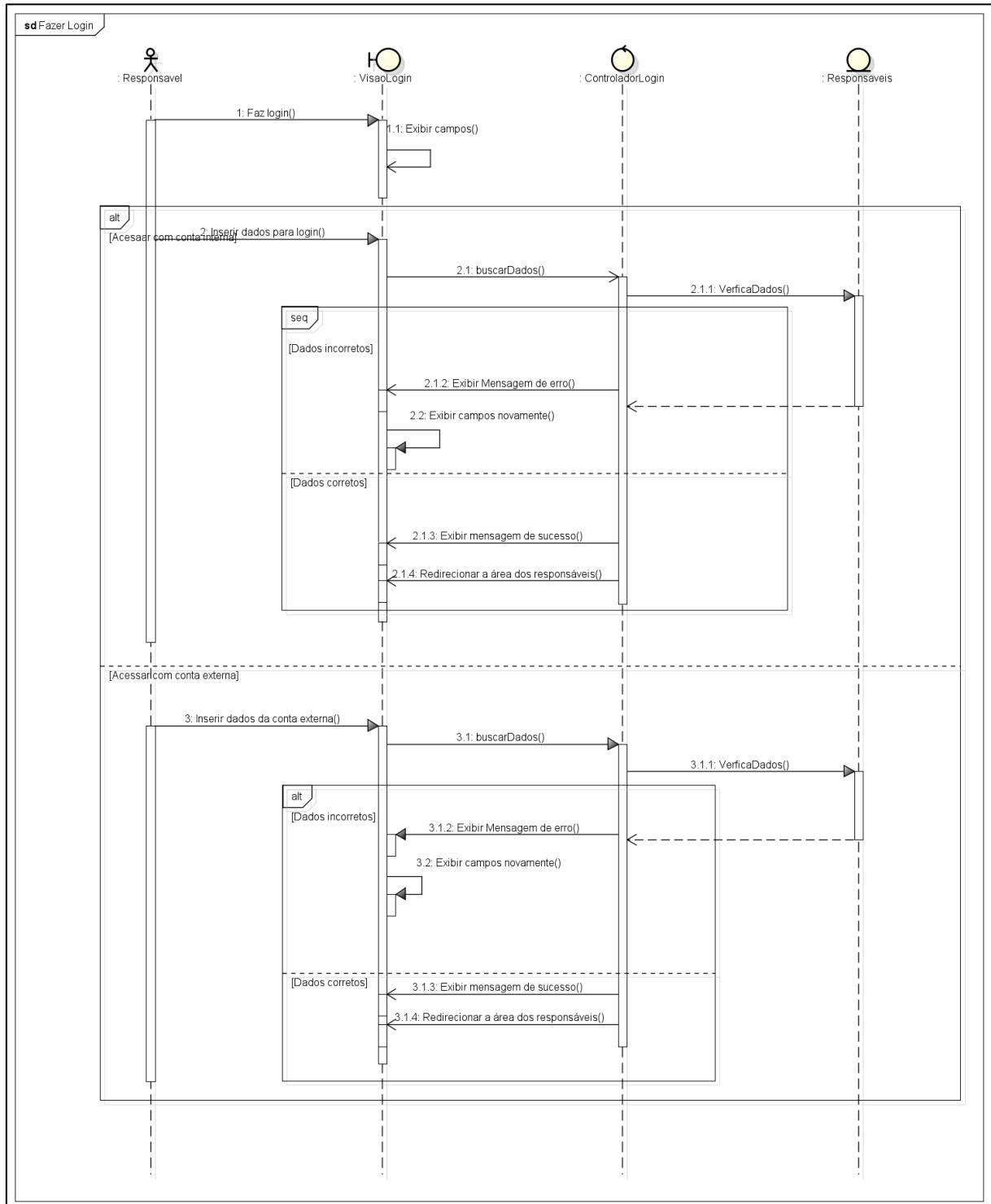
Fonte: De autoria própria, 2024.

Figura 35 - Diagrama de Sequência: Excluir Perfil



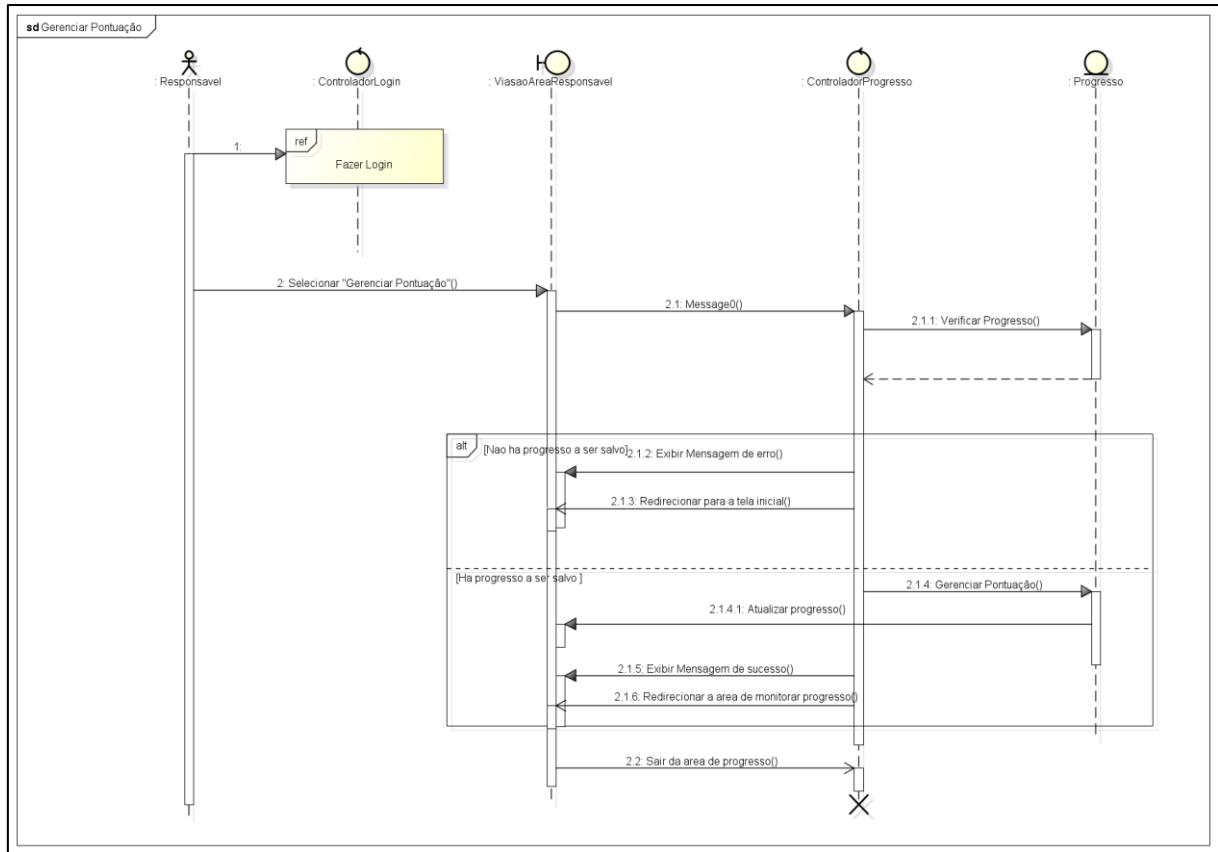
Fonte: De autoria própria, 2024.

Figura 36 - Diagrama de Sequência: Fazer Login



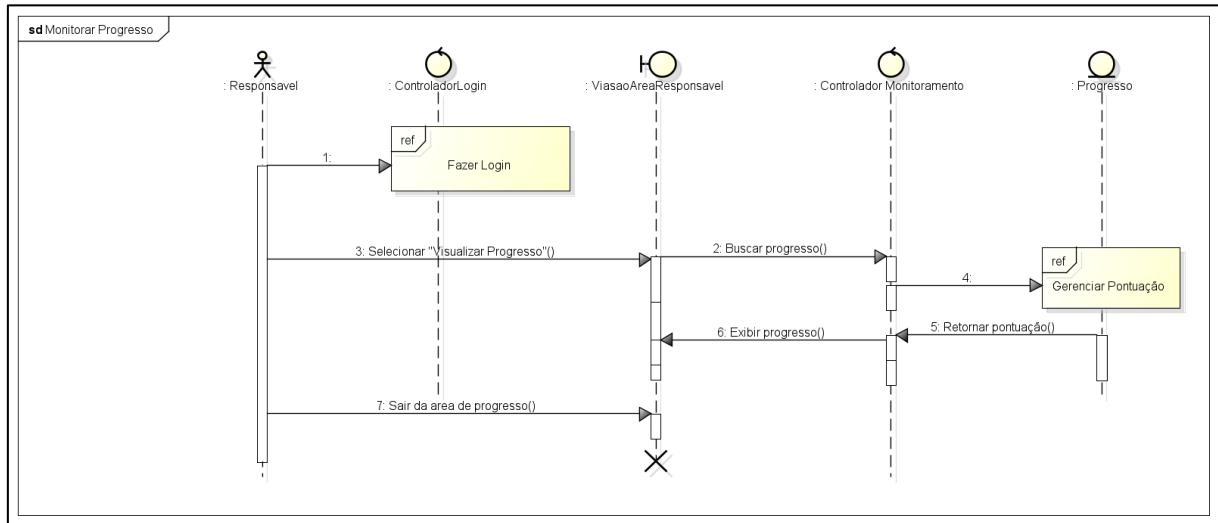
Fonte: De autoria própria, 2024.

Figura 37 - Diagrama de Sequência: Gerenciar Pontuação.



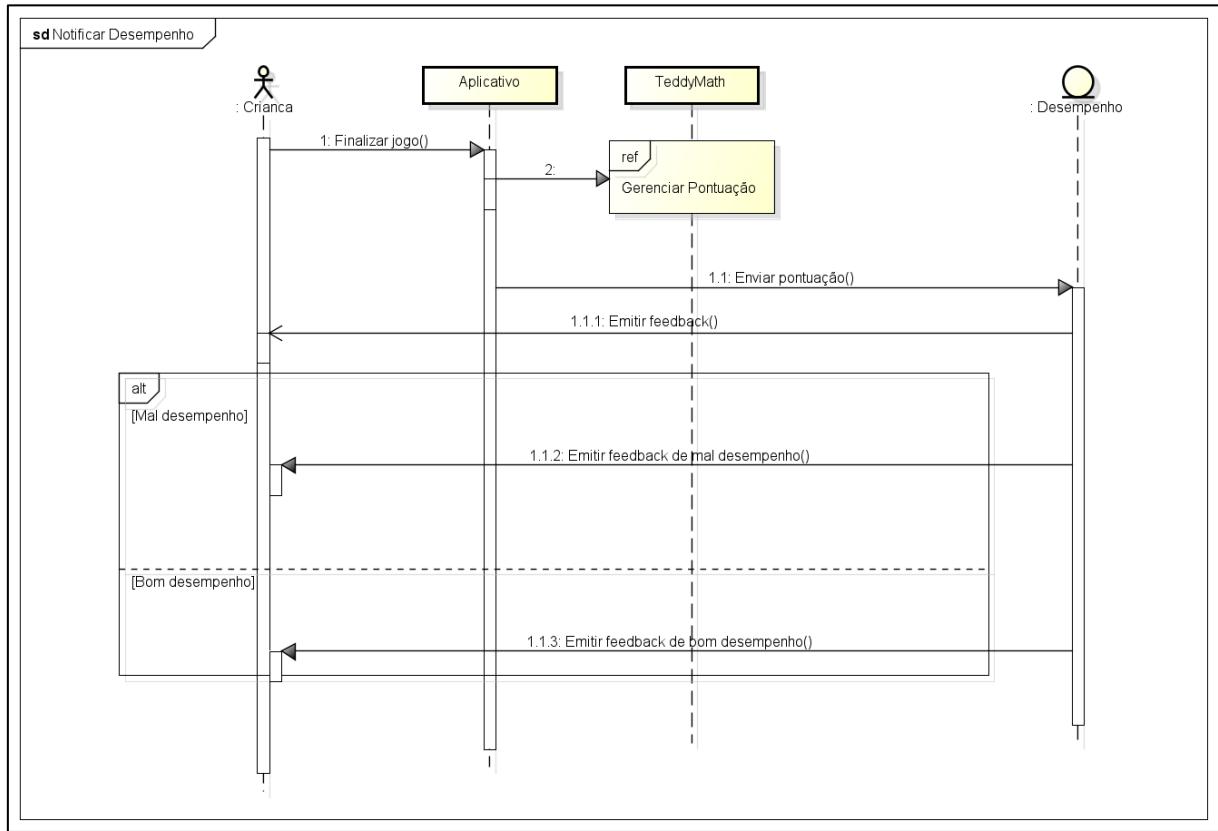
Fonte: De autoria própria, 2024.

Figura 38 - Diagrama de Sequência: Monitorar Progresso.



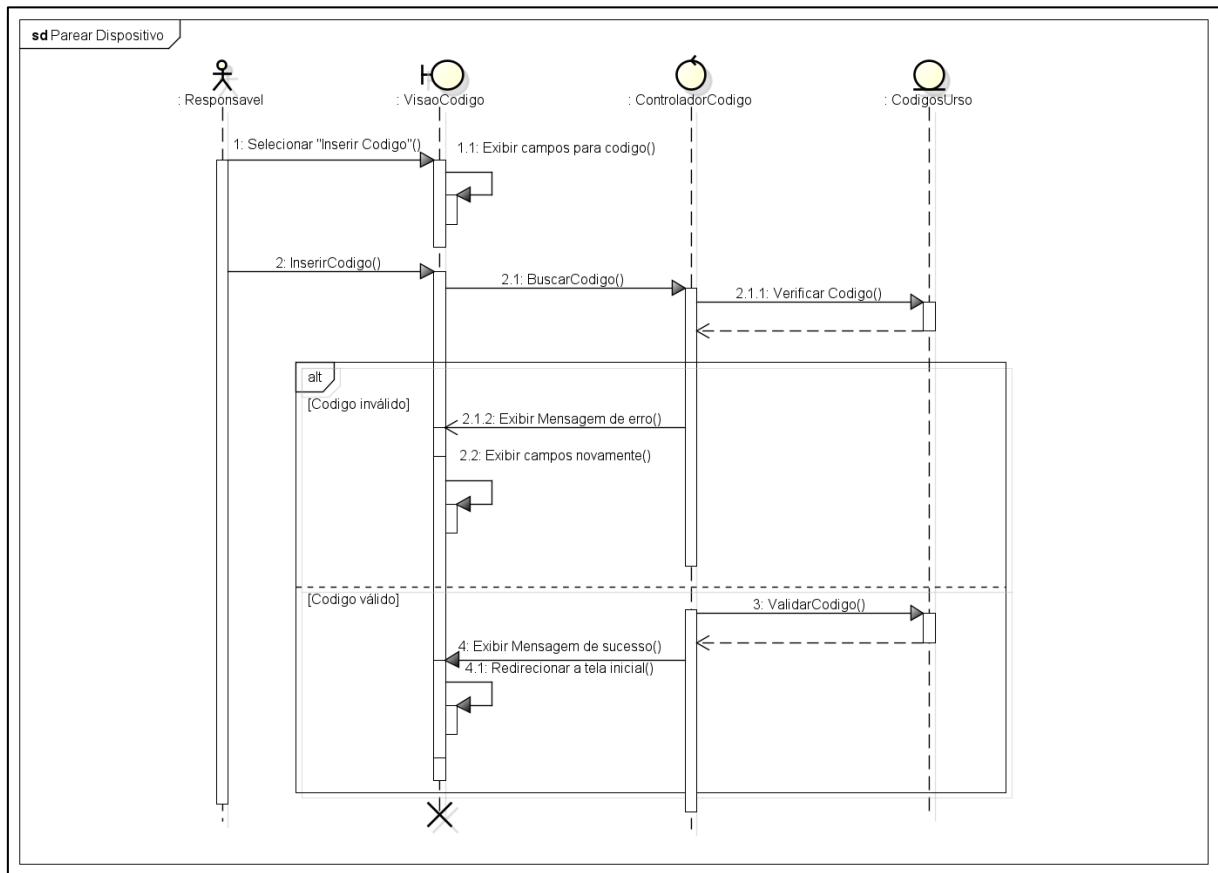
Fonte: De autoria própria, 2024.

Figura 39 - Diagrama de Sequência: Notificar Desempenho.



Fonte: De autoria própria, 2024.

Figura 40 - Diagrama de Sequência: Parear Dispositivo.

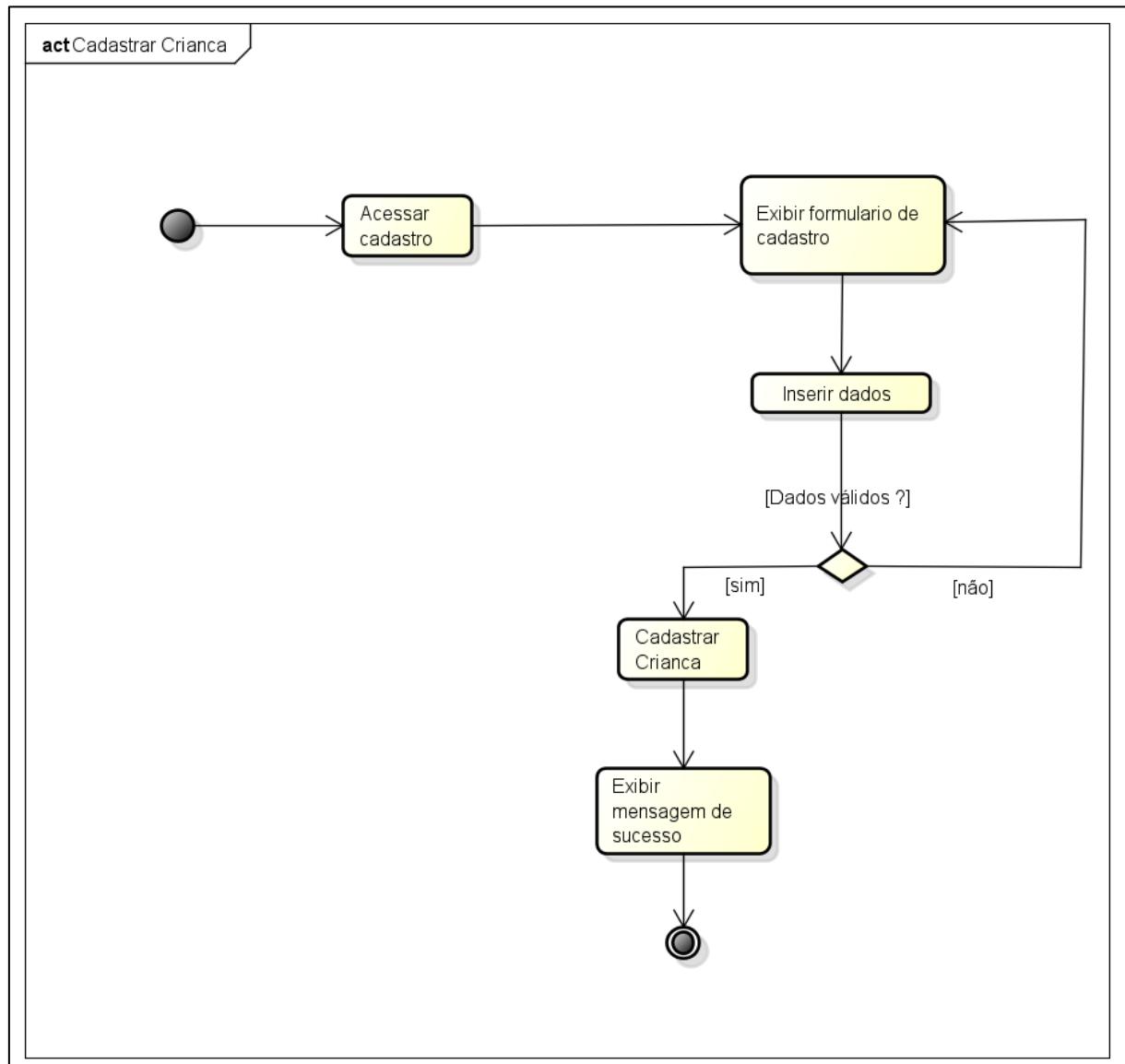


Fonte: De autoria própria, 2024.

3.3 Diagrama de Atividade

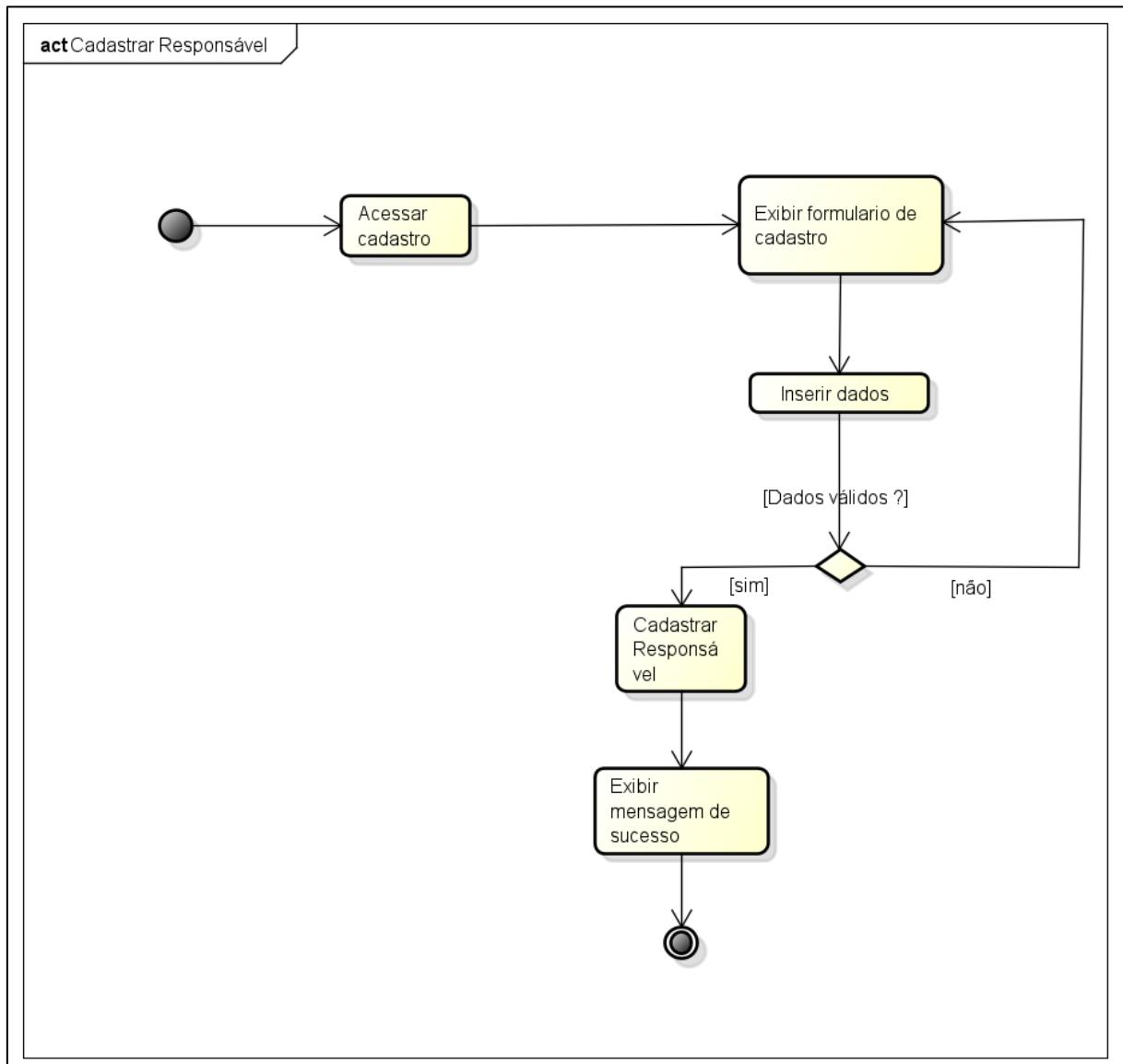
Segundo, os Diagramas de Atividade são necessários para descrever a lógica e fluxo operacional de cada ação e função, detalhando todas as etapas sequências de cada funcionalidade existente do sistema. Abaixo encontra-se os respectivos diagramas de atividade do projeto:

Figura 41 - Diagrama de Atividade: Cadastrar Criança.



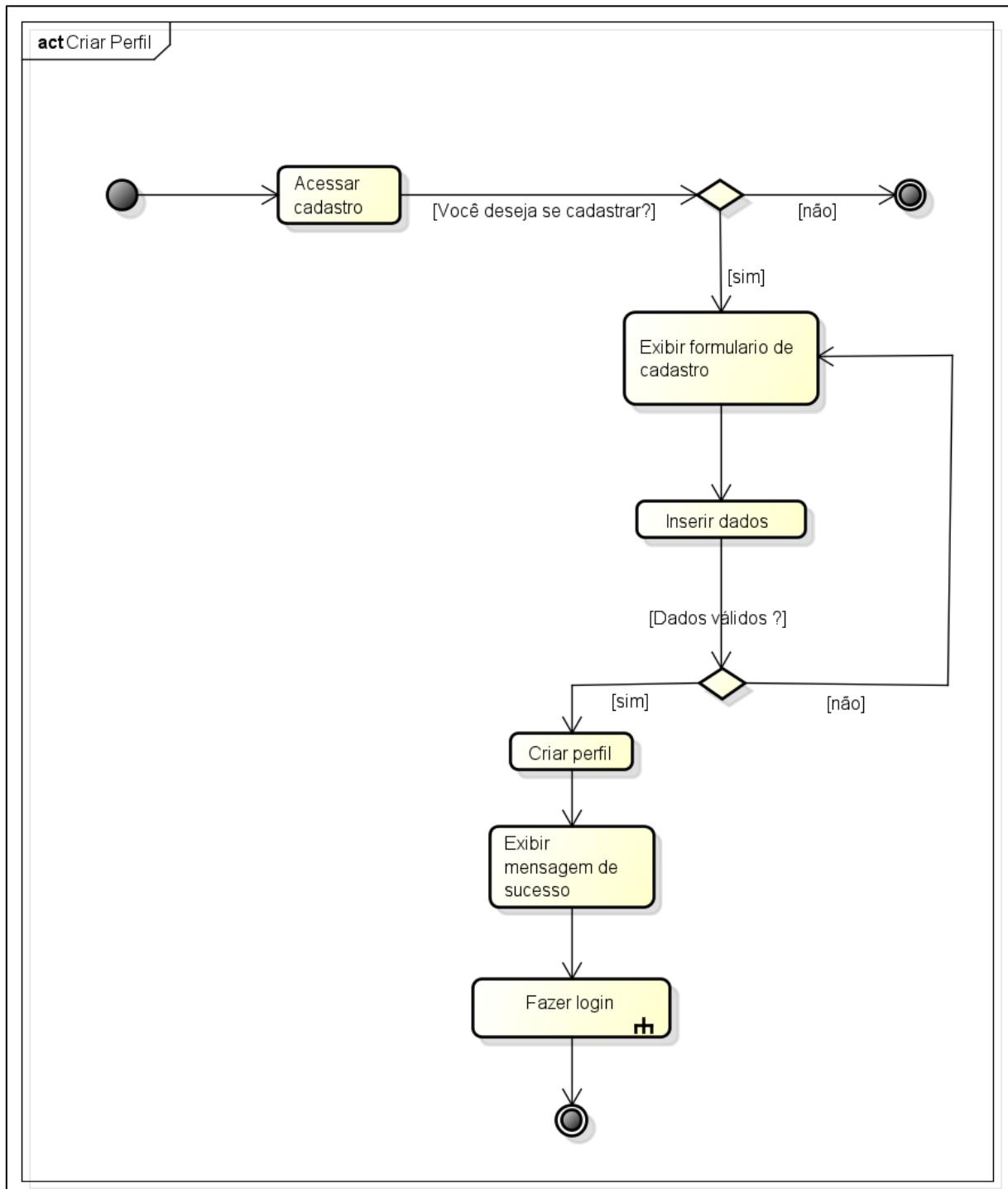
Fonte: De autoria própria, 2024.

Figura 42 - Diagrama de Atividade: Cadastrar Responsável.



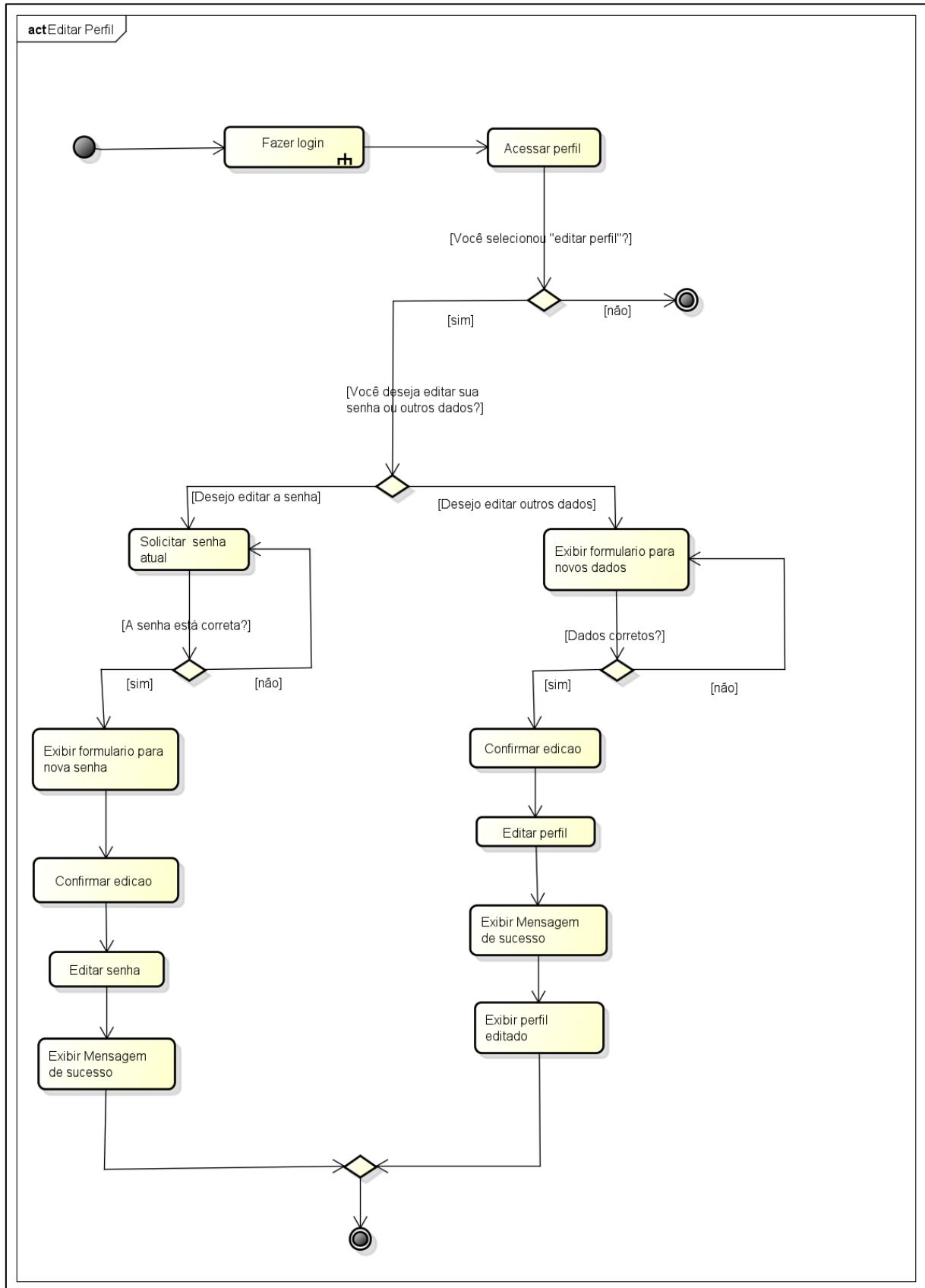
Fonte: De autoria própria, 2024.

Figura 43 - Diagrama de Atividade: Criar Perfil.



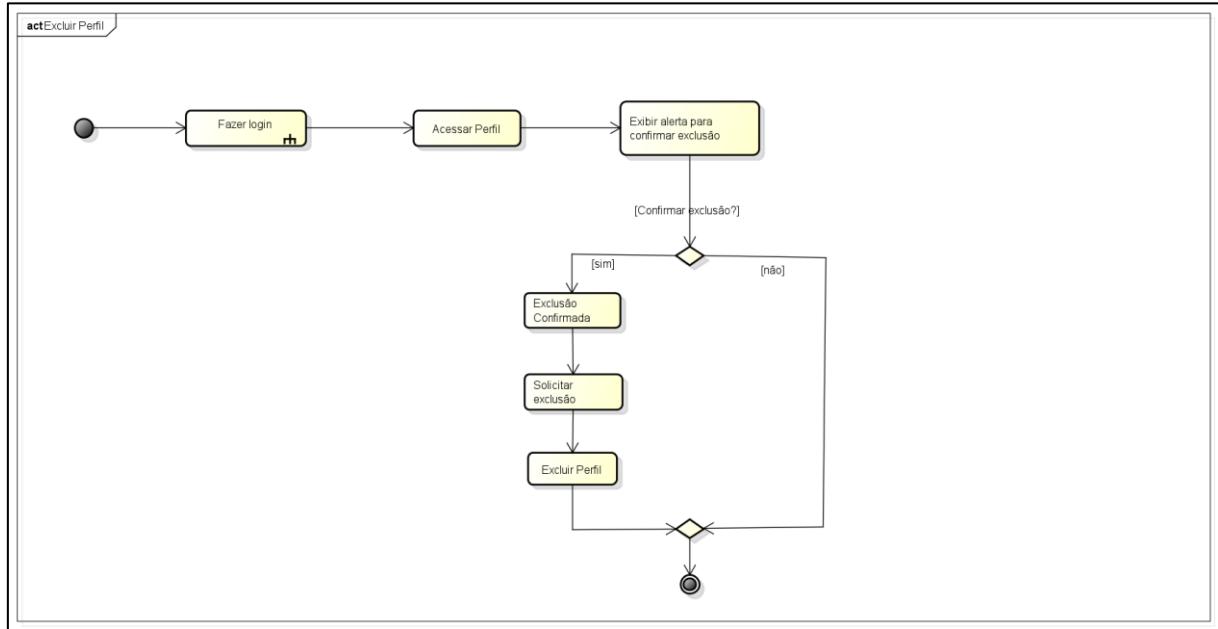
Fonte: De autoria própria, 2024.

Figura 44 - Diagrama de Atividade: Editar Perfil.



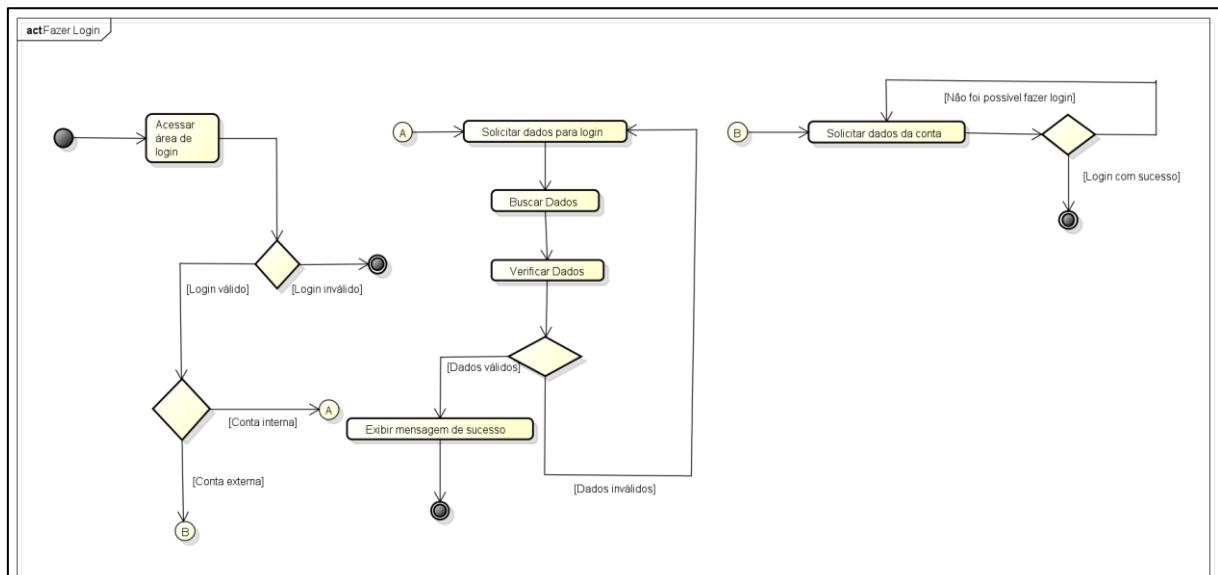
Fonte: De autoria própria, 2024.

Figura 45 - Diagrama de Atividade: Excluir Perfil.



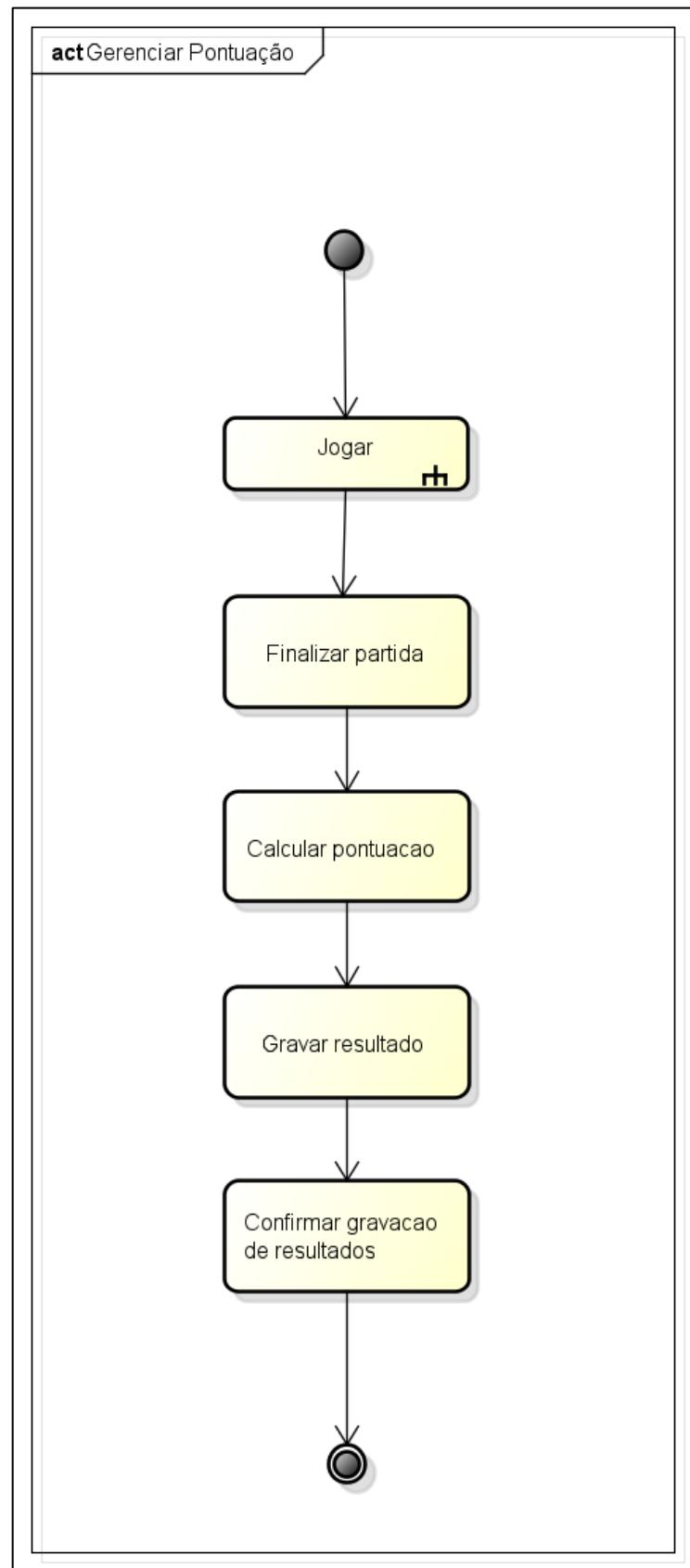
Fonte: De autoria própria, 2024.

Figura 46 - Diagrama de Atividade: Fazer Login.



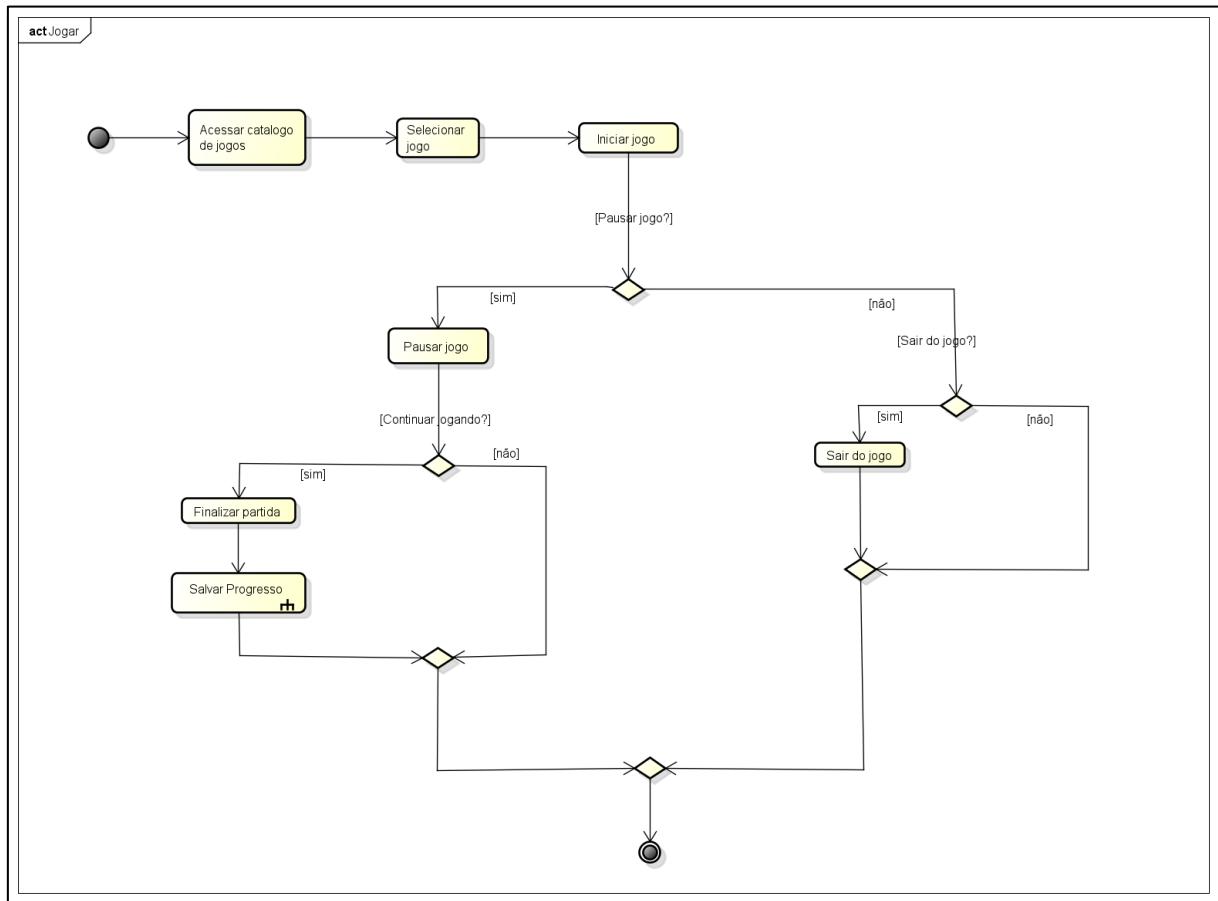
Fonte: De autoria própria, 2024.

Figura 47 - Diagrama de Atividade: Gerenciar Pontuação.



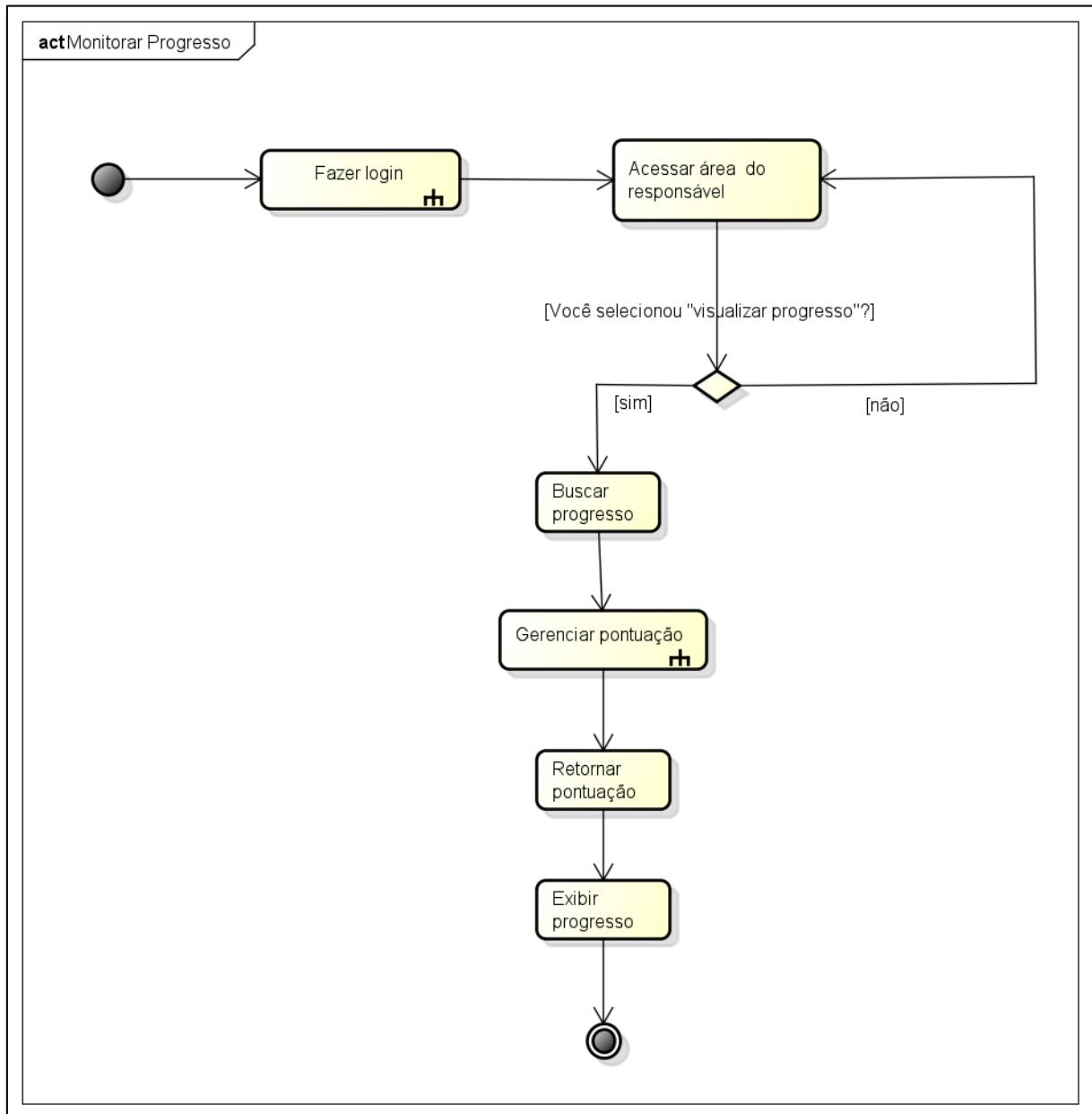
Fonte: De autoria própria, 2024.

Figura 48 - Diagrama de Atividade: Jogar.



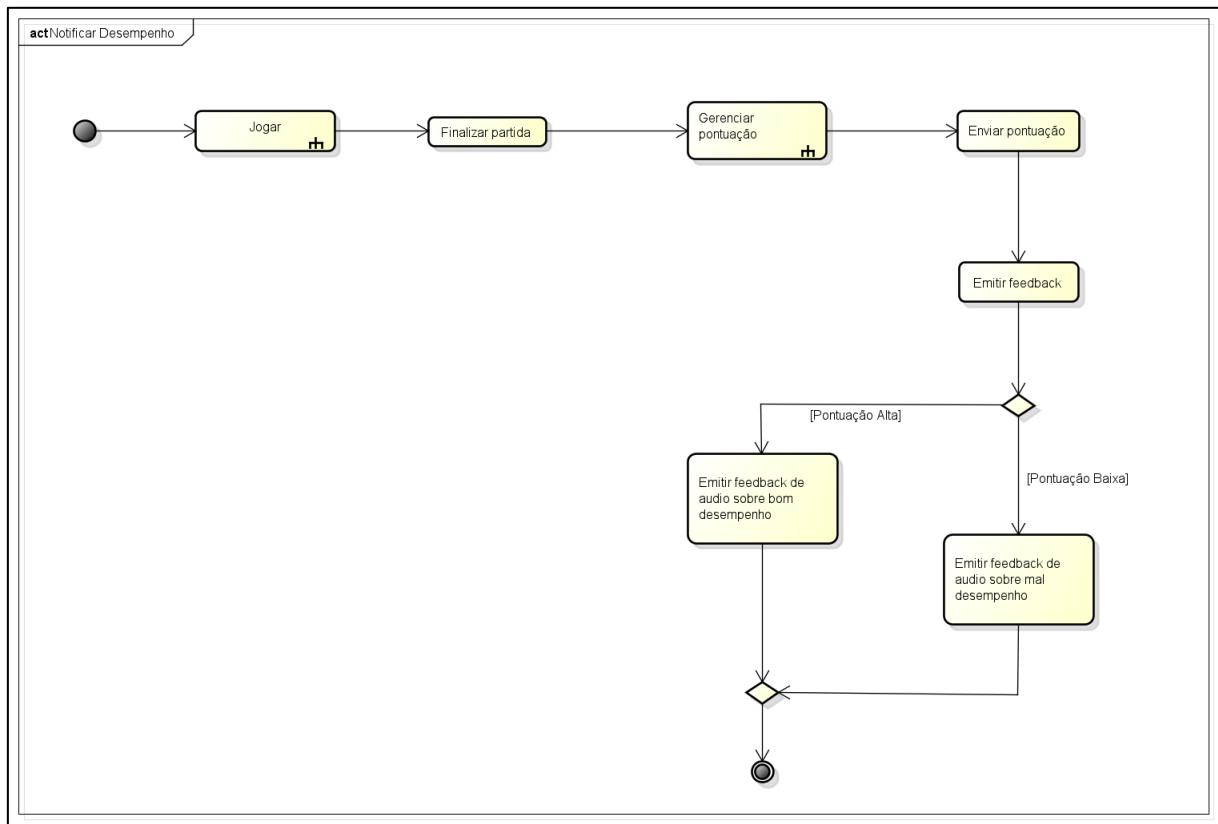
Fonte: De autoria própria, 2024.

Figura 49 - Diagrama de Atividade: Monitorar Progresso.



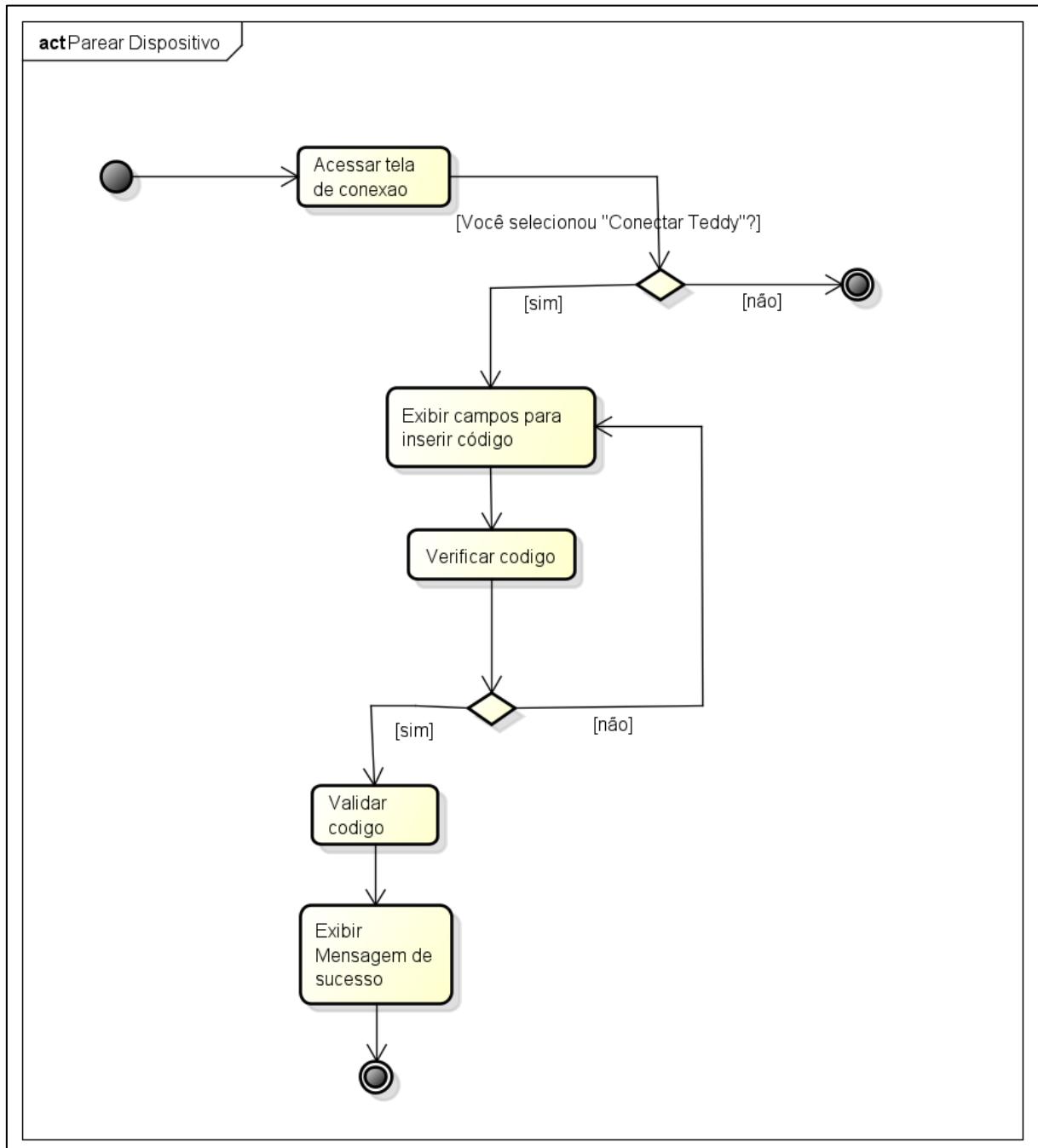
Fonte: De autoria própria, 2024.

Figura 50 - Diagrama de Atividade: Notificar Desempenho.



Fonte: De autoria própria, 2024.

Figura 51 - Diagrama de Atividade: Parear Dispositivo.

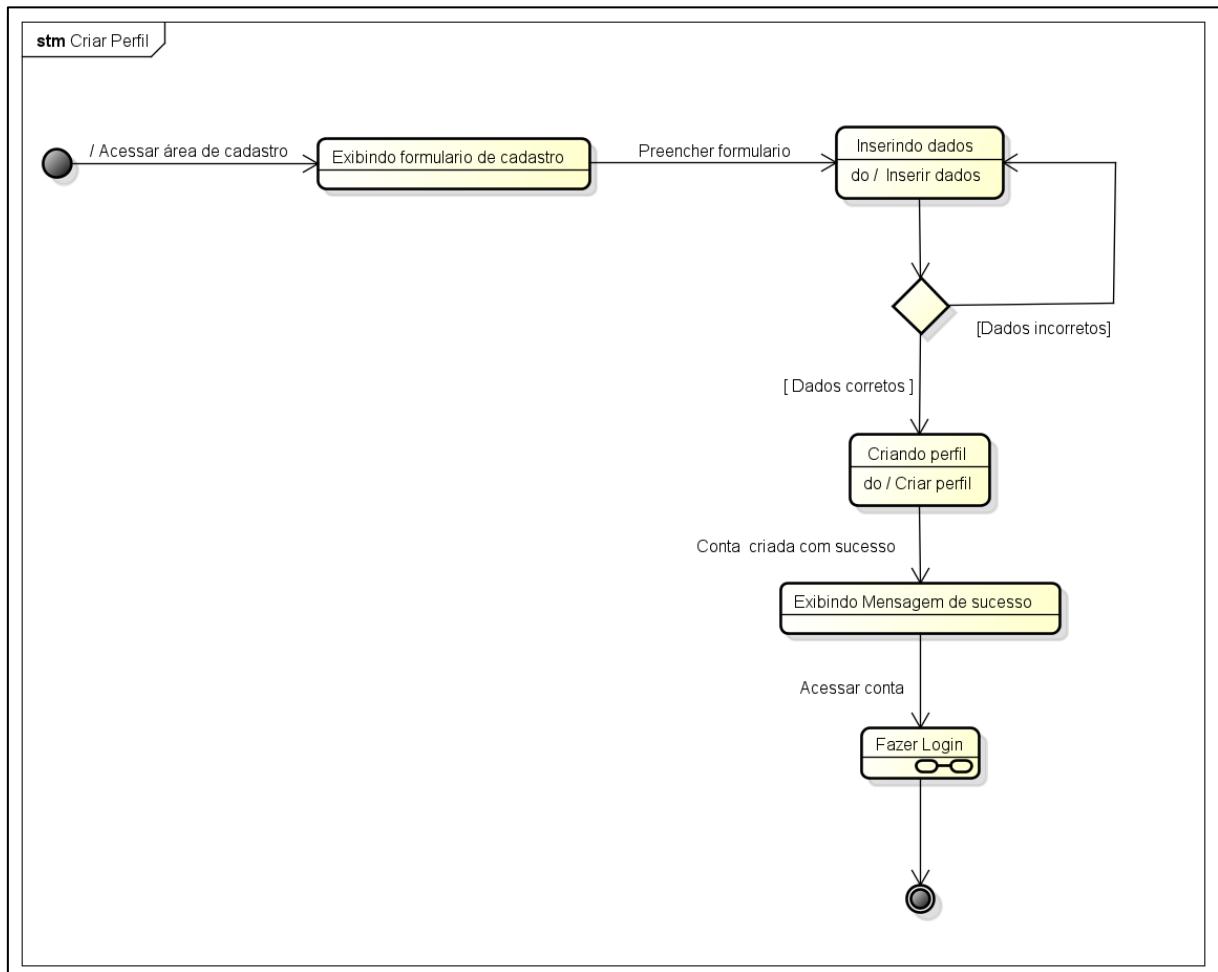


Fonte: De autoria própria, 2024.

3.4 Diagrama de Máquina de Estados

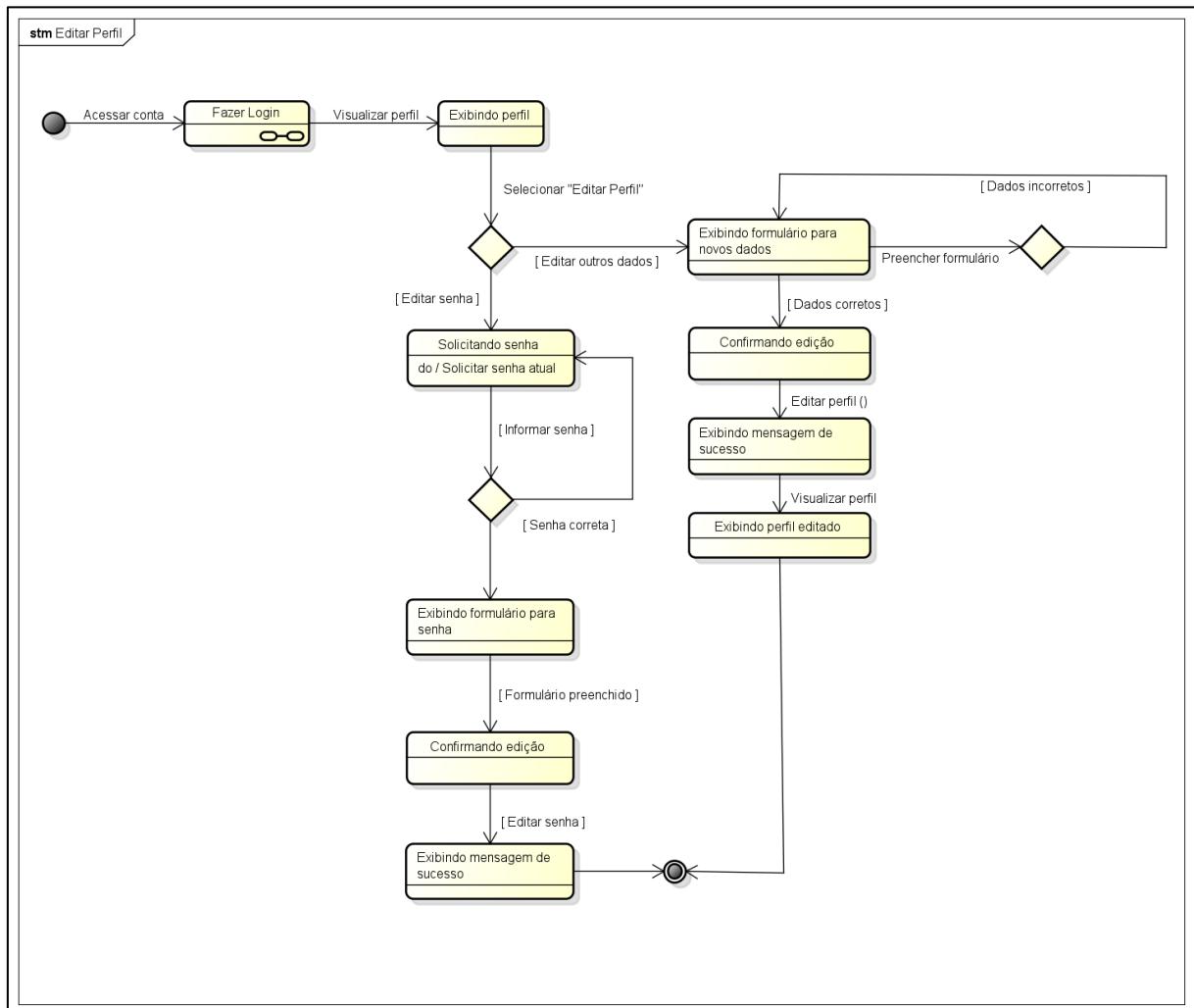
Com os Diagramas de Máquina de Estado é possível modelar o comportamento dinâmico do sistema e/ou objeto ao interagir com eventos, sejam eles externos ou internos. Abaixo estão os respectivos diagramas de máquina de estado do projeto:

Figura 52 - Diagrama de Máquina de Estados: Criar Perfil.



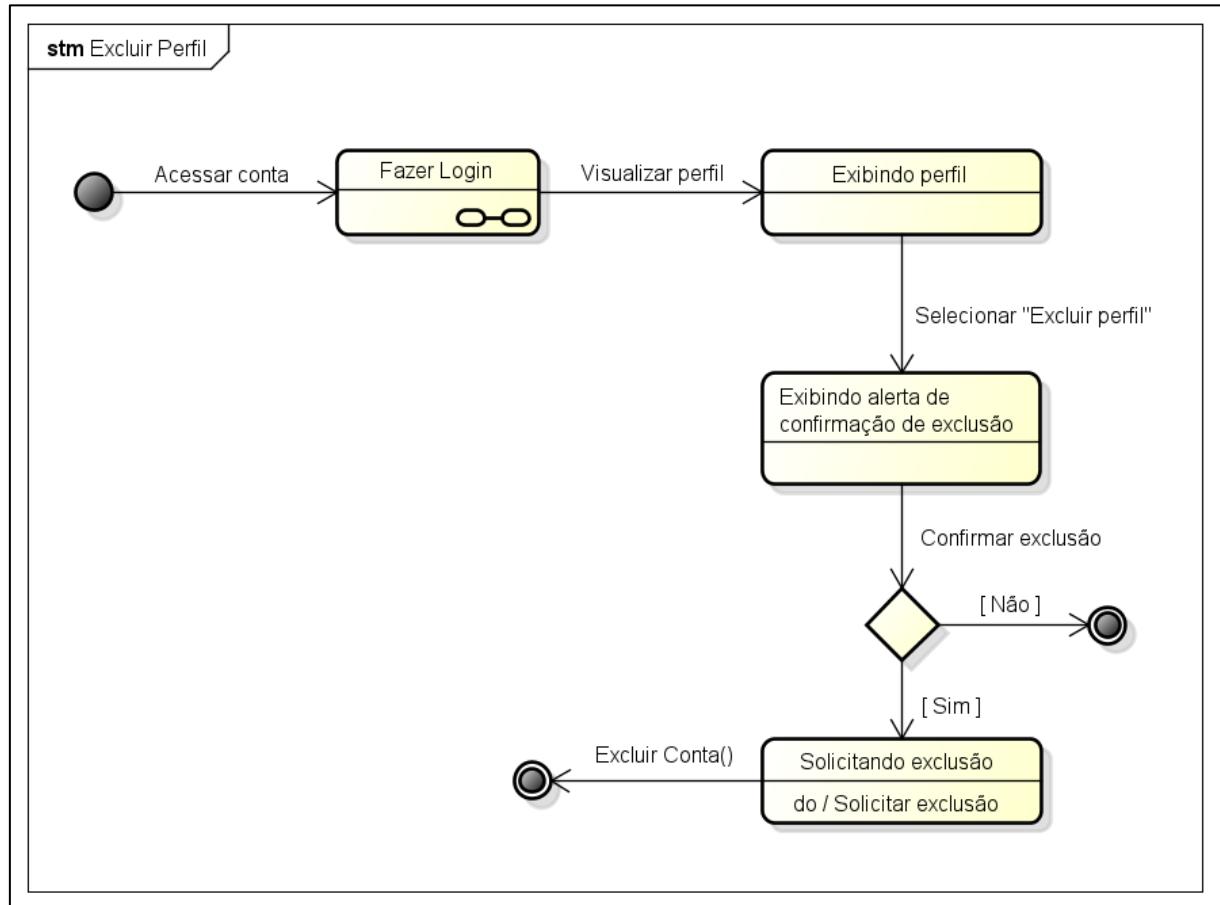
Fonte: De autoria própria, 2024.

Figura 53 - Diagrama de Máquina de Estados: Editar Perfil.



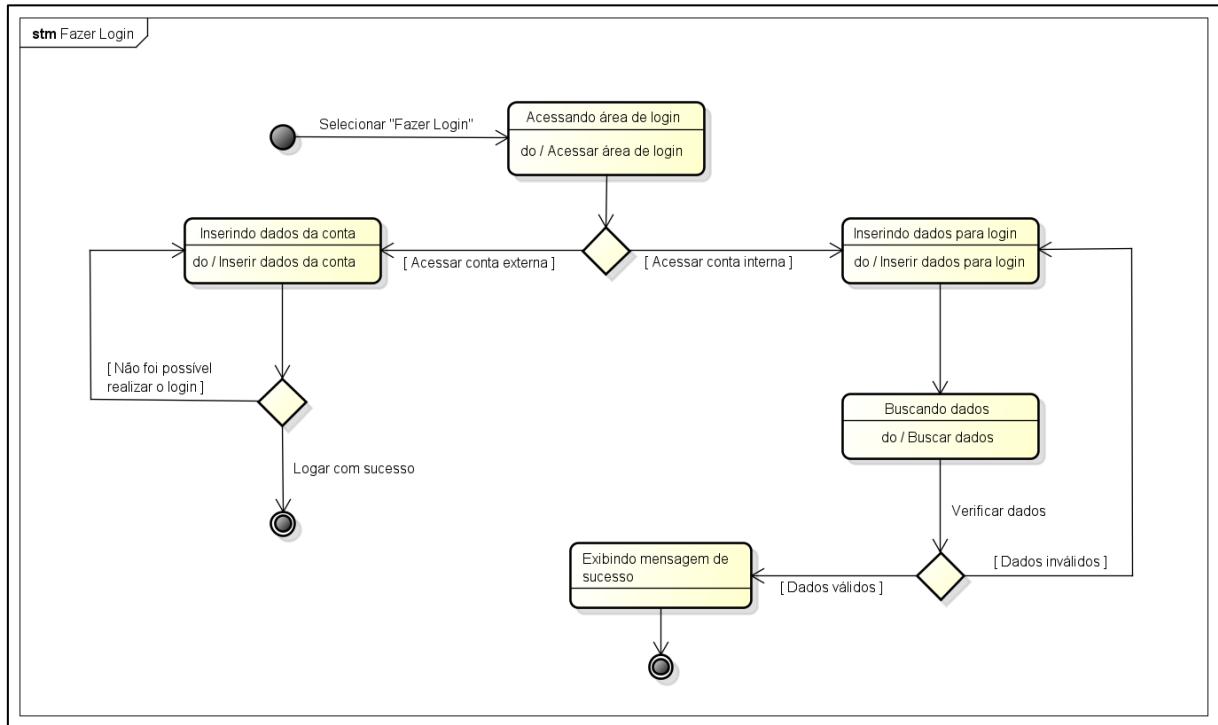
Fonte: De autoria própria, 2024.

Figura 54 - Diagrama de Máquina de Estados: Excluir Perfil.



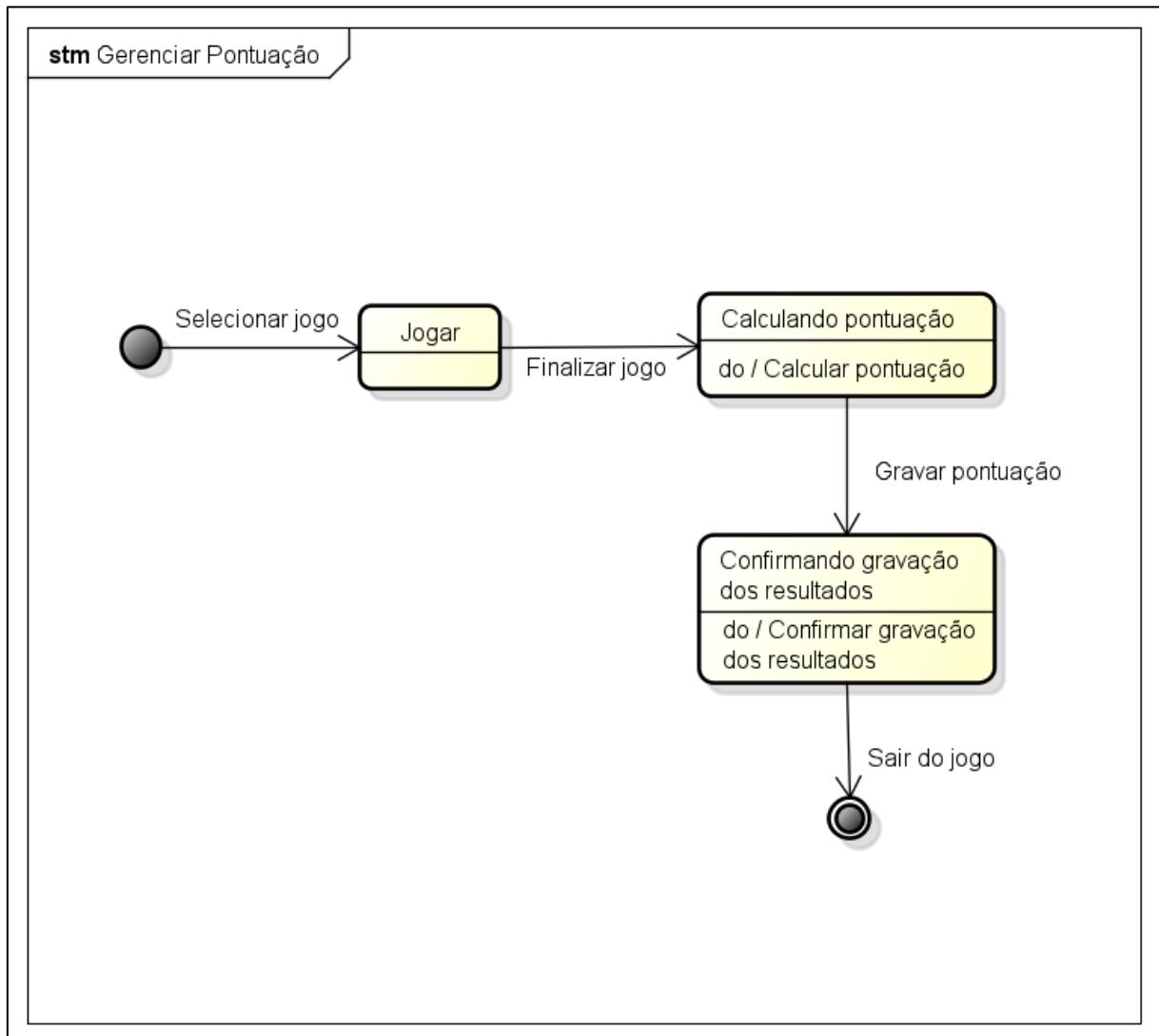
Fonte: De autoria própria, 2024.

Figura 55 - Diagrama de Máquina de Estados: Fazer Login.



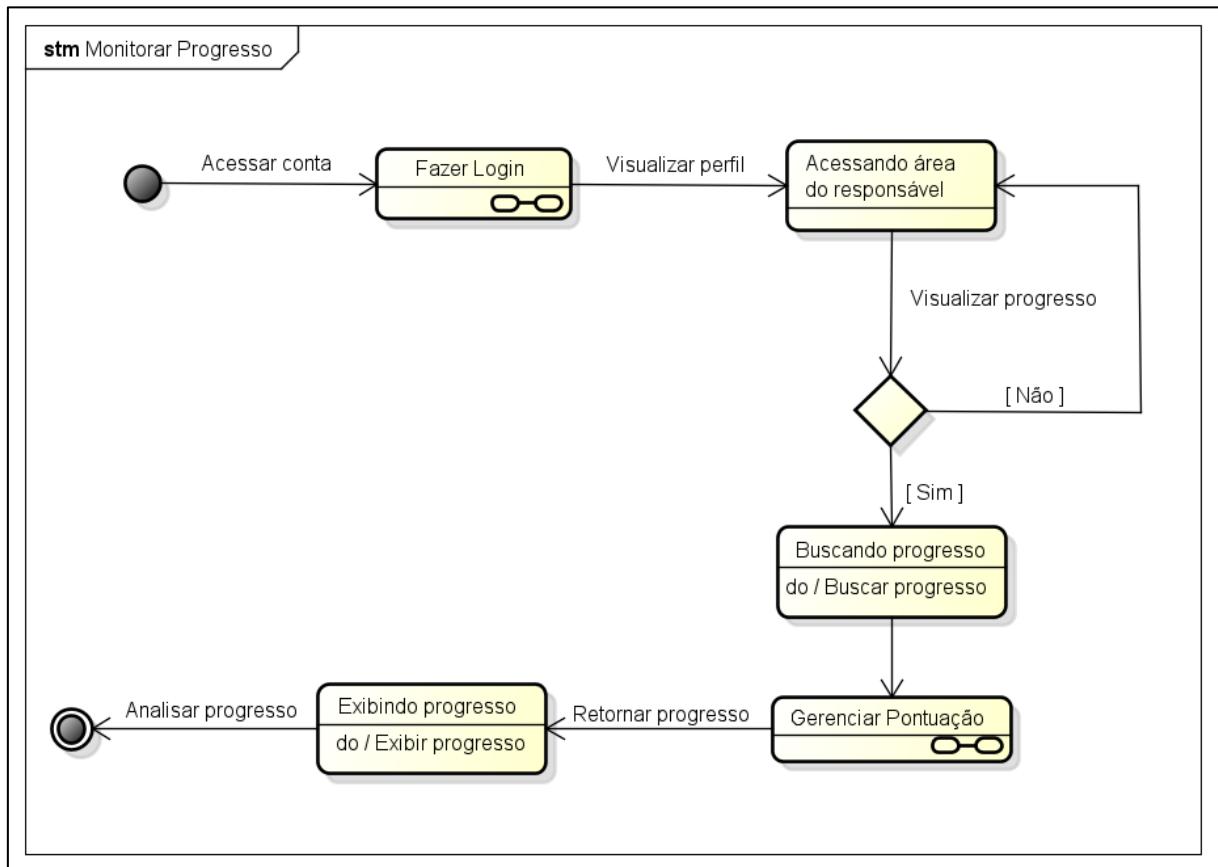
Fonte: De autoria própria, 2024.

Figura 56 - Diagrama de Máquina de Estados: Gerenciar Pontuação.



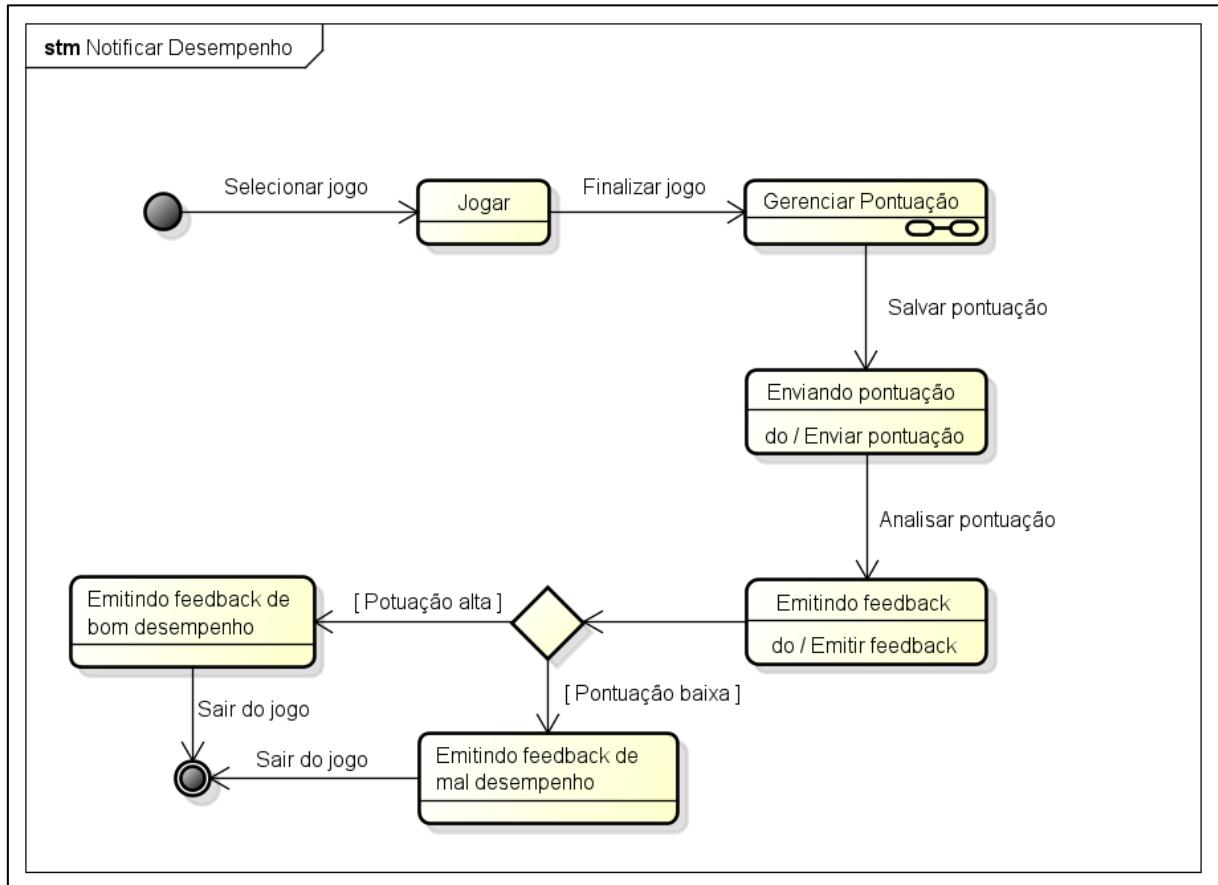
Fonte: De autoria própria, 2024.

Figura 57 - Diagrama de Máquina de Estados: Monitorar Progresso.



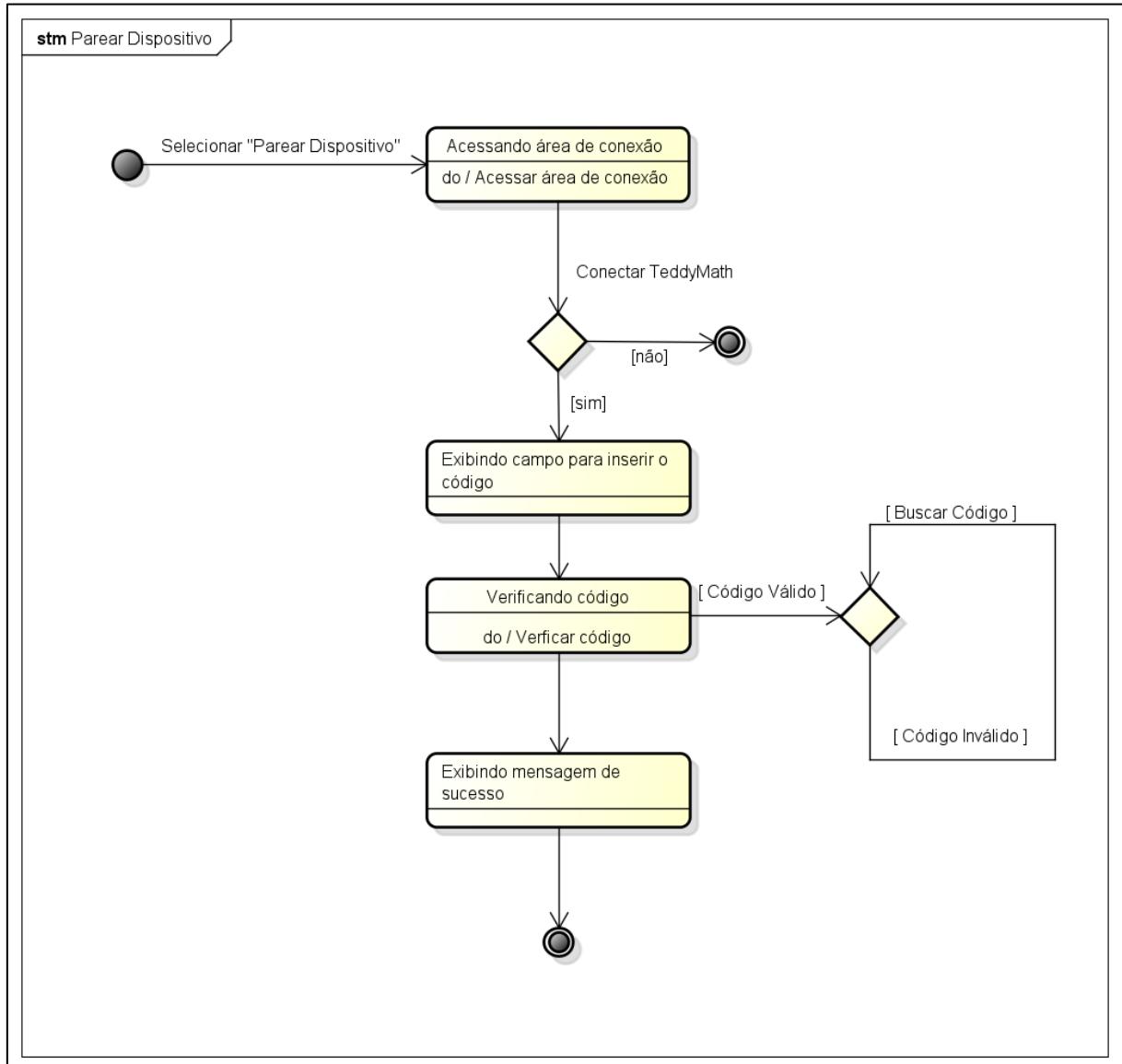
Fonte: De autoria própria, 2024.

Figura 58 - Diagrama de Máquina de Estados: Notificar Desempenho.



Fonte: De autoria própria, 2024.

Figura 59 - Diagrama de Máquina de Estados: Parear Dispositivo

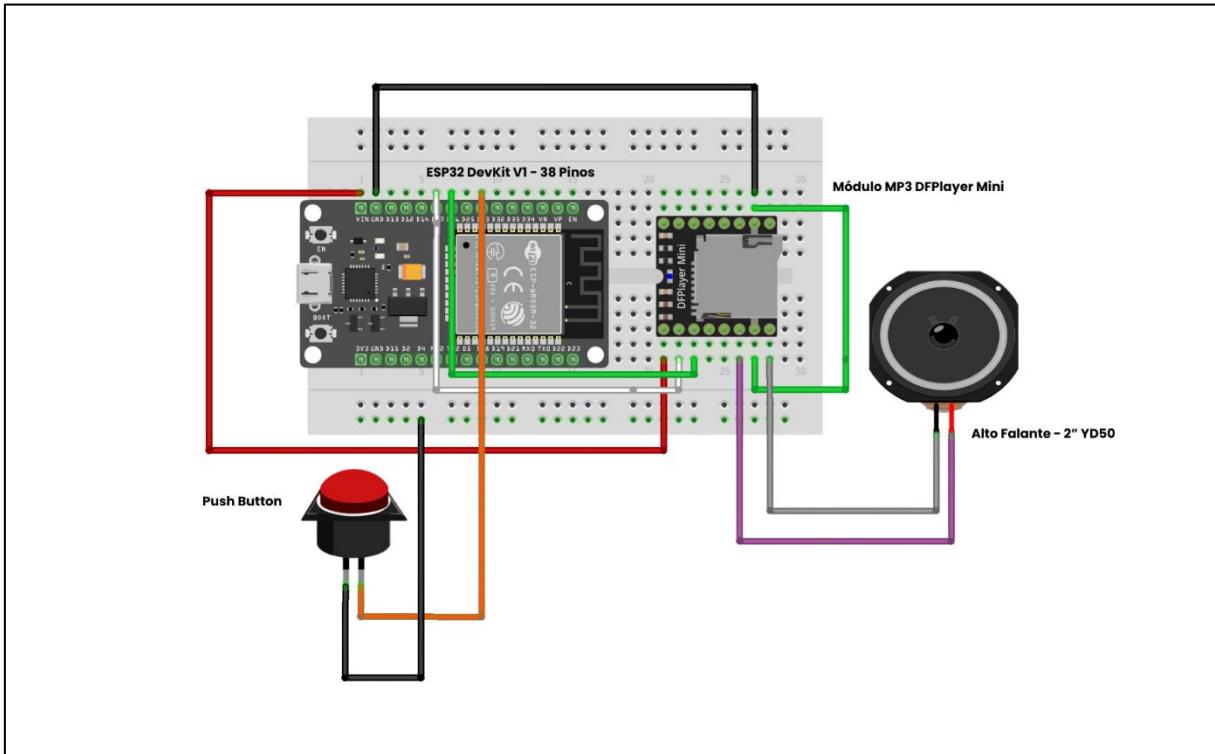


Fonte: De autoria própria, 2024.

3.5 Montagem do circuito do dispositivo

Nesta seção será ilustrado por meio de uma figura como os componentes eletrônicos foram conectados e estruturados para fazer parte do dispositivo físico do sistema TeddyMath, por qual a criança poderá interagir de forma lúdica.

Figura 60 - Diagrama de conexão entre os componentes.



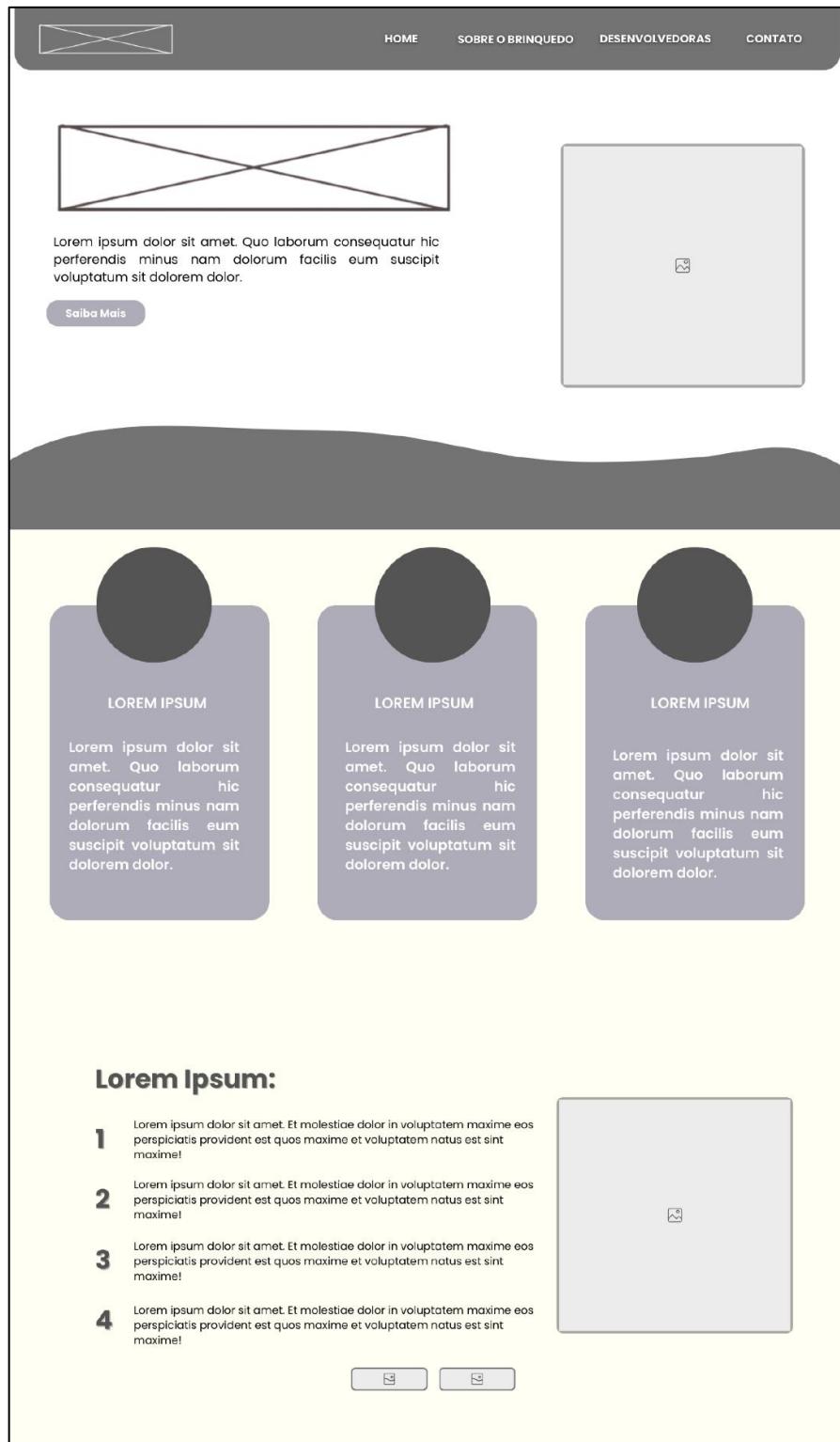
Fonte: De autoria própria, 2024.

3.6 Prototipagem das telas da aplicação

Esta seção apresenta as interfaces gráficas que correspondem as telas do *website* e do aplicativo, criadas com base nos *wireframes* de baixa e alta fidelidade. O *website* é apenas informativo para fins da divulgação do projeto, sem funcionalidade. Por outro lado, o aplicativo apresenta uma variedade de telas interativas, incluindo jogos, acesso ao perfil dos pais, além de possuir com uma navegação fluída e divertida. No total, são 35 telas que direcionam os pais e as crianças numa jornada de aprendizagem. O primeiro *wireframe* apresentado são as páginas Inicial, Vantagens e

Como Funciona do site informativo do projeto TeddyMath, sendo este de baixa fidelidade:

Figura 61 - Wireframe Baixa Fidelidade do Site: Página Inicial, Vantagens e Como Funciona



Fonte: De autoria própria, 2024.

Consecutivo, é mostrado a mesma página em alta fidelidade:

Figura 62 - Wireframe Alta Fidelidade do Site: Página Inicial, Vantagens e Como Funciona

TEDDY MATH

HOME SOBRE O BRINQUEDO DESENVOLVEDORAS CONTATO

SEU URSO AMIGO PARA APRENDER BRINCANDO

Com o TeddyMath, as crianças descobrem o mundo da matemática de forma divertida e interativa. O nosso ursinho, sempre pronto para ajudar, transforma o aprendizado em uma aventura cheia de brincadeiras e descobertas!

[Saiba Mais](#)

TECNOLOGIA E DIVERSÃO JUNTAS
O TeddyMath conecta-se ao app via Bluetooth, criando uma experiência interativa que torna o aprendizado de matemática mais envolvente e divertido.

APRENDIZADO PERSONALIZADO
O ursinho responde aos progressos da criança, adaptando atividades e desafios para garantir que cada etapa do aprendizado seja compreendida de maneira divertida e eficaz.

COMPANHEIRO DE ESTUDOS
O TeddyMath, junto com o app, oferece suporte em tempo real, guiando a criança nas atividades e tornando o aprendizado um momento especial e inesquecível.

Como Funciona: Baixando TeddyMath

- 1 Abra a App Store ou Google Play no seu dispositivo e procure por "TeddyMath". O aplicativo está disponível gratuitamente para download!
- 2 Clique em "Instalar" e aguarde o aplicativo ser baixado e instalado no seu dispositivo.
- 3 Após abrir o app, siga as instruções para conectar seu ursinho TeddyMath via Bluetooth. É simples e rápido!
- 4 Agora é só criar o perfil da criança e começar a explorar atividades divertidas e educativas com o Teddy!

[GET IT ON Google Play](#) [Download on the App Store](#)

Olá, Alana

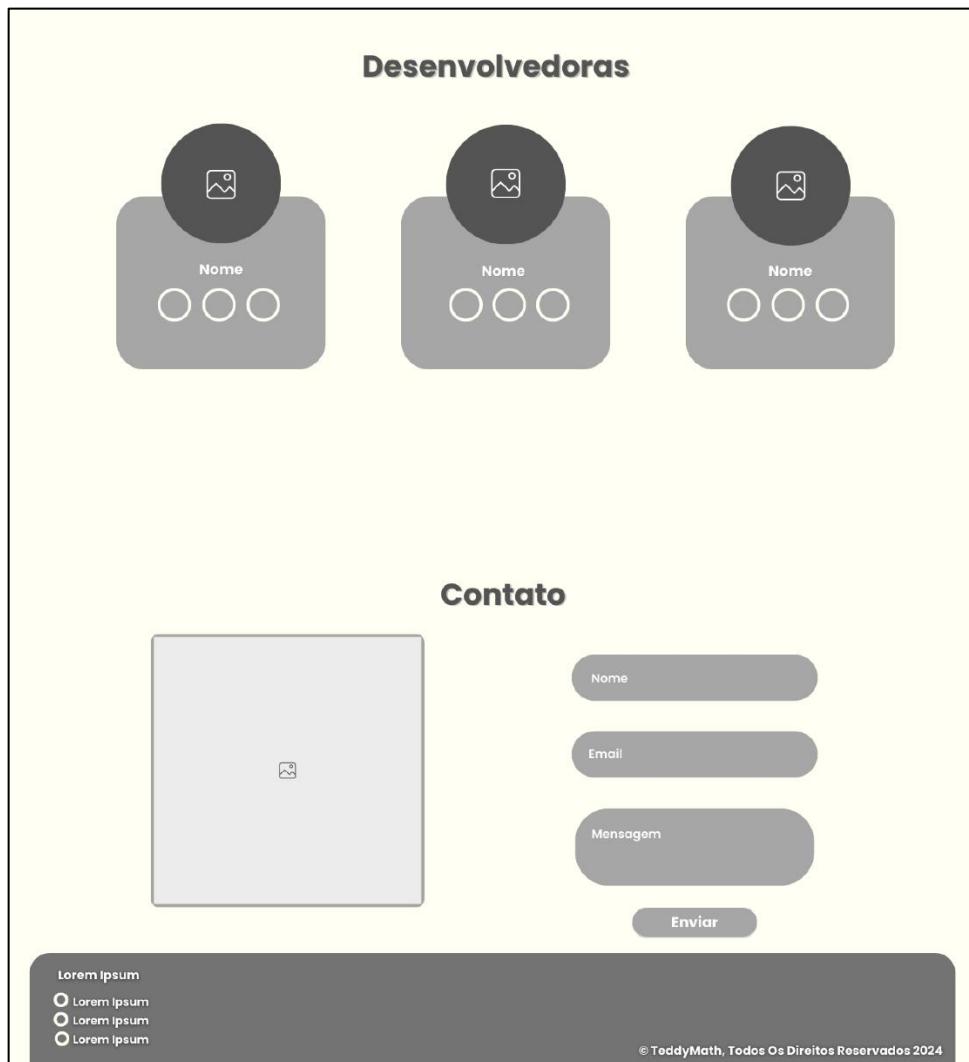
Colorê Caça Formas

TeddyQuiz Tabuada

Fonte: De autoria própria, 2024.

Na sequência, por sua vez é apresentado a continuação do site informativo, sendo estas: Desenvolvedoras e Contato em baixa fidelidade.

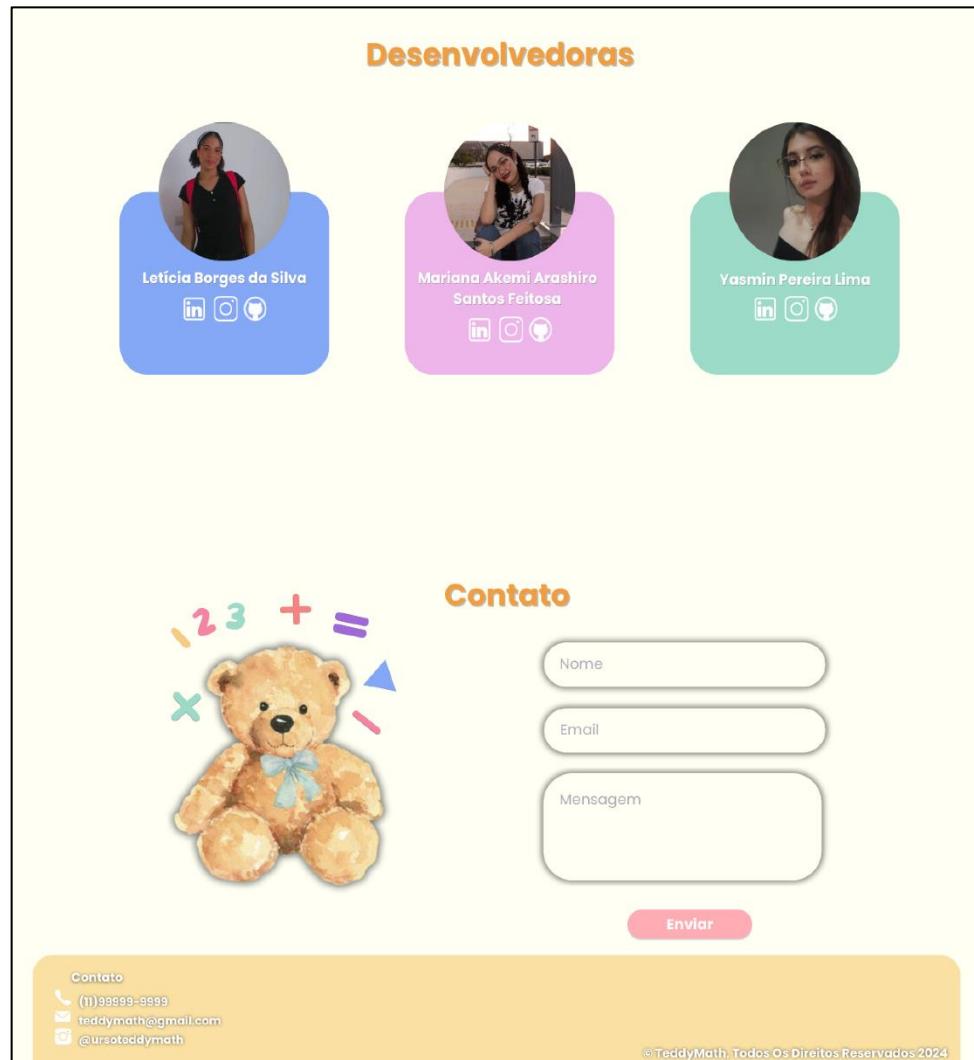
Figura 63 - Wireframe Baixa Fidelidade do Site: Desenvolvedoras e Contato



Fonte: De autoria própria, 2024.

Subsequente, é mostrada a mesma página em alta fidelidade:

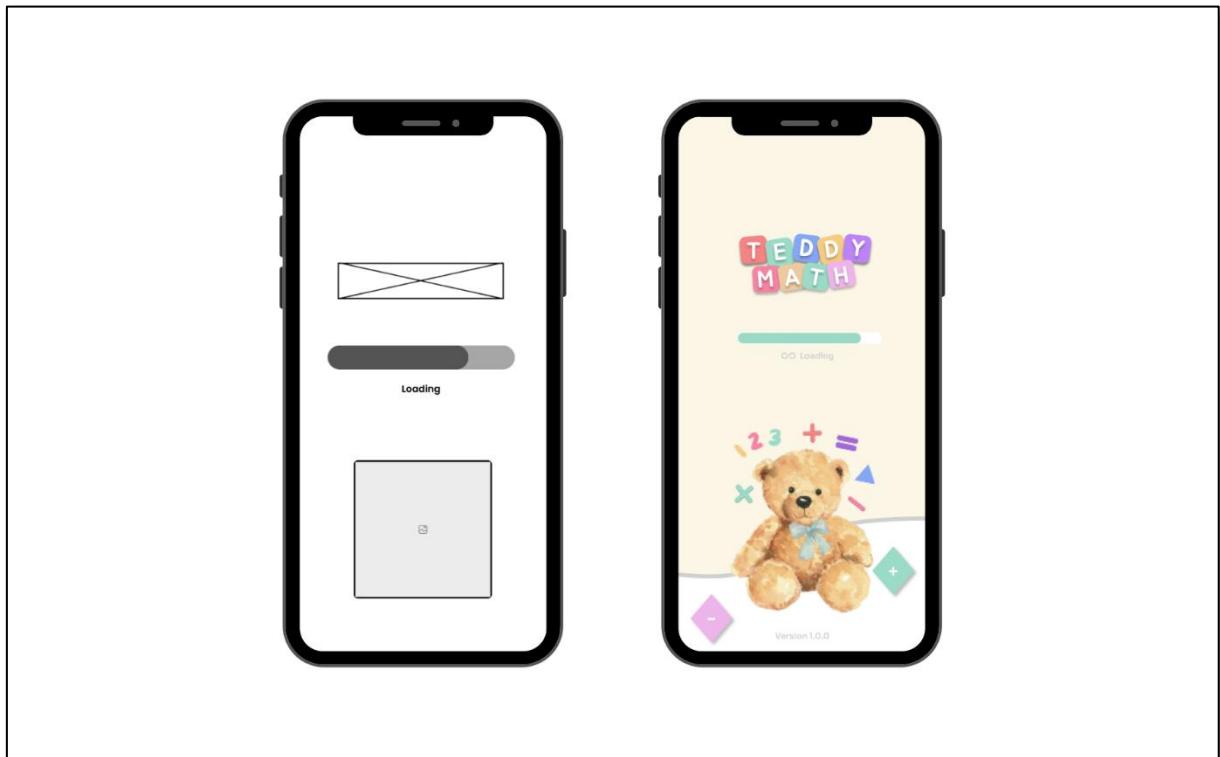
Figura 64 - Wireframe Alta Fidelidade do Site: Desenvolvedoras e Contato



Fonte: De autoria própria, 2024.

Agora, passando para as interfaces gráficas do aplicativo, a figura mostrada abaixo apresenta a tela inicial de carregamento do aplicativo em baixa e alta fidelidade:

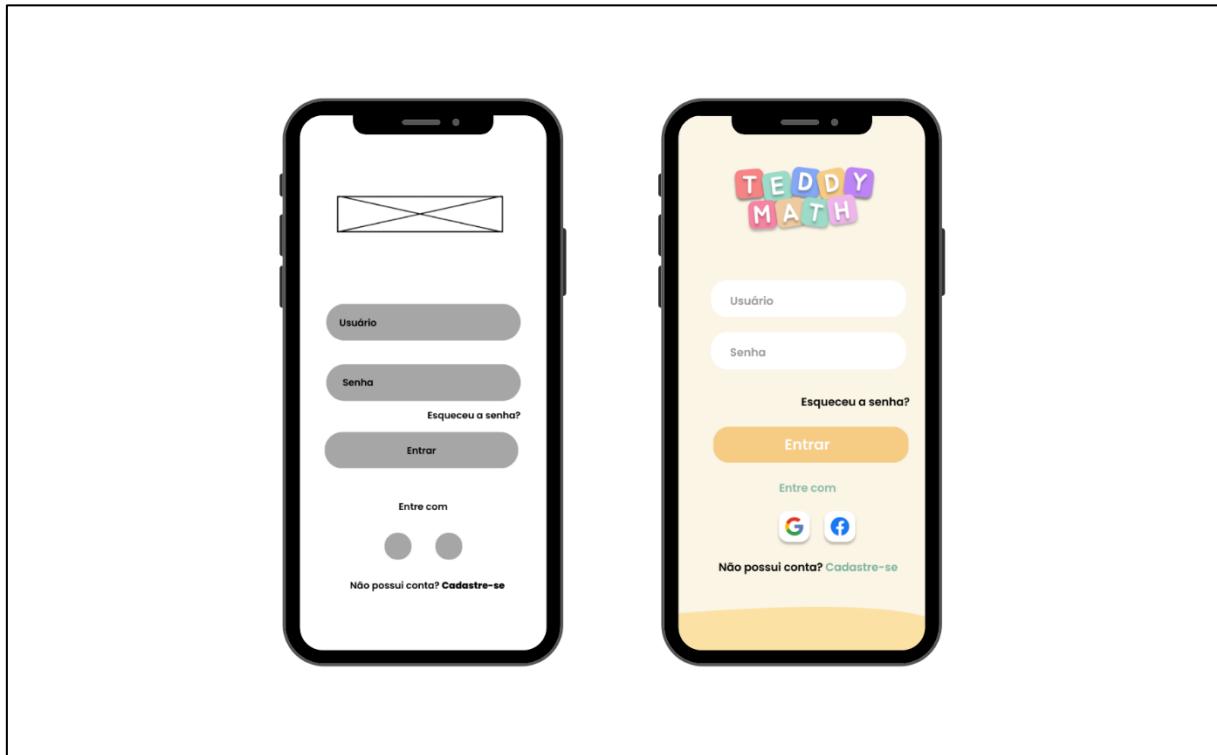
Figura 65 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Inicial.



Fonte: De autoria própria, 2024.

Na figura abaixo é ilustrado a tela de login, caso o responsável queira realizar o login:

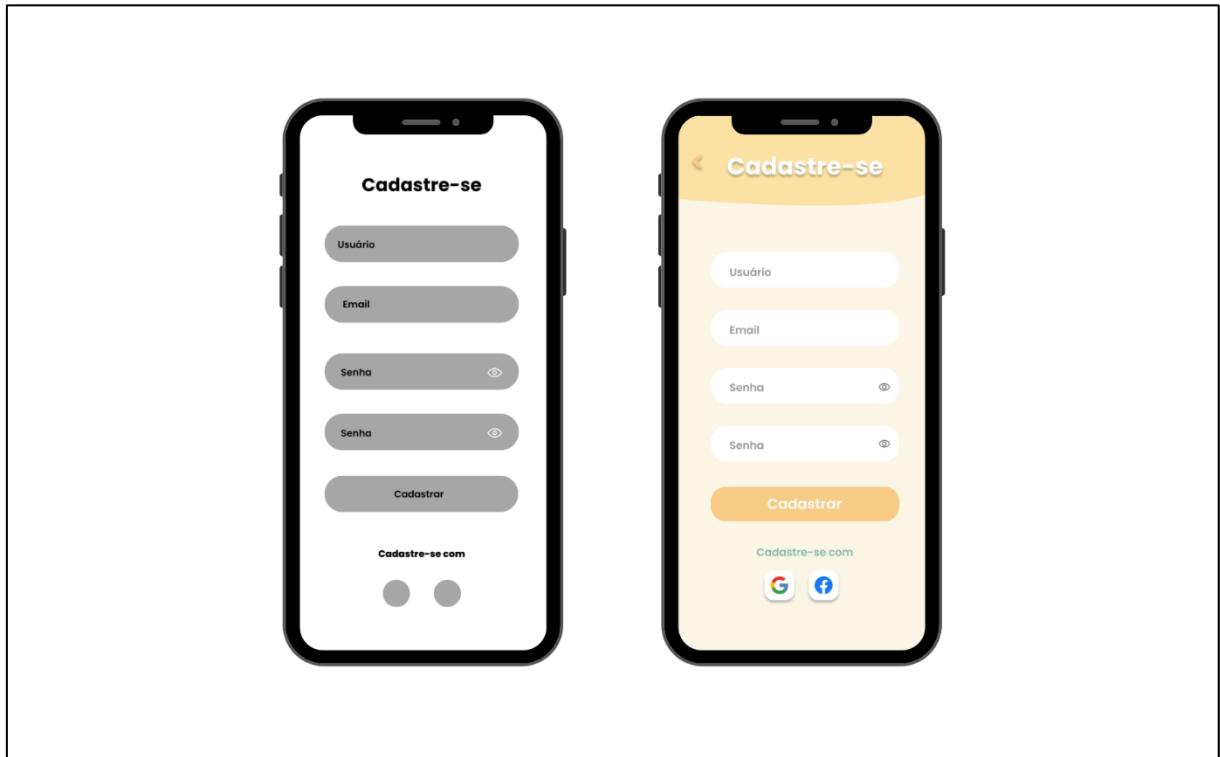
Figura 66 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Login.



Fonte: De autoria própria, 2024.

No caso de o responsável não ter criado uma conta, este pode clicar no botão de “cadastre-se” como ilustrado na figura abaixo:

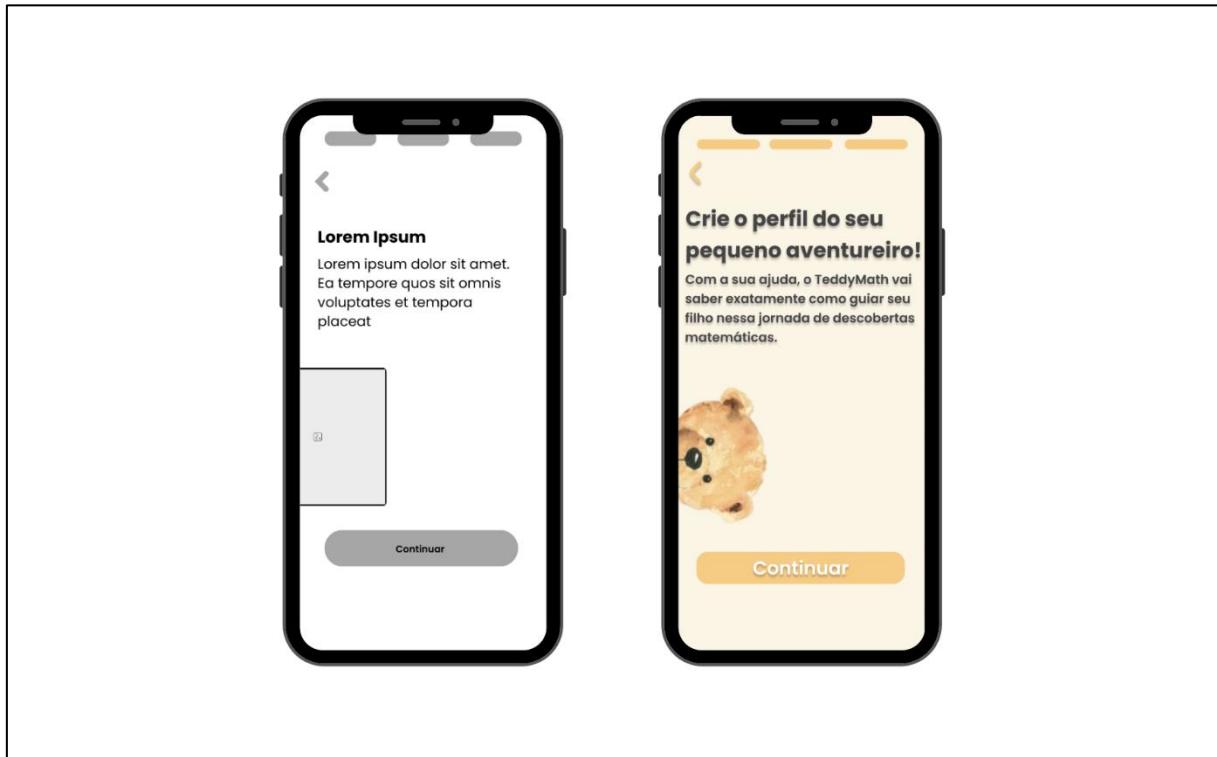
Figura 67 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Cadastro.



Fonte: De autoria própria, 2024.

Após o responsável criar sua conta, este é redirecionado para criar o perfil da criança, como mostrado na figura abaixo:

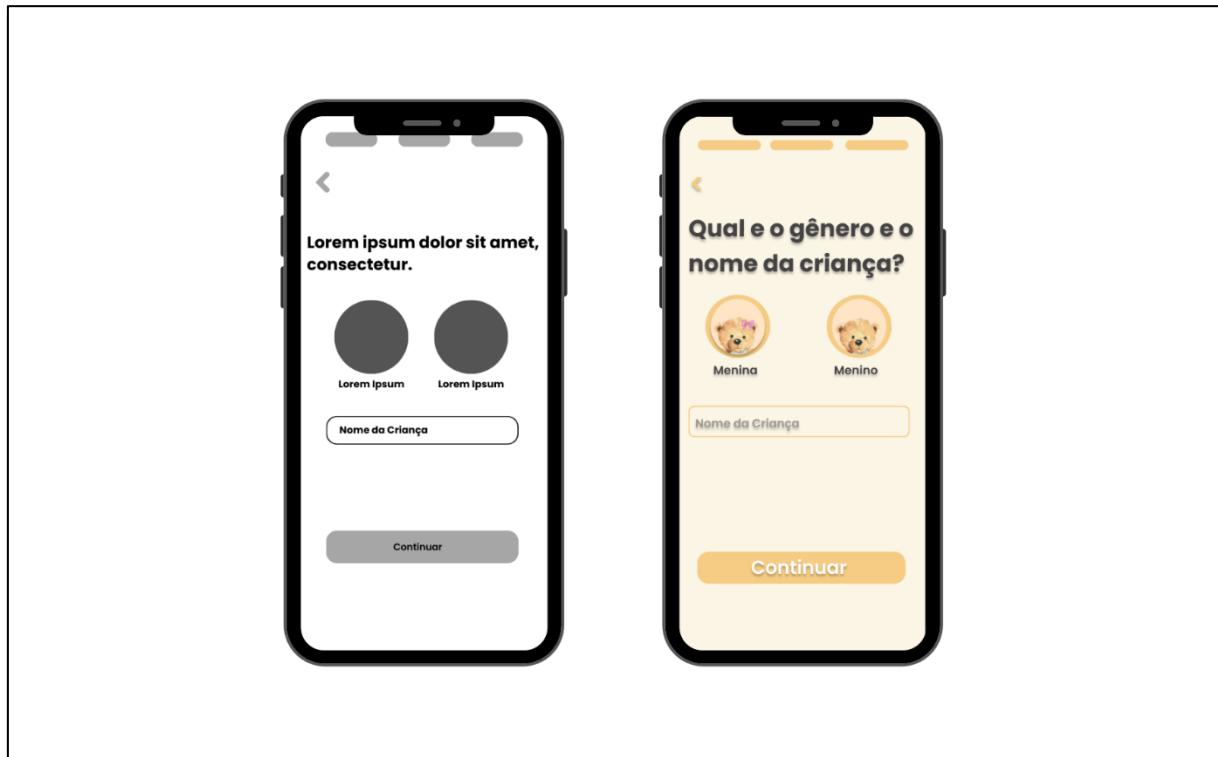
Figura 68 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Perfil.



Fonte: De autoria própria, 2024.

Há seguir, apresenta-se o próximo passo da criação do perfil da criança, onde o responsável deve informar o gênero e o nome da criança:

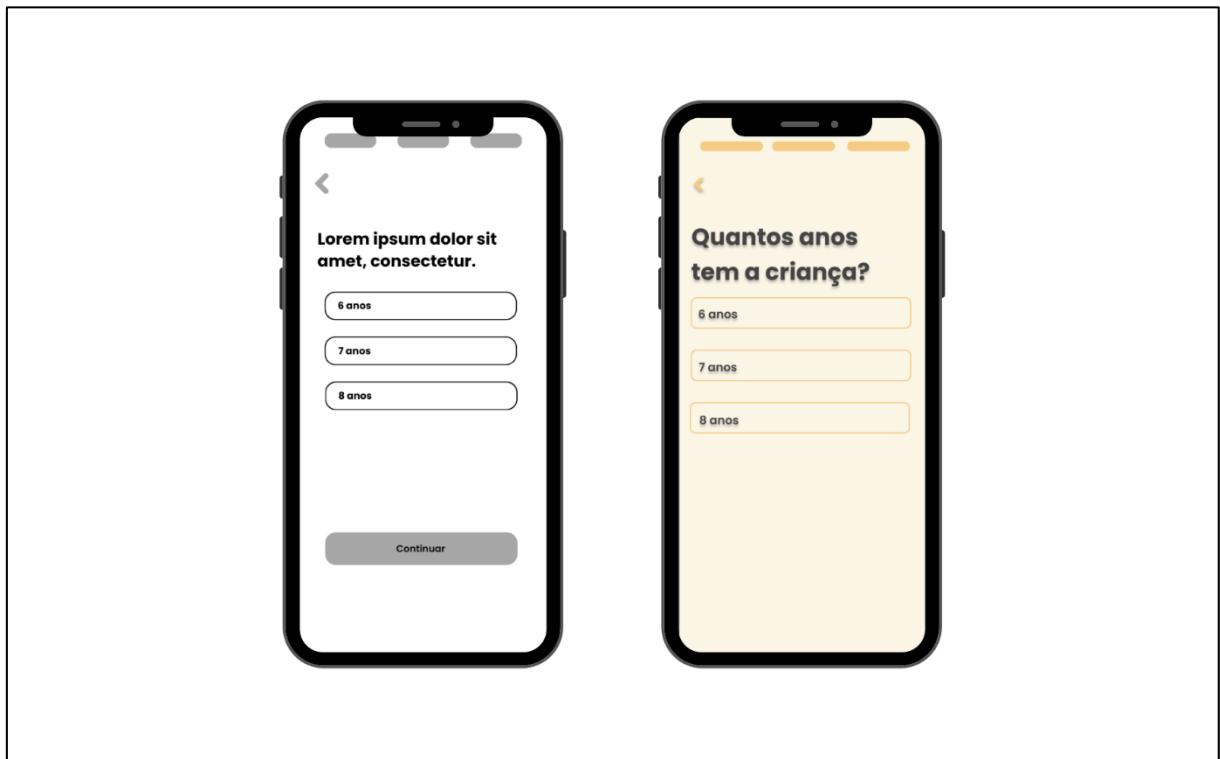
Figura 69 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Perfil Nome e Gênero



Fonte: De autoria própria, 2024.

Abaixo, apresenta-se a continuação da criação do perfil da criança, onde o responsável informa a idade da criança, focando na faixa etária de 6 a 8 anos:

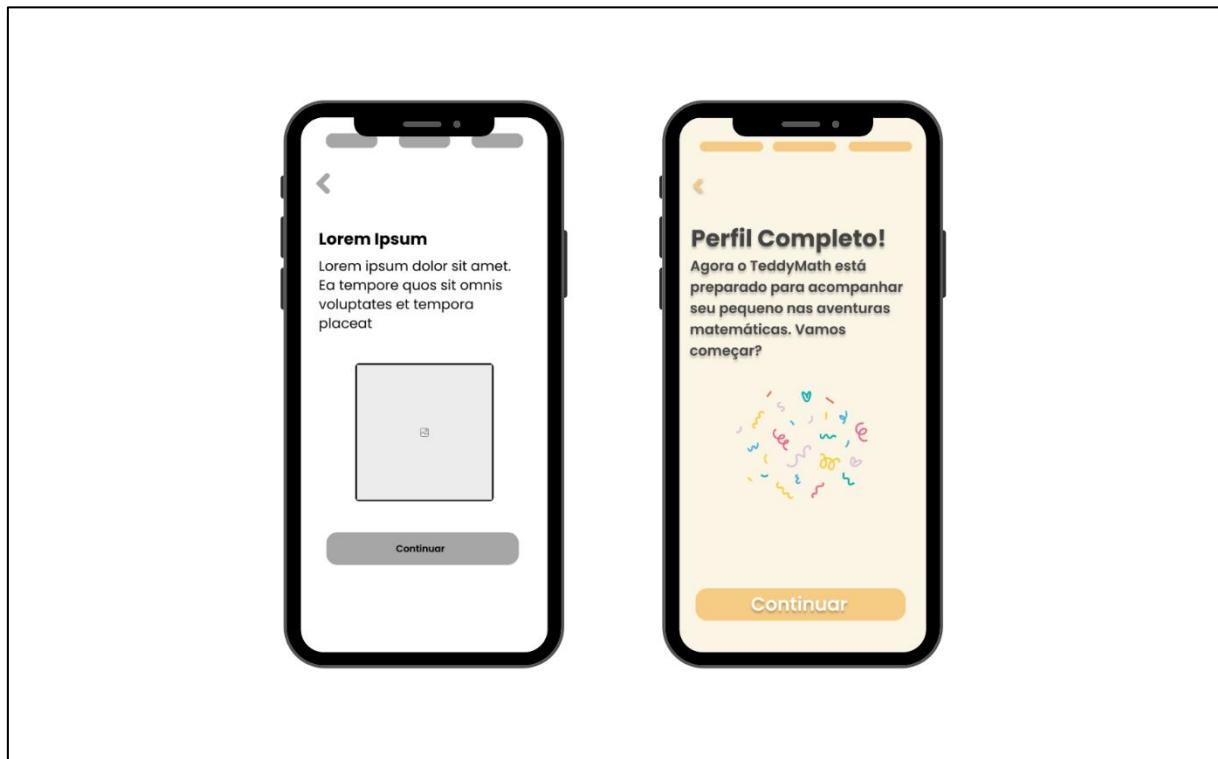
Figura 70 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Perfil Idade



Fonte: De autoria própria, 2024.

Abaixo, é mostrado a tela que aparece após o responsável concluir a criação do perfil da criança:

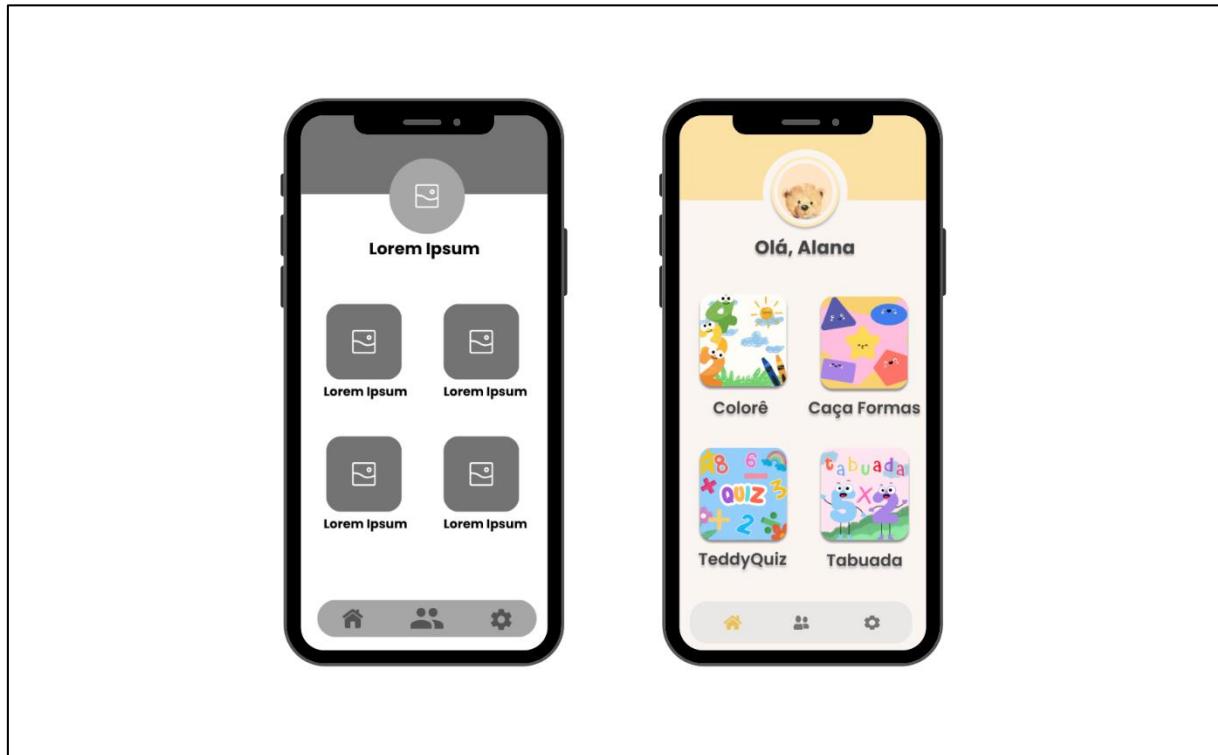
Figura 71 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Perfil Completo.



Fonte: De autoria própria, 2024.

Após concluir essa etapa, o responsável já pode entregar o celular para criança, onde essa pode escolher entre 3 jogos e 1 tabuada qual ela prefere jogar, dentre eles Colore, um jogo focado na pintura dos números; Caça-Formas, um jogo de memória das formas geométricas; TeddyQuiz, que estimula o raciocínio lógico por meio de testes de adição, subtração, multiplicação e divisão; e Tabuada, que tem como objetivo aprender a tabuada quando a criança pressionar o botão no urso, e este tocará um áudio com as tabuadas. Como é ilustrado na figura abaixo:

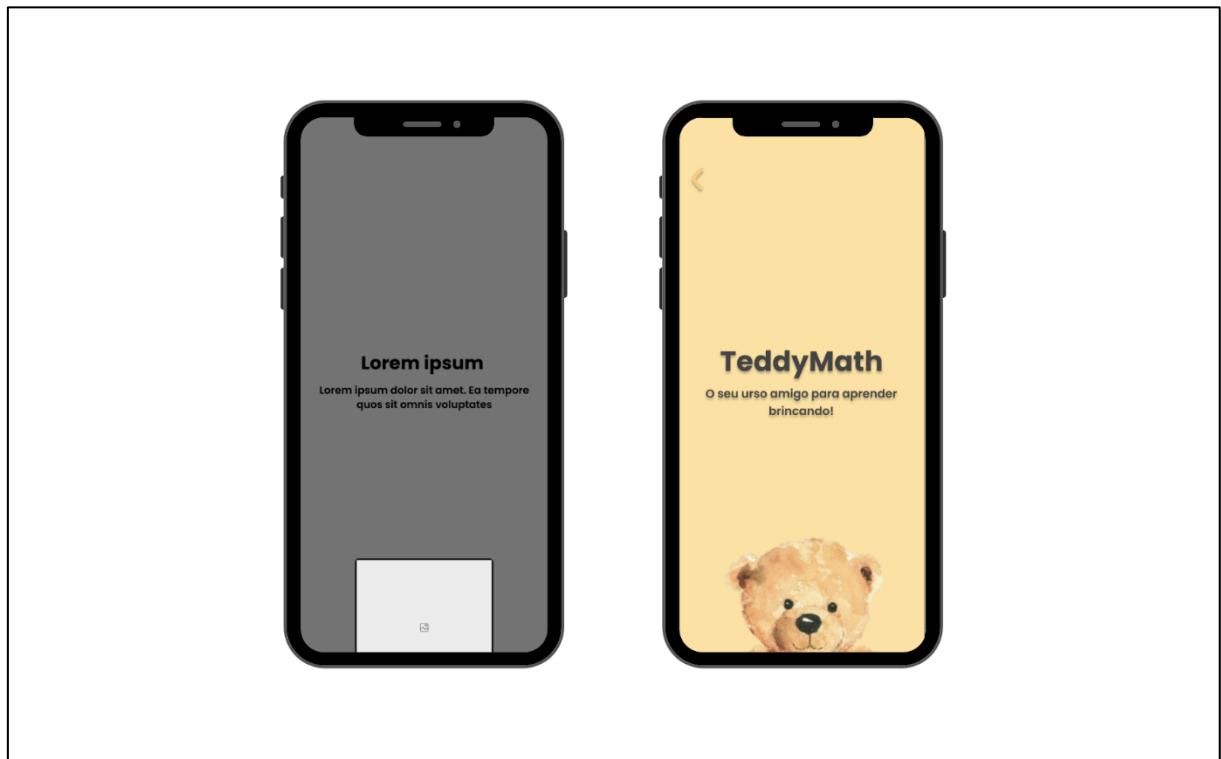
Figura 72 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Menu.



Fonte: De autoria própria, 2024.

Após a criança selecionar o jogo é mostrada a tela de carregamento antes do jogo iniciar, ilustrado na figura abaixo:

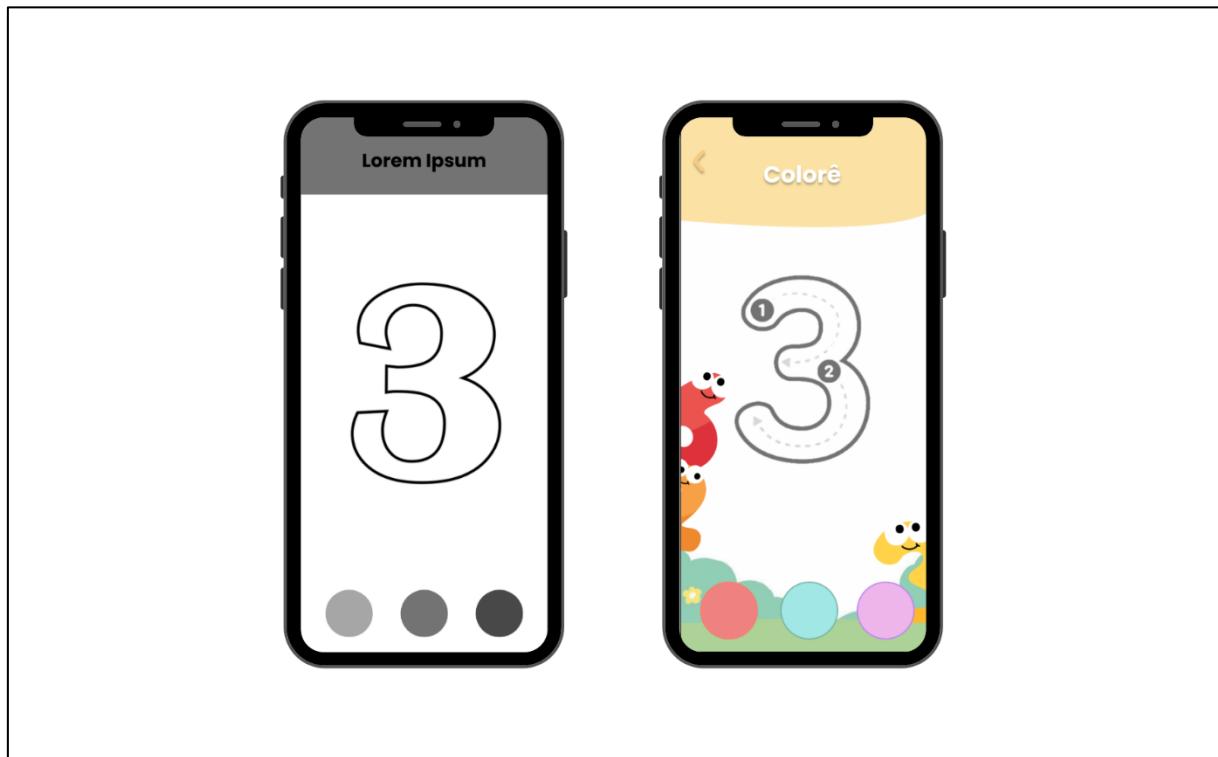
Figura 73 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Splash.



Fonte: De autoria própria, 2024.

Iniciando pelo primeiro jogo selecionado, é mostrado o Colorê, um jogo interativo que estimula a criatividade e o aprendizado das crianças por meio da pintura dos números, onde está escolhe o número e se dispõe das cores para pintar, como ilustrado adiante:

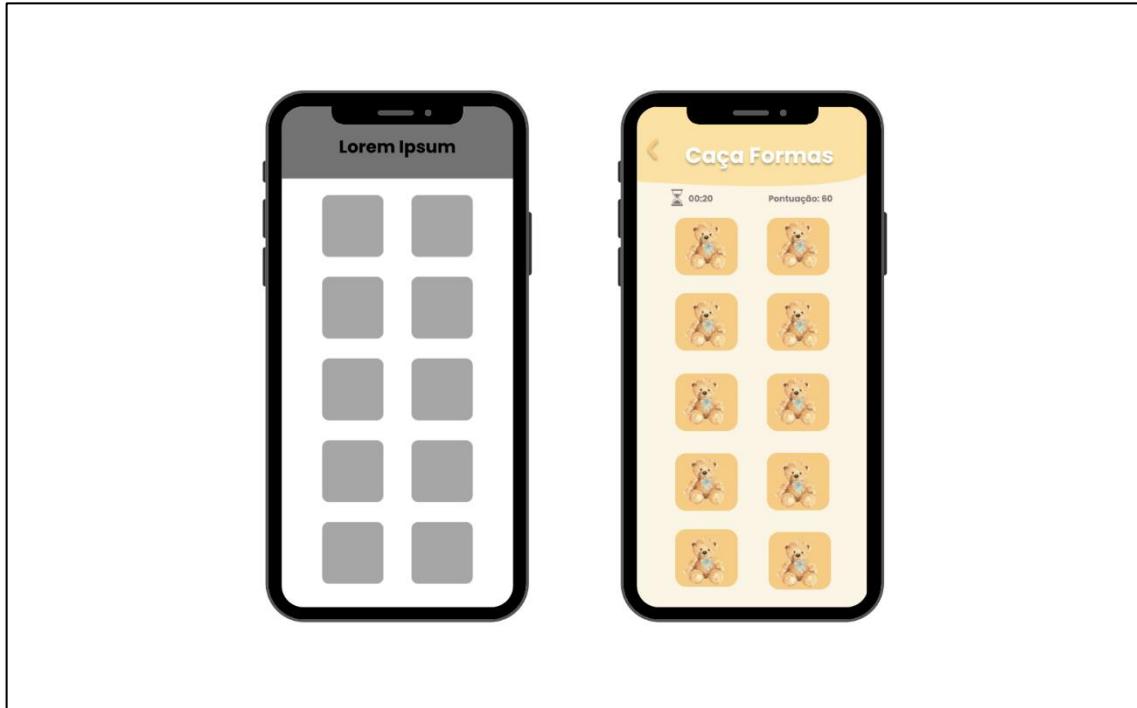
Figura 74 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Colorê.



Fonte: De autoria própria, 2024.

O jogo Caça Formas é um jogo da memória onde a criança tem um tempo de 20 segundos para acertar os pares das formas geométricas. Este ajuda no desenvolvimento da memória visual e o conhecimento das crianças. Conforme mostrado na figura a seguir:

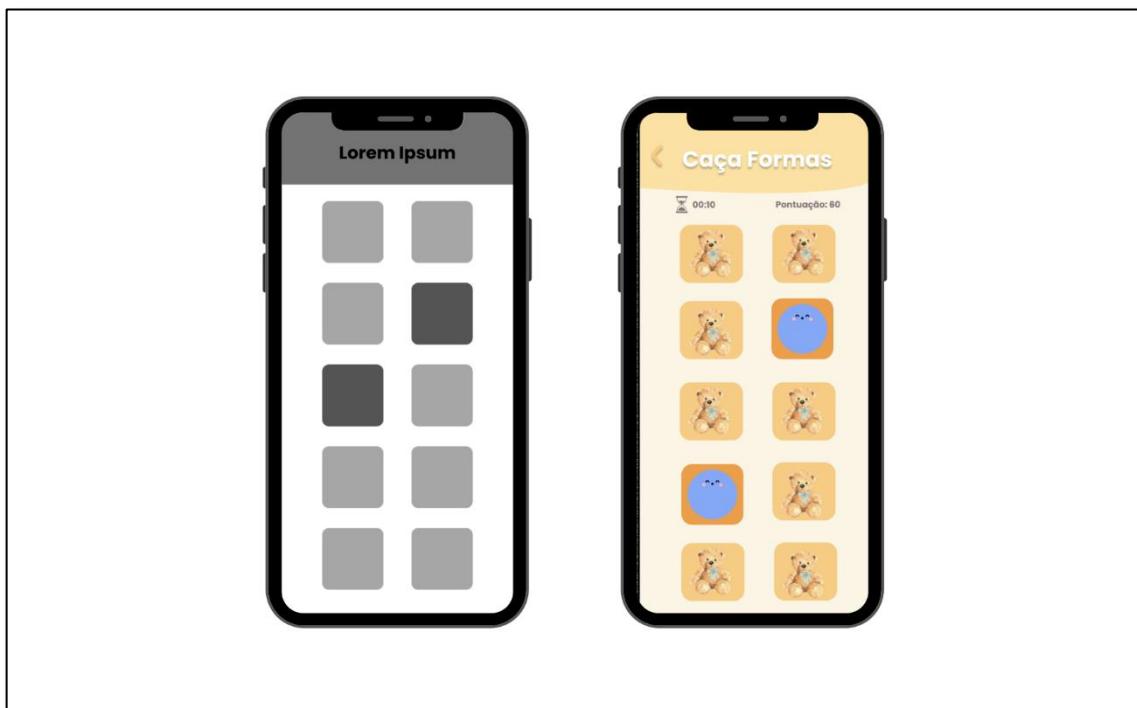
Figura 75 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Jogo Caça Formas.



Fonte: De autoria própria, 2024.

Subsequente, está a continuação da simulação do jogo:

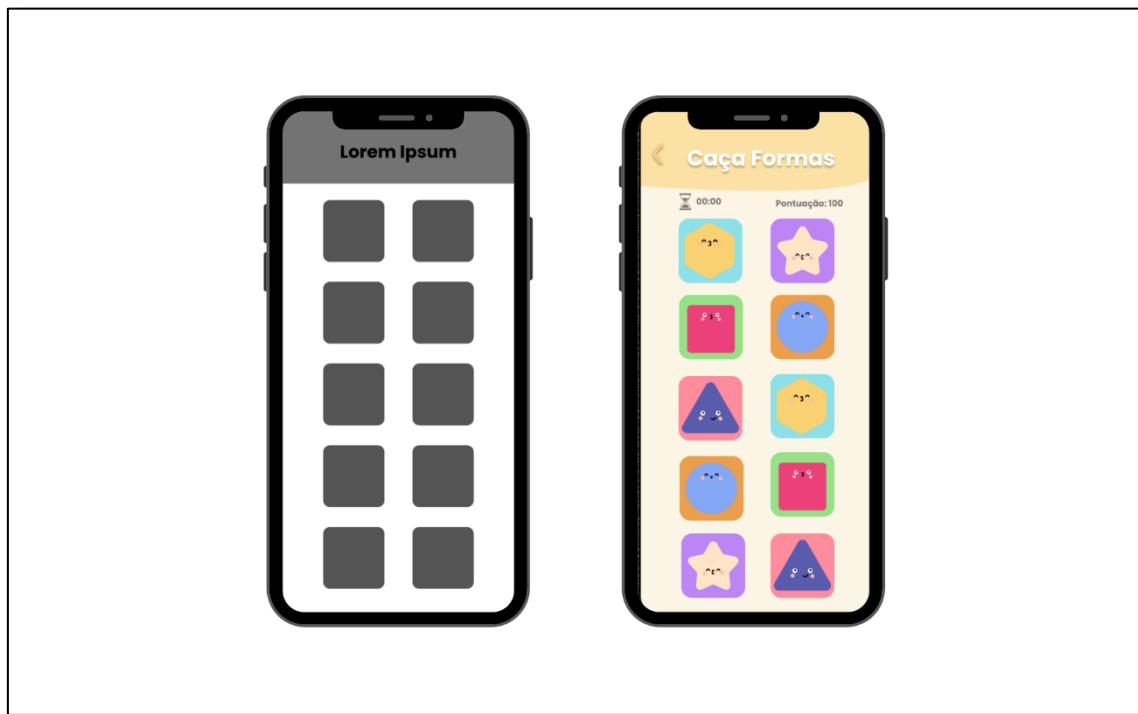
Figura 76 - Wireframe de Baixa e Alta Fidelidade da Tela Caça-Formas (Continuação).



Fonte: De autoria própria, 2024.

Por fim, é apresentado a tela final do jogo simulando a pontuação 100:

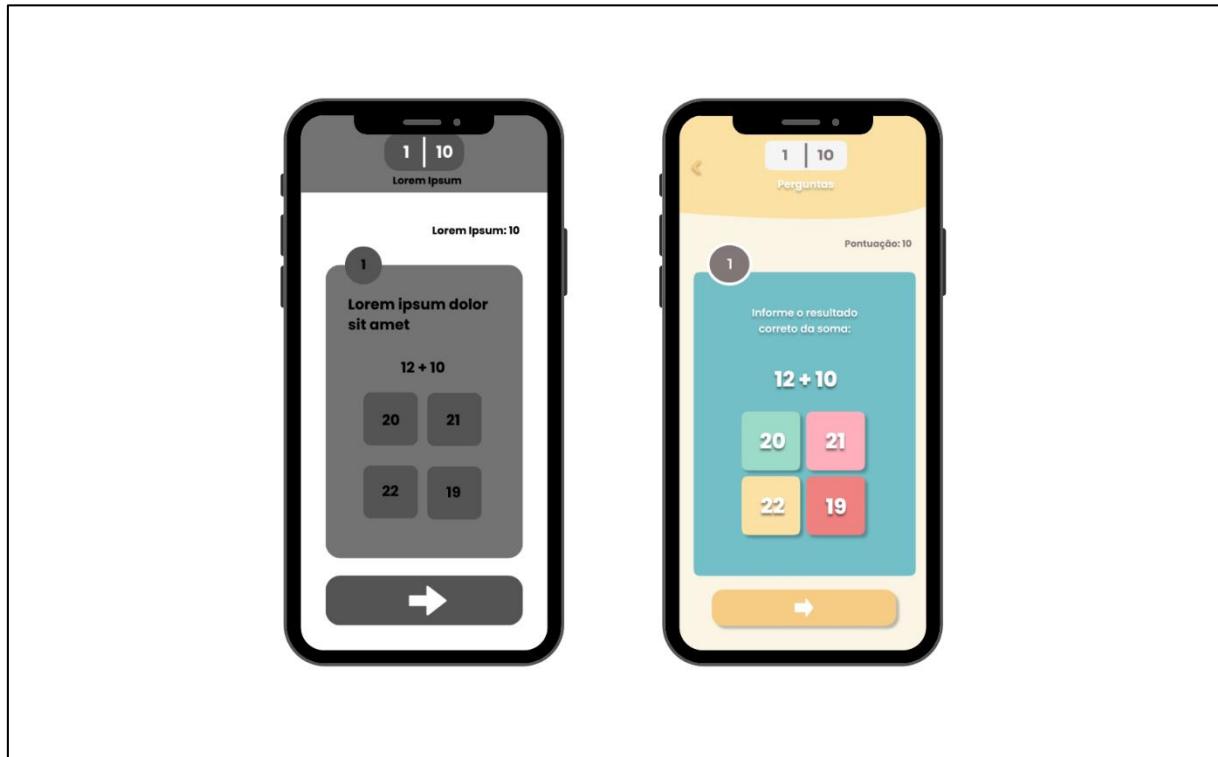
Figura 77 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Caça Formas (Concluído).



Fonte: De autoria própria, 2024.

Na ilustração a seguir, observa-se o jogo TeddyQuiz um jogo educativo contendo 10 perguntas sobre adição, subtração, multiplicação e divisão, assim promovendo o raciocínio lógico e matemático das crianças:

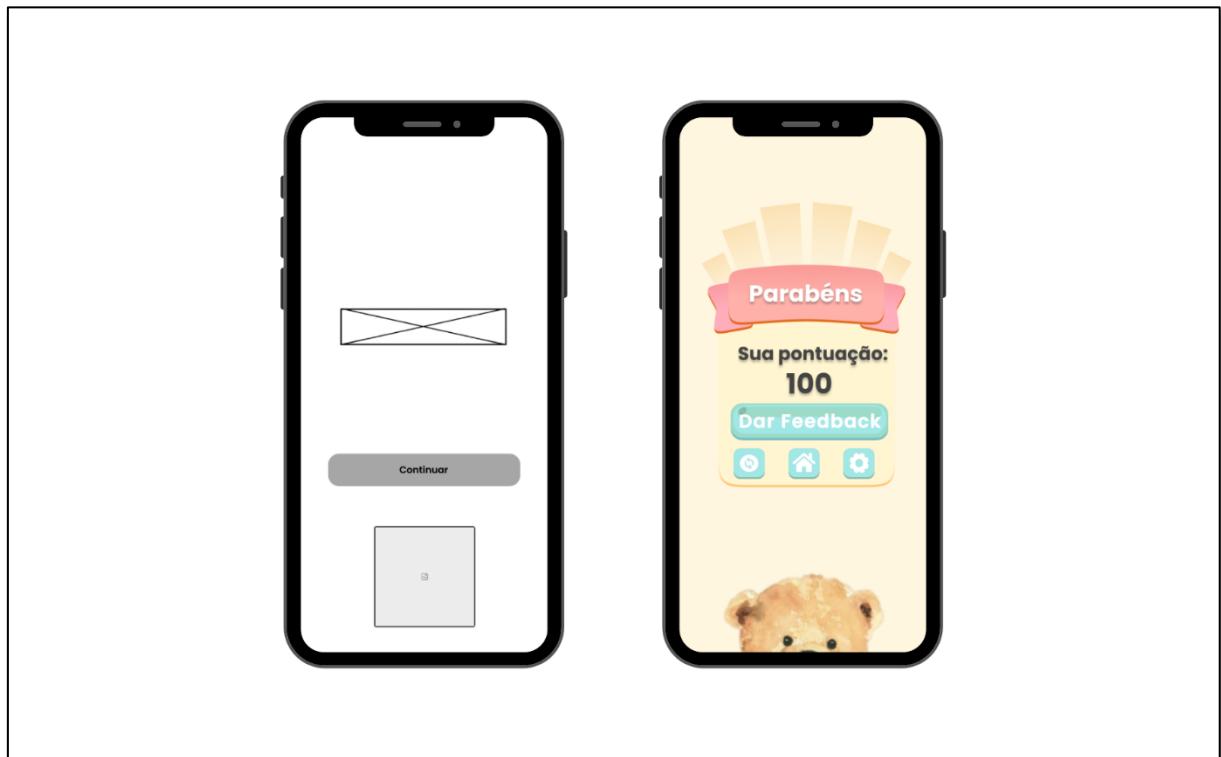
Figura 78 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela TeddyQuiz.



Fonte: De autoria própria, 2024.

Após a criança finalizar qualquer jogo, a pontuação da criança é salva no banco de dados e este redireciona para a tela correspondente a sua pontuação, como mostrado na figura abaixo, e a partir dela o urso emite um feedback em áudio, bom ou ruim, quando clicar no botão “Dar Feedback”:

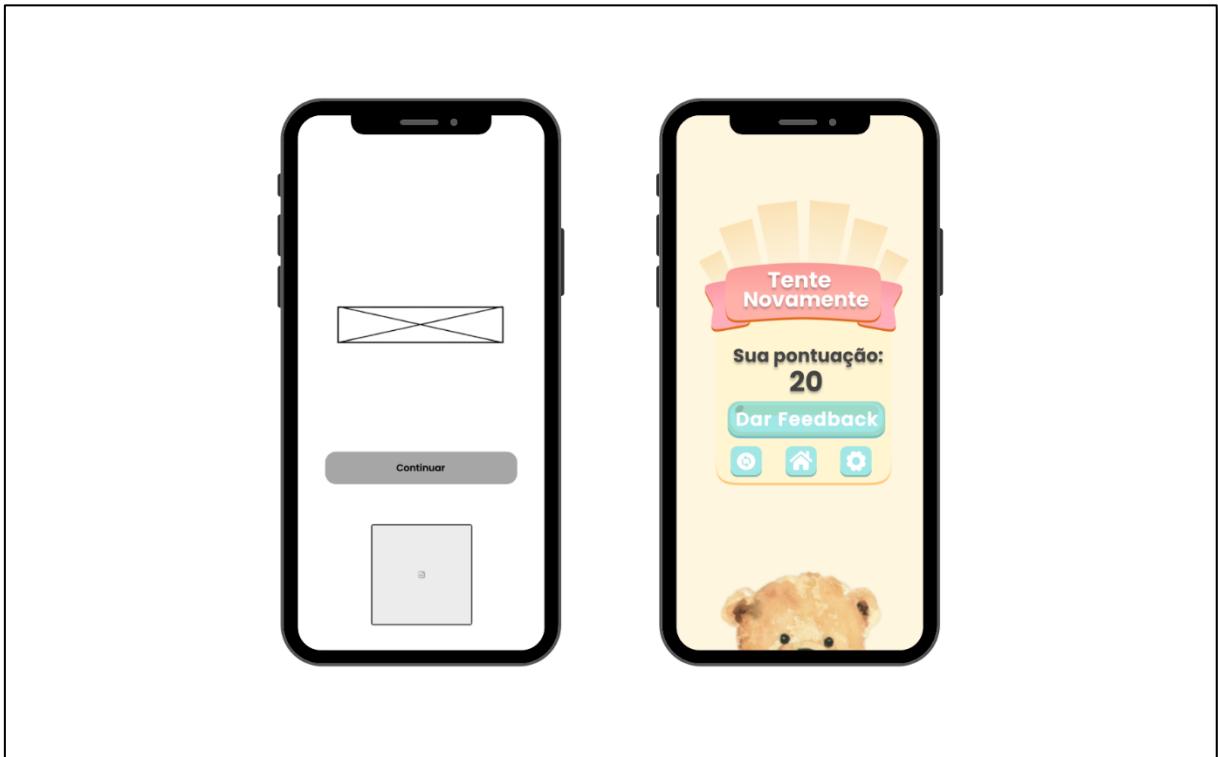
Figura 79 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Parabéns.



Fonte: De autoria própria, 2024.

Há seguir, é mostrado a mesma tela, em baixa e alta fidelidade, porém na versão tente novamente emitindo o feedback ruim:

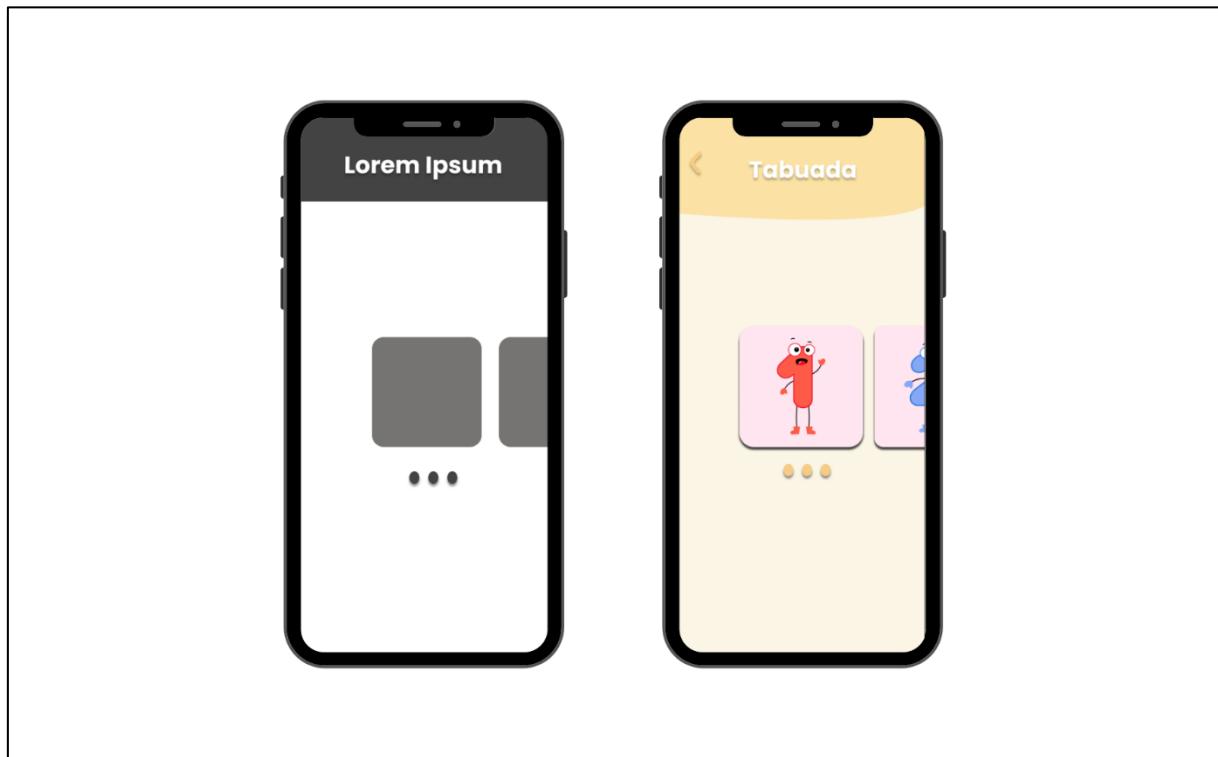
Figura 80 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Tente Novamente



Fonte: De autoria própria, 2024.

Na figura abaixo é ilustrado uma tela interativa conectada com o urso. Quando a criança estiver com o urso e este estiver recitando as tabuadas, ela pode acompanhar por escrito clicando nos ícones da tabuada referente ao que o urso estiver reproduzindo:

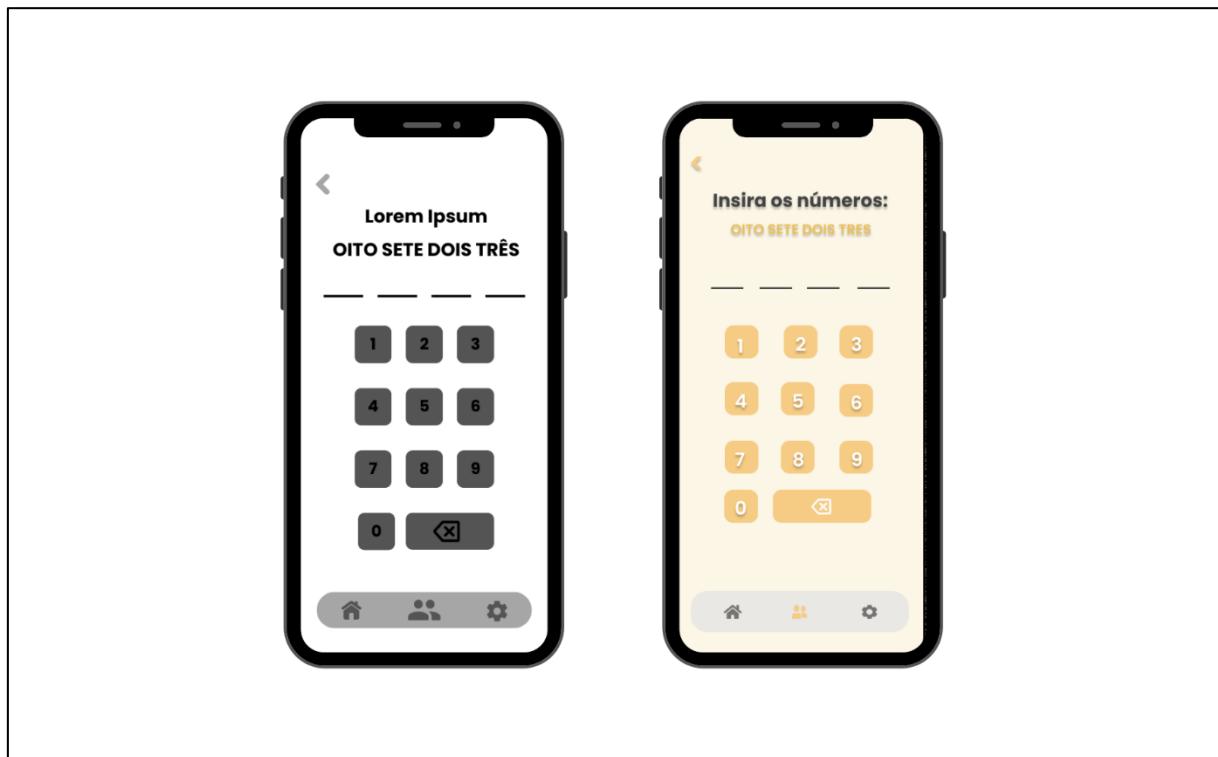
Figura 81 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Tabuada.



Fonte: De autoria própria, 2024.

Subsequente, é apresentado a tela de acesso para a área do responsável, onde somente estes podem acessar e acompanhar o progresso da criança. A tela conta com uma “senha” onde estes colocam os números mostrados e se dispõe do conteúdo.

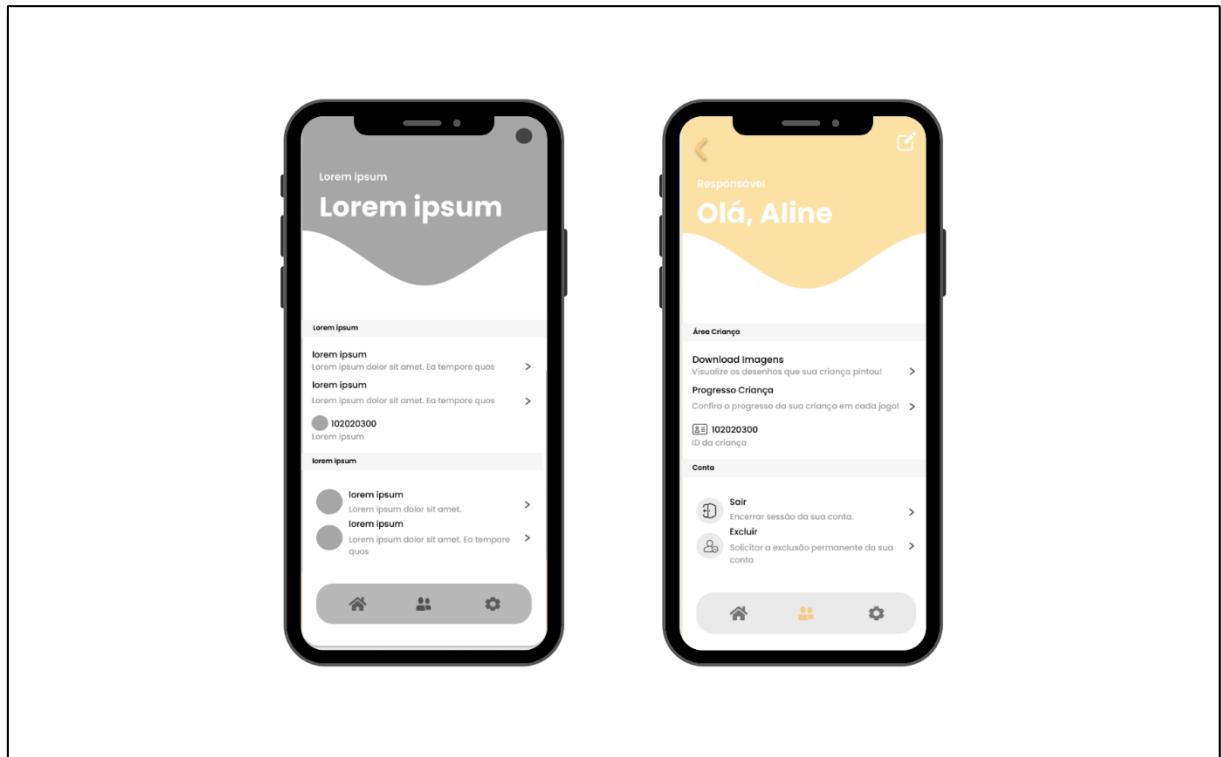
Figura 82 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Acesso Área do Responsável



Fonte: De autoria própria, 2024.

Adiante, a tela exibida é a área do responsável, onde esse tem acesso a alguns conteúdos como *download* das imagens dos números pintados no jogo Colorê, monitorar o progresso da criança e também tem a opção de sair ou excluir a conta.

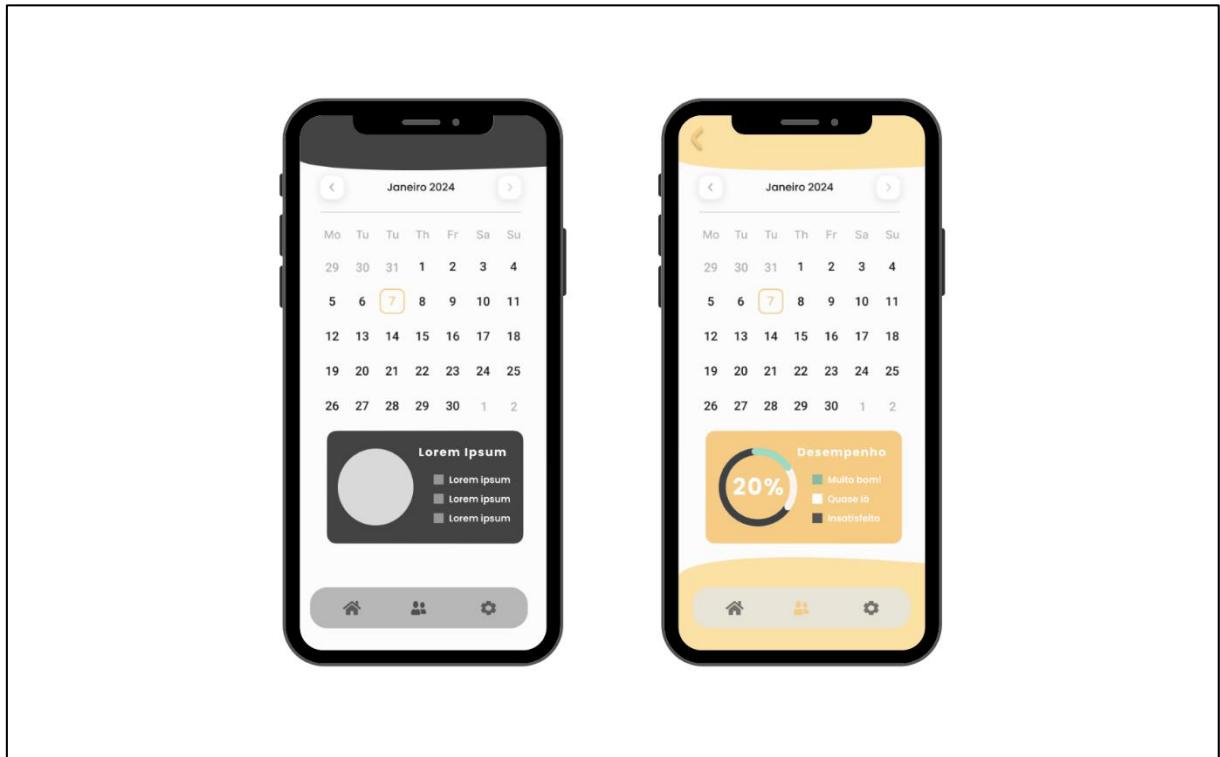
Figura 83 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Área do Responsável.



Fonte: De autoria própria, 2024.

Ao clicar em “Progresso Criança”, o responsável tem acesso a um calendário mostrando os dias que a criança utilizou o aplicativo e assim podendo acompanhar o seu desempenho nos jogos, analisando a sua evolução no aprendizado de matemática, como mostrado a seguir:

Figura 84 - Wireframe Baixa e Alta Fidelidade do Aplicativo: Tela Progresso Criança.

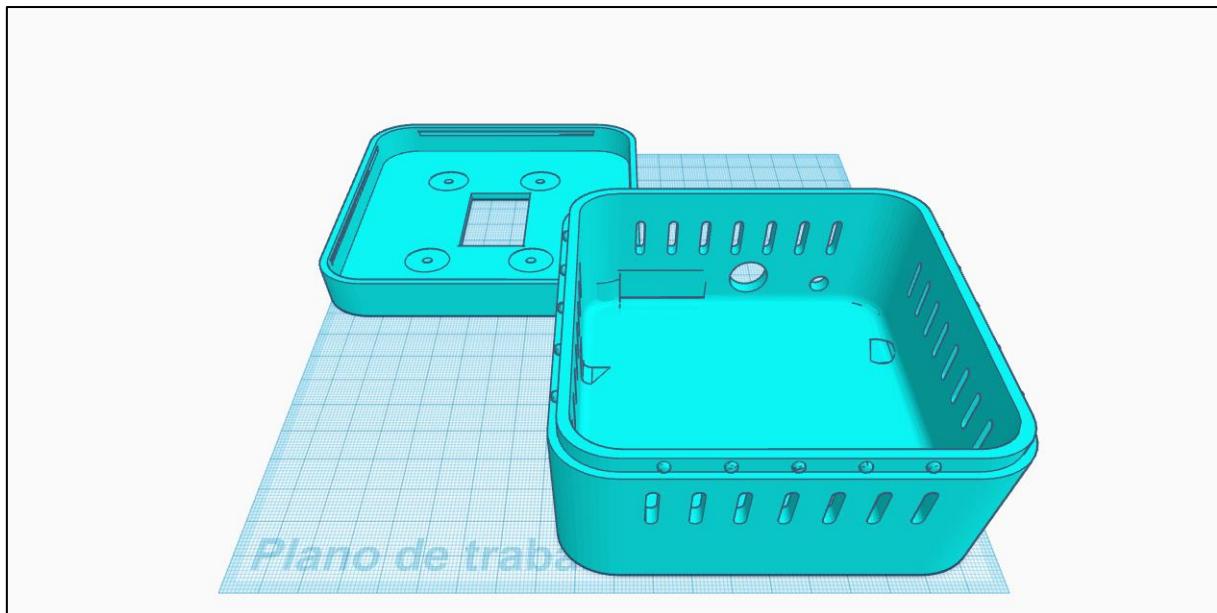


Fonte: De autoria própria, 2024.

3.7 Modelagem 3D para o Hardware

Nesta seção é mostrado a modelagem 3D que foi utilizado na produção do projeto. Abaixo encontra-se a figura da case em impressão 3D do projeto TeddyMath, onde foram colocados todos os seus componentes para a produção do dispositivo IoT.

Figura 85 - Modelagem 3D Case do Circuito Eletrônico.



Fonte: De autoria própria, 2024.

Subsequente na figura abaixo é mostrada a mesma case após realizar a impressão 3D.

Figura 86 - Case do Circuito Eletrônico.



Fonte: De autoria própria, 2024.

3.8 TeddyMath: Brinquedo IoT

O projeto combina uma aplicação *hardware* e um aplicativo desenvolvido utilizando React Native e C++. A principal finalidade do brinquedo interativo é auxiliar as crianças a aprender matemática, fornecendo feedback durante os jogos e incentivando-as a compreender o assunto de uma maneira lúdica.

O brinquedo IoT, um ursinho de pelúcia, ilustrado na figura abaixo, foi projetado para interagir com as crianças de forma lúdica e educativa. Equipado com um alto-falante, o urso oferece feedback sobre os jogos, motivando os jovens alunos a abordar a matemática de uma maneira divertida e envolvente. O aparelho também possui um botão na sua mão que ao ser pressionado, faz com que o urso recite a tabuada, reforçando o aprendizado de matemática.

Figura 87 - TeddyMath Brinquedo IoT.

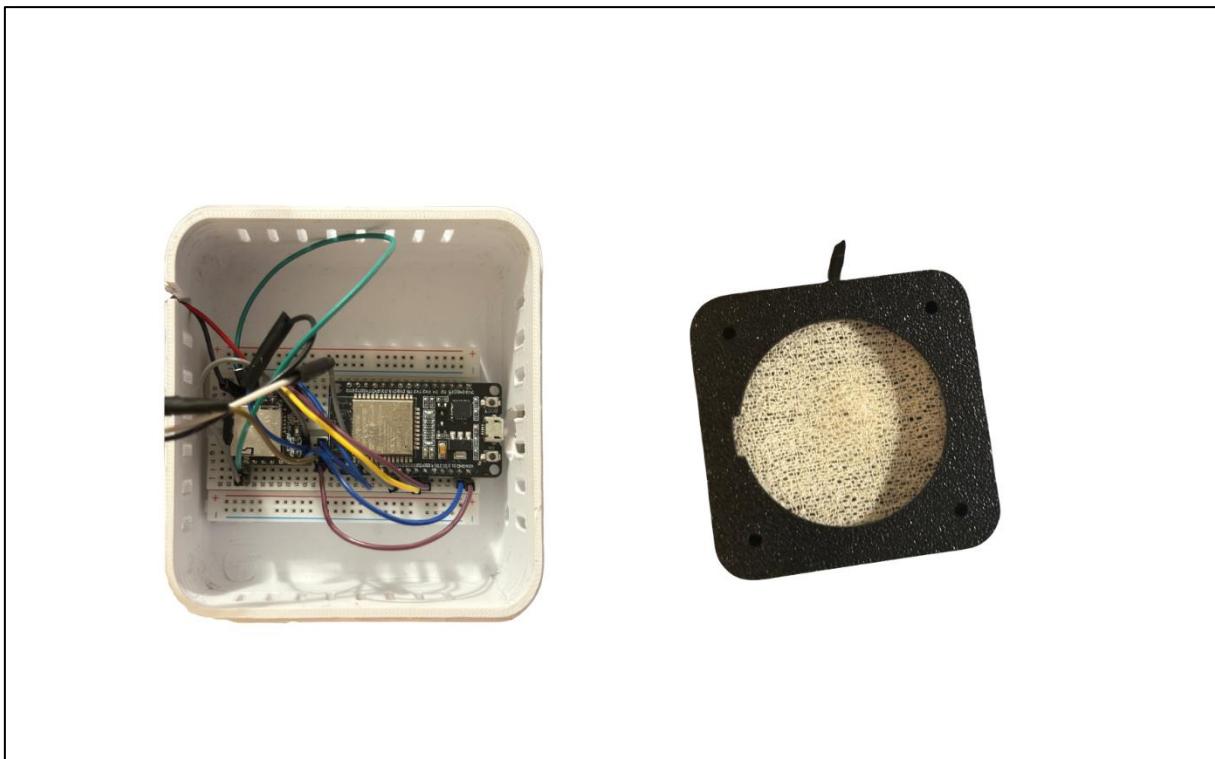


Fonte: De autoria própria, 2024.

O urso é conectado com o aplicativo e tem sua funcionalidade principal com os jogos, quando a criança conclui um jogo, o aplicativo exibe uma tela de *feedback* com a pontuação e a opção de apertar o botão "Dar Feedback". Ao pressionar esse botão, o

ursinho de pelúcia, que inclui um microcontrolador ESP32 e um módulo DFPlayer Mini, fornece um *feedback* sonoro, reforçando o aprendizado. O ESP32 está conectado a um botão que ativa o áudio, e o DFPlayer Mini reproduz os arquivos de som armazenados em um cartão microSD. Esse circuito eletrônico, mostrado abaixo, composto pelo ESP32, DFPlayer Mini e o alto-falante que está alocado na caixa preta isolado por uma espuma para um som melhor, o circuito é configurado para criar uma experiência interativa e educativa.

Figura 88 - Circuito do Dispositivo.



Fonte: De autoria própria, 2024.

4 CONSIDERAÇÕES FINAIS

Com base nas dificuldades enfrentadas por muitas crianças do ensino fundamental no aprendizado de matemática básica, levanta-se a hipótese de que a implementação de um aplicativo móvel conectado a um brinquedo interativo, com jogos relacionados a conteúdos matemáticos, poderia ser uma abordagem inovadora e eficaz para enfrentar esses desafios de forma acessível. No entanto, essa hipótese ainda não foi totalmente validada. Para alcançar resultados mais consistentes, seria necessário implementar essa solução em larga escala, tanto nas escolas quanto nas casas das

crianças com dificuldades com o assunto. Uma análise longitudinal das repercussões dessa abordagem poderia demonstrar que a combinação de elementos lúdicos, interativos e pedagogicamente estruturados proporcionada por essa ferramenta tecnológica desperta o interesse das crianças pela matemática desde cedo, o que, por sua vez, facilita sua compreensão e retenção de conceitos fundamentais. Portanto, embora os resultados iniciais sejam encorajadores, a confirmação ou refutação dessa hipótese ainda depende de mais testes e avaliações. Pode-se concluir que, embora a ferramenta tenha o potencial de impactar positivamente a educação matemática, seu sucesso depende de uma implementação mais ampla e avaliação contínua. Somente por meio de testes em larga escala e monitoramento constante será possível verificar se a solução contribui genuinamente para o desenvolvimento da proficiência matemática entre as crianças. O projeto, portanto, abre um caminho promissor para a integração da tecnologia na educação, mas seu impacto definitivo ainda precisa ser fundamentado.

Este projeto é voltado para crianças do ensino fundamental que enfrentam dificuldades em matemática, o presente estudo adota uma abordagem qualitativa e exploratória com o objetivo de avaliar a eficácia de um dispositivo educacional no contexto escolar. A pesquisa se baseia em uma variedade de fontes, incluindo sites, livros, artigos de notícias e uma análise abrangente da Base Nacional Comum Curricular (BNCC). Essa compilação de informações aborda os desafios persistentes na aprendizagem de matemática, especialmente na educação primária, que formam a base para o desenvolvimento do dispositivo educacional. No entanto, como qualquer projeto, o TeddyMath também apresentou algumas limitações como o uso do ESP32 para o controle do urso interativo, onde esse se conecta com o aplicativo apenas se houver Wi-Fi disponível, o que pode limitar o uso do sistema em ambientes sem acesso à internet.

Em conclusão, o uso da tecnologia na educação matemática, como exemplificado pelo TeddyMath, tem o potencial de tornar o aprendizado mais acessível e eficaz, além de aumentar a autoconfiança das crianças. Os aprimoramentos futuros podem incluir reconhecimento de voz, onde por meio desse método o dispositivo poderá reconhecer comandos e perguntas realizadas pela criança e com a junção da inteligência artificial,

criar uma experiência personalizada com base no nível de aprendizado da criança, assim enriquecendo ainda mais o desenvolvimento educacional e social das crianças.

REFERÊNCIAS

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivan. **UML: Guia Do Usuário.** 2.ed. [S. I.]. GEN LTC, 2006.

BRANDT, Celia Finck; MORETTI, Méricles Thardeu. **Ensinar e aprender matemática: possibilidades para a prática educativa.** 1.ed. [S. I.]. Editoria UEPG, 2016.

COSTA, Carlos J. E- book **Desenvolvimento para Web.** Portugal: Lusocrérito, 2007. Livro Digital.

D'AMBRÓSIO, Ubiratan. **Da realidade à ação: reflexões sobre educação e matemática.** 2.ed. [S. I.]. Editorial Summus, 1986.

DURAES, Wellington; FERREIRA, Fernando H. I. B; MANZAN, Renato. E-book **Arquitetura de soluções IoT: Desenvolva com Internet das Coisas para o mundo.** [S.I.]. Casa do Código, 2022. Livro Digital.

EIS, Diego; FERREIRA, Elcio. **HTML5 e CSS3 com Farinha e Pimenta.** 1.ed. São Paulo: Tableless, 2012.

ELETRÔNICA ÔMEGA. E-book **Internet das Coisas para iniciantes com ESP-32.** 1. ed. [S.I.]. Arduino Ômega, 2021. Livro Digital.

FLANAGAN, David. **Javascript: O Guia Definitivo.** São Paulo: Bookman, 2012.

FERREIRA, Silvio. **Guia Prático de HTML.** 1.ed. [S.I.]. Universo dos Livros , 2018.

FOWLER, Martin. E-book **UML Essencial: Um Breve Guia para Linguagem Padrão.** 3.ed. [S.I.]. Bookman, 2005. Livro Digital.

GUEDES, Gilleannes T. A. E-book **UML 2 - Uma Abordagem Prática.** 3.ed. [S. I.]. Novatec, 2018. Livro Digital.

JOBSTRAIBIZER, Flávia. **Criação de Sites com CSS.** 1. ed. São Paulo: Digerati, 2008.

MELO, Ana Cristina. **Desenvolvendo Aplicações Com UML 2.2.** 3.ed. [S. I.]. Brasport, 2011.

MONK, Simon. **30 Projetos com Arduino.** 2. ed. Porto Alegre: Artmed, 2014.

NETO, William Bruno. E-book **Construindo APIs testáveis com Node.js.** [S.I.]. 2017. Livro Digital.

NIELD, Thomas. **Introdução à Linguagem SQL: Abordagem Prática Para Iniciantes.** 1.ed. São Paulo,SP: Novatec, abril.2016.

PAIXÃO,Marcus Aurélio. E-book **IoT: fundamentos e componentes (Série Universitária).** 1. ed. [S.I.]. Editora Senac São Paulo, 2024. Livro Digital.

PEREIRA, C. R. **Node.js: Aplicações web real-time com Node.js.** 1. ed. Editora Casa do Código, 2014

SANTOS, Sandro. **Introdução à IoT: Desvendando a Internet das Coisas.** 1.ed. [S. I.]. Clube de Autores, 2018.

SCHEIDT,Felipe Alex. **Fundamentos de CSS.** 1.ed. Foz do Iguaçu: Outbox Interativa,2015.

SILVA, Maurício Samy. **React - Aprenda Praticando: Desenvolva Aplicações web Reais com uso da Biblioteca React e de Seus Módulos Auxiliares.** 1. ed. São Paulo: Novatec, 2021.

TAYLOR, Allen G. **SQL Para Leigos.** 8.ed. [S. I.]. Alta Books, 2016.

WIENER, Richard S; PINSON, Lewis J. **C ++ Programação Orientada para Objeto : Manual Prático e Profissional.** 1.ed.[S.I.]. Makron Books, 1991.

ZAKAS, Nicholas C. **JavaScript de Alto Desempenho.** 1.ed. [S.I.]. Novatec, jul. 2010.

BRASIL, Brasil. **Secretaria de Educação Fundamental. Parâmetros curriculares nacionais: matemática** – MEC/SEF. Brasília, p. 15,1997.

BRITO, Ricardo W. **Bancos de dados nosql x sgbds relacionais: análise comparativa.** Fortaleza,n. 1, p. 2. 2010.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular** – MEC/SEF. Brasília, p. 263, 2017.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular** – MEC/SEF. Brasília, p. 264, 2017.

NASCIMENTO, Allyson; SOUZA, Carlos; OLIVEIRA, Georgia. **Controle de Iluminação Através da Internet Utilizando as Tecnologias ReactJS, Firebase e ESP32.** Natal, n. 1, v. 6, p. 47, 2023.

OLIVEIRA, Jefferson Penna. **Uma abordagem para captura automática de links de rastreamento entre especificação de casos de uso e modelos UML.** Recife, n. 1, p. 25, 2015.

PEREIRA, Luiz Antônio de Moraes. **Análise e Modelagem de Sistemas com a UML: Com Dicas e Exercícios Resolvidos.** Rio de Janeiro, n. 1, p. 7 - 23, 12 jan. 2011.

SILVA, Rogério Oliveira; MARTINS, Bonny Rodrigues; DINIZ, Walisson Gama. **Tecnologias em Projeção.** [S.I.]. n. 1, v. 8, p. 87 ,27 ago. 2017.

TORQUATO, Uanderson Rosário. SILVA, Flaviana dos Santos. **ENSINO DA MATEMÁTICA NA EJA: ESTUDO BIBLIOGRÁFICO À LUZ DA APRENDIZAGEM SIGNIFICATIVA.** REAMEC - Rede Amazônica de Educação em Ciências e Matemática, Cuiabá, v. 12, p. 19, 2024. Disponível em: <https://periodicoscientificos.ufmt.br/ojs/index.php/reamec/article/view/17241>. Acesso em: 06 jun. 2024.

AGUIAR, Gustavo Stor. **NODE.JS: Estudo Tecnológico e Desenvolvimento Full-stack Javascript de Plataforma De competições em Problemas Algorítmicos.** 2015. Dissertação (Graduação em Ciência da Computação) – Universidade Federal de Pernambuco, Recife, 2015.

BAIÃO, Francisco José. **NODEFUNCIONALIDADES E TECNOLOGIAS DA IMPRESSORA 3D.** 2012. Monografia (Graduação em Engenharia de Computação) – Universidade São Francisco, Itatiba, 2012.

BRAGA, Rondinelly Castelo. **Desenvolvimento Nativo Vs React Native: um Estudo de Portabilidade do Jogo Híbrido Coup Acessível.** 2019. Dissertação (Graduação em Sistemas e Mídias Digitais) – Universidade Federal do Ceará, Instituto UFC Virtual, Fortaleza, 2019.

CALDEIRA, Carlos Pampulim. **Introdução ao Html: (HyperText Markup Language).** 2015. Dissertação (Departamento de Informática) –Universidade de Évora, Évora, Portugal, 2015.

COUTO, Moacyr Azevedo Júnior; VIRTUOSO, Guilherme H. F; MARTINS, Paulo João; **Propriedades Desejáveis a uma Linguagem de Programação: Uma Análise Comparativa entre as Linguagens C, C++ e Java.** 2011. Dissertação (Curso Ciência da Computação) - Universidade do Extremo Sul Catarinense (UNESC), Criciúma, Santa Catarina, 2011.

FERREIRA, Matheus Alexandrino. **Desenvolvimento de tecnologia assistiva para dosagem de medicamentos líquidos.** 2023. (Graduação em Engenharia Elétrica) Repositório Digital - (Instituto Federal de Educação, Ciência e Tecnologia da Paraíba) – Campos João Pessoa, 2023.

GALVÃO, Pedro Sereno. **Comprehensive Repository Analysis of Mobile Projects Built with React Native.** 2018. Dissertação (Graduação em Ciência da Computação) –Universidade Federal de Pernambuco – UFPE, Centro de Informática, Recife, 2018.

LEITÃO, Paulo Rogério Matheus Bonfim. **Desenvolvimento Nativo Vs React Native: uma Análise Comparativa na Codificação de uma Aplicação para Fitness.** 2019. Dissertação (Graduação em Sistemas e Mídias Digitais) – Universidade Federal do Ceará, Instituto UFC Virtual, Fortaleza, 2019.

LIMA, Gabriel. **SmartLock Lite: um sistema de controle de acesso usando o microcontrolador ESP32.** 2022. Relatório Teórico (Instituto de Computação) – Universidade Estadual de Campinas. Campinas, 2022.

MAIA, Ricardo J. S. **Interfaces tátteis baseadas em HTML5/ CSS3/ JavaScript.** 2011.Dissertação (Mestrado Integrado em Engenharia Informática e Computação) – Faculdade de Engenharia da Universidade do Porto, Portugal, 2011.

MASSOLA, Silze Cristina; PINTO, Giuliano Scombatti. **O uso da Internet Das Coisas (IOT) a favor da saúde.** 2018. Dissertação – Faculdade de Tecnologia de Taquaritinga (FATEC) Campinas, 2018.

OLIVEIRA, Jefferson Penna. **Uma abordagem para captura automática de links de rastreamento entre especificação de casos de uso e modelos UML.** 2015. Dissertação (Pós-Graduação em Informática Aplicada) – Universidade Federal Rural de Pernambuco, Recife, 2015.

REIS, João Anselmo Bandeira. **Cubinho: Artefato Robótico Motivacional para Pessoas com Transtorno do Espectro Autista Utilizando o Aplicativo Prevenir.** 2021. Monografia apresentada como requisito parcial (Graduação no Curso de Engenharia da Computação) – Universidade de Brasília, Instituto de Ciências Exatas, Brasília, 2021.

RICARTE, Ivan Luiz Marques. **Programação Orientada a Objetos com C++.** 1995. Dissertação (Departamento de Engenharia de Computação e Automação Industrial) – Faculdade de Engenharia Elétrica e Universidade Estadual de Campinas, Campinas, 1995.

SANTOS, Chayanne Barbosa. **Aprimoramento de tecnologia assistiva para dosagem de medicamentos líquidos a pessoas com deficiência visual.** 2023. (Graduação em Engenharia Elétrica) Repositório Digital - (Instituto Federal de Educação, Ciência e Tecnologia da Paraíba) – Campos João Pessoa, 2023.

SANTOS, Wallef Ferreira. **Equipamento de baixo custo para monitorar temperatura e umidade de forma contínua e remota: aplicação na compostagem.** 2021. Trabalho de Conclusão de Curso (Curso de Engenharia Ambiental e Sanitária) – Universidade Federal de Alagoas, Centro de Tecnologia, Maceió, 2021.

VITAL, Laiane Ferreira; TEIXEIRA, Verônica Rejane Lima. **O Uso das Tecnologias no Ensino da Matemática nos Anos Finais do Ensino Fundamental.** 2022. Dissertação (TCC em Ciências Humanas) – Faculdade De Ciências Humanas do Sertão Central - FACHUSC, 2022.

AMORIM, Matheus de Souza; SCHEIBNER, Eduardo. **Projeto Arduino para alimentação automatizada de animais.** REFAQI - REVISTA DE GESTÃO EDUCAÇÃO E TECNOLOGIA, [S. I.], v. 3, n. 1, p. 14, 2018. Disponível em: <https://refaqi.faqi.edu.br/index.php/refaqi/article/view/114>. Acesso em: 15 maio. 2024.

DUTRA, Aldeci dos Santos. **DIFÍCULDADE NA APRENDIZAGEM DE MATEMÁTICA NO CONTEXTO DO ENSINO FUNDAMENTAL.** Revista Humanidades e Inovação, [S. I.], v. 6, n. 12, p. 178, 2019. Disponível em: <https://revista.unitins.br/index.php/humanidadeseinovacao/article/view/1077/1100>. Acesso em: 18 mai. 2024.

EBERHARDT, Ilva. F. N; COUTINHO, Carina V. S. **DIFÍCULDADES DE APRENDIZAGEM EM MATEMÁTICA NAS SÉRIES INICIAIS: DIAGNÓSTICO E INTERVENÇÕES.** Vivências:

Revista.reitoria.uri.br/~vivencias/Numero_013/artigos/artigos_vivencias_13/n13_08.pdf&ved=2ahUKEwjKsKbE9_WGAxXmgGEGHVdXDCcQFnoECB0QAQ&usg=AOvVaw2EZFW-jmBQD1NySXhKonkJ. Acesso em: 06 de jun. de 2024.

JÚNIOR, Edilson dos Passos Neri; VAZ, Cristina Lúcia Dias. **Guia de Impressão 3D.** EDITAEDI- ASSESSORIA DE EDUCAÇÃO A DIST NCIA- UFPA, [S. I.], v. 1, n. 1, p. 2, 2020. Disponível em:

https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://livroaberto.ufpa.br/jspui/bitstream/prefix/840/1/Folheto_GuialImpressao3D.pdf&ved=2ahUKEwjJ6LPi6PWGAxUxnpUCHStODIcQFnoECA4QAQ&usg=AOvVaw1WUtDxwlkNvO7qk7hudy0L. Acesso em: 17 de mai. 2024.

NUNES, Gabriel Novakovski. **As vantagens do Node.js.** REFAQI - REVISTA DE GESTÃO EDUCAÇÃO E E TECNOLOGIA, [S. I.], v. 4, n. 2, p. 2, 2018. Disponível em: <https://refaqi.faqi.edu.br/index.php/refaqi/article/view/96>. Acesso em: 7 maio. 2024.

ROSA, Cleiton; SANTOS, Michelle; SOUZA, Simone Sinara. **Aula 13: Push Button.** Revista Robótica, [S. I.], v. 1, p. 5, 2021. Disponível em:
https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://aluno.escoladigital.pr.gov.br/sites/alunos/arquivos_restritos/files/documento/2021-05/aula_13_push_button.pdf&ved=2ahUKEwj7xbOt7aOKAxW5ILkGHf5VBksQFnoECBsQAQ&usg=AOvVaw3bb1yQjRAMH27P7uGg0nEO. Acesso em: 18 mai. 2024.

SILVA, Luiz Carlos Freitas. **AS DIFICULDADES EM APRENDER E ENSINAR MATEMÁTICA.** 2009. Monografia (Licenciado em Matemática) – Universidade Estadual de Goiás, Jussara, 2009.

ABREU, Leandro. **Conheça o que é wireframe de site e aprenda como aplicá-lo na sua estratégia online.** [S. I.]. ROCKCONTENT, 9 jul. 2019. Disponível em: <https://rockcontent.com/br/blog/wireframe-de-site/>. Acesso em: 26 abril. 2024, às 20:48.

ADOBE. StyleSheet - **AS3.** [S.I.]. ADOBE, 2018. Disponível em: https://help.adobe.com/pt_BR/FlashPlatform/reference/actionscript/3/flash/text/StyleSheet.html. Acesso em: 15 mai. 2024, às 22:12.

AERO ENGENHARIA. **O que é : Componentes de Sistemas.** [S. I.]. AERO ENGENHARIA, 26 jul. 2023. Disponível em: <https://aeroengenharia.com/glossario/o-que-e-componentes-de-sistemas/>. Acesso em: 12 ago. 2024, às 10:07.

ANTUNES, Ana Maria. **Aplicativos, Softwares e Jogos de Celulares na Matemática.** [S. I.]. SÓ MATEMÁTICA, 2020. Disponível em: https://www.somatematica.com.br/artigos/jogos_celulares_matematica/p2.php. Acesso em: 05 jun. 2024, às 10:39.

ARAÚJO, Rogério. **Inserindo CSS em uma página HTML.** [S.I.]. GRAN, 31 de out. 2023. Disponível em: <https://blog.grancursosonline.com.br/inserindo-css-em-uma-pagina-html/>. Acesso em: 26 de mai. de 2024, às 22:25.

ARRUDA. **JavaScript é a linguagem mais popular no GitHub e TypeScript ultrapassa Java.** [S.I.]. TECMUNDO, 08 nov. 2024. Disponível em: <https://www.tecmundo.com.br/software/273582-javascript-linguagem-popular-github-typescript-ultrapassa-java.htm>. Acesso em: 29 de abril. de 2024, às 19:26.

AUTODESK. **SOFTWARE DE MODELAGEM 3D.** [S.I.]. AUTODESK, 2024. Disponível em: <https://www.autodesk.com.br/solutions/3d-modeling-software>. Acesso em: 17 de mai. de 2024, às 20:12.

AWARI. Guia completo sobre o Figma em português: dicas, tutoriais e recursos. [S. I.]. AWARI, 08 mai. 2023. Disponível em: <https://awari.com.br/guia-completo-sobre-o-figma-em-portugues-dicas-tutoriais-e-recursos/>. Acesso em: 27 abril. 2023, às 20:45.

BNCC. Base Nacional Comum Curricular. [S.I.]. MEC,SEB, 2017. Disponível em: <http://basenacionalcomum.mec.gov.br/abase/>. Acesso em: 06 de jun. 2024, às 11:47.

DEVMEDIA. O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML. [S.I.]. DEVMEDIA, 2011. Disponível em: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408#:~:text=UML%20%C3%A9%20um%20acr%C3%B4nimo%20para,orientados%20a%20objetos%20que%20desenvolvemos>. Acesso em: 29 de abril. de 2024, às 10:38.

DIRANI, Lucas Ruiz. Conhecendo os diagramas de classe da UML. [S. I.]. MEDIUM, 27 jul. 2022. Disponível em: <https://medium.com/fora-de-assunto/conhecendo-os-diagramas-de-classe-da-uml-6f237f53cb45>. Acesso em: 12 mai. 2024, às 18:54.

EBAC. O que é modelagem 3D?. [S.I.]. EBAC, 23 de set. de 2019. Disponível em: <https://ebaconline.com.br/blog/modelagem-3d-o-que-e-e-como-funciona>. Acesso em: 17 de mai. de 2024, às 20:37.

ERNESTO, Manuel. A jornada do Firebase - Cloud Firestore. [S.I.]. MEDIUM, 16 de jan. de 2019. Disponível em: <https://refaqi.faqi.edu.br/index.php/refaqi/article/view/114>. Acesso em: 05 de mai. de 2024, às 23:45.

FIREBASE. Make your app the best it can be with Firebase and generative AI. [S.I.]. FIREBASE. 2023. Disponível em: <https://firebase.google.com/?hl=pt>. Acesso em: 04 de mai. de 2024, às 17:35.

GOOGLE CLOUD. **Firestore.** [S.I.]. GOOGLE CLOUD. 2024. Disponível em: <https://cloud.google.com/firestore?hl=pt-BR>. Acesso em: 05 de ago. de 2024, às 19:59.

IBM, **Máquinas de Estado.** [S. I.]. IBM, 05 de mar. 2021. Disponível em: <https://www.ibm.com/docs/pt-br/dmrt/9.5?topic=diagrams-state-machines>. Acesso em: 15 ago. 2024, às 20:53.

LOPES, Michele. **O que é Figma e como usar?** [S. I.]. EBAC, 31 ago. 2022. Disponível em: <https://ebaconline.com.br/blog/o-que-e-figma-e-como-usar>. Acesso em: 27 mai. 2024, às 21:29.

LUCIDCHART. **O que é wireframe?** [S. I.]. LUCIDCHART, [S. d.]. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-wireframe>. Acesso em: 03 mai. 2024, às 20:34.

MDN WEB DOCS. **JavaScript.** [S.I.]. MDN WEB DOCS, 2023. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript>. Acesso em: 29 de abril. de 2024, às 12:06.

MIGUEL, Vicente Patrício. **Introdução ao Diagrama de Sequência.** [S. I.]. MEDIUM, 10 de nov. 2019. Disponível em: <https://medium.com/documenta%C3%A7ao-uml/introdu%C3%A7%C3%A3o-ao-diagrama-de-sequ%C3%A3ncia-1ea5e9563594>. Acesso em: 13 abril. 2024, às 21:31.

MIRO. **Wireframes.** [S. I.]. MIRO, [S. d.]. Disponível em: <https://miro.com/pt/wireframe/o-que-e-wireframe/>. Acesso em: 26 abril. 2024, às 21:20.

NODE.JS. **Sobre o Node.js.** [S. I.]. NODE.JS, [S. d.]. Disponível em: <https://nodejs.org/en/about>. Acesso em: 11 mai. de 2024, às 21:55.

NOLETO, Cairo. **UML: o que é, para que serve e quando usar essa linguagem de notação?**. [S.I.]. BETRYBE, 25 ago.2022. Disponível em:
<https://blog.betrybe.com/tecnologia/uml/>. Acesso em: 29 de abril. de 2024, às 19:56.

QEDU. **Aprendizado adequado.** [S. I.]. QEDU, 2021. Disponível em: <https://pt-br.react.dev/learn/describing-the-ui>. Acesso em: 29 abril. De 2024, às 14:44.

REACT. **Construindo a UI.** [S.I.]. REACT. 26 de abril. 2024. Disponível em:
<https://pt-br.react.dev/learn/describing-the-ui>. Acesso em: 01 de mai. de 2024, às 19:04.

SAGA. **SOFTWARE DE MODELAGEM 3D.** [S.I.]. SAGA, 2 de dez. de 2020.
Disponível em: <https://blog.saga.art.br/o-que-e-modelagem-3d/>. Acesso em: 17 de mai. de 2024, às 20:43.

SANTANA, Bruno. Eleve Seus Designs: **O Que é Figma e Como Usá-lo Corretamente!** . [S. I.]. HOSTINGER, 11 ago. 2023. Disponível em:
<https://www.hostinger.com.br/tutoriais/figma-o-que-e>. Acesso em: 27 abril. 2024, às 20:31 .

SOUZA,Ivan . **Saiba o que é NPM (Node Package Manager) e como instalar.** [S.I.]. ROCKCONTENT, 28 de set. de 2020. Disponível em:
<https://rockcontent.com/br/blog/npm/>. Acesso em: 18 mai. 2024, às 15:35.

TORELLA, Maria. **Arduino Ou ESP: Descubra A Melhor Opção!.** [S. I.]. LOBODAROBÓTICA, 2020. Disponível em: <https://lobodarobotica.com/blog/arduino-ou-esp-descubra-a-melhor-opcao/>. Acesso em: 15 mai. 2024, às 21:27.

USINAINFO. **Kit Push Button 12x12 com Capas Coloridas 25 Unidades + Case.** [S.I.]. USINAINFO, 2024. Disponível em:
<https://www.usinainfo.com.br/push-buttons/kit-push-button-12x12-com-capas-coloridas-25-unidades-case-5480.html>. Acesso em: 25 de mai. de 2024, às 15:57.

VENTURA, Plínio. **Entendendo o Diagrama de Atividades da UML.** [S.I.].

ATEOMOMENTO, 26 de out. 2016. Disponível em:

<https://www.ateomomento.com.br/uml-diagrama-de-atividades/>. Acesso em 13 de mai. de 2024, às 00:13.

VIANA, Daniel. **Firebase: descubra no que esta plataforma pode te ajudar.** [S.I.].

TREINAWEB, jan. de 2018. Disponível em:

<https://www.treinaweb.com.br/blog/firebase-descubra-no-que-esta-plataforma-pode-te-ajudar>. Acesso em: 05 de mai. de 2024, às 21:57.

VIANA, Theyse . **Mais da metade dos alunos do CE terminam Ensino**

Fundamental com aprendizado ‘crítico’ em Matemática. [S.I.]. DIÁRIO DO

NORDESTE, 2024. Disponível em:

<https://diariodonordeste.verdesmares.com.br/ceara/mais-da-metade-dos-alunos-do-ce-terminam-ensino-fundamental-com-aprendizado-critico-em-matematica-1.3516864>. Acesso em: 05 de mai. 2024, às 11:24.