

**UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E
ENGENHARIAS**

ANDERSON MINOZZO BEGOSSI

**PROTÓTIPO DE APLICATIVO PARA O AUXÍLIO DE CRIANÇAS
COM DISLEXIA**

CAXIAS DO SUL

2021

ANDERSON MINOZZO BEGOSSI

**PROTÓTIPO DE APLICATIVO PARA O AUXÍLIO DE CRIANÇAS
COM DISLEXIA**

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do título de Bacharel em
Ciência da Computação na Área do
Conhecimento de Ciências Exatas e
Engenharias da Universidade de Caxias
do Sul.

Orientador: Prof. Me. Alexandre
Erasmus Krohn Nascimento

CAXIAS DO SUL

2021

ANDERSON MINOZZO BEGOSSI

**PROTÓTIPO DE APLICATIVO PARA O AUXÍLIO DE CRIANÇAS
COM DISLEXIA**

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do título de Bacharel em
Ciência da Computação na Área do
Conhecimento de Ciências Exatas e
Engenharias da Universidade de Caxias
do Sul.

Aprovado em 22/06/2021

BANCA EXAMINADORA

Prof. Me. Alexandre Erasmo Krohn Nascimento
Universidade de Caxias do Sul - UCS

Prof^ª. Me^a. Gabriele Dani Bonetti
Universidade de Caxias do Sul - UCS

Prof^ª. Dr^a. Scheila de Ávila e Silva
Universidade de Caxias do Sul - UCS

AGRADECIMENTOS

Agradeço aos meus pais Ivandro Begossi e Denize Minozzo, e minha irmã, Amanda Minozzo Begossi, pelo incentivo a sempre seguir o caminho do conhecimento, e pelo total apoio durante essa etapa, assim como em todos os momentos da minha vida.

Aos meus familiares e amigos, por estarem ao meu lado ajudando a superar novos desafios constantemente, e por compreenderem a total entrega que o caminho da graduação exige. Um agradecimento especial à minha namorada, Larissa Kappes, por sempre estar ao meu lado nos momentos mais difíceis, e por ter sempre me motivado a seguir em frente perante a todos os obstáculos encontrados.

Ao meu orientador, Alexandre E. K. Nascimento, por acreditar no potencial deste trabalho, além de orientar e oferecer todo o suporte necessário até a conclusão.

Ao Felipe Ponce, do blog DislexClub, e, à equipe do Instituto ABCD, que foram sempre receptivos em todas as vezes que foi necessário entrar em contato, e estiveram sempre dispostos a colaborar na realização deste trabalho.

E por fim, o meu muito obrigado a todos que direta ou indiretamente fizeram parte da minha formação.

“Eu acredito que às vezes são as pessoas que ninguém espera nada que fazem as coisas que ninguém consegue imaginar.”

Alan Turing

RESUMO

A dislexia está presente em várias discussões na área da Educação, uma vez que as habilidades afetadas pelo transtorno são o alicerce de qualquer aprendizado. Com uma identificação precoce, e, a aplicação de um programa de intervenção adequado, as chances de minimizar as dificuldades são maiores. Nos últimos anos, é crescente o número de crianças com menos de 12 anos que possuem um celular próprio. Visto isso, o presente trabalho tem por objetivo apresentar um protótipo de aplicativo para dispositivos móveis, para o apoio à crianças com dislexia. Para tal, foi realizada uma revisão bibliográfica acerca do transtorno, assim como uma descrição das tecnologias necessárias para o desenvolvimento da ferramenta. Em seguida, foi elaborado um projeto de *software*. A ferramenta foi idealizada tendo como inspiração um programa de remediação que utiliza a técnica da nomeação automática rápida. O projeto da solução foi construído a partir das diretrizes do processo ICONIX. Foram criados artefatos, como, protótipos de tela, diagrama de casos de uso, diagramas de sequência e diagramas de robustez, que serviram como base documentativa para a etapa de implementação. Foi desenvolvida uma web API REST, capaz de servir ao aplicativo dados armazenados remotamente. A API foi desenvolvida com a tecnologia *Node.js*, e o aplicativo com o *framework React Native*. As tecnologias escolhidas se mostraram adequadas e o desenvolvimento pôde ser realizado no tempo previsto. Após a implementação, a API foi disponibilizada publicamente através de um serviço de infraestrutura em nuvem, e o aplicativo foi publicado na loja de aplicativos *Google Play Store*.

Palavras-chave: Dislexia. Aplicativo. Dispositivos móveis.

LISTA DE FIGURAS

Figura 1 – Áreas encefálicas envolvidas na leitura	19
Figura 2 – Ativação cerebral durante a leitura	19
Figura 3 – Diferença na leitura de um disléxico e de um não disléxico	21
Figura 4 – Ficha de atividade usada no estímulo da consciência fonológica	23
Figura 5 – Prancha de nomeação rápida de palavras	24
Figura 6 – Prancha do ProNAR-LE composta por figuras com estímulos dissilábicos	26
Figura 7 – Visão geral do processo ICONIX	33
Figura 8 – Comunicação da solução proposta	35
Figura 9 – Tipos de classes do padrão UML	36
Figura 10 – Modelo de domínio	38
Figura 11 – Diagrama de casos de uso	39
Figura 12 – Capturas de tela 1: Fazer Login	40
Figura 13 – Caso de Uso 1: Fazer Login	41
Figura 14 – Diagrama de robustez 1: Fazer Login	41
Figura 15 – Diagrama de sequência 1: Fazer Login	42
Figura 16 – Diagrama de classes 1: Fazer Login	42
Figura 17 – Captura de tela 2: Fazer Cadastro	43
Figura 18 – Caso de Uso 2: Fazer Cadastro	44
Figura 19 – Diagrama de robustez 2: Fazer Cadastro	44
Figura 20 – Diagrama de sequência 2: Fazer Cadastro	45
Figura 21 – Capturas de tela 3: Realizar NAR de Palavras	45
Figura 22 – Caso de Uso 3: Realizar NAR de Palavras	47
Figura 23 – Diagrama de robustez 3: Realizar NAR de Palavras	48
Figura 24 – Diagrama de sequência 3: Realizar NAR de Palavras, parte 1	49
Figura 25 – Diagrama de sequência 3: Realizar NAR de Palavras, parte 2	49
Figura 26 – Diagrama de sequência 3: Realizar NAR de Palavras, parte 3	50
Figura 27 – Diagrama de classes 3: Realizar NAR de Palavras	50
Figura 28 – Capturas de tela 4: Realizar NAR de Figuras	51
Figura 29 – Caso de Uso 4: Realizar NAR de Figuras	52
Figura 30 – Capturas de tela 5: Fazer Revisão	53
Figura 31 – Caso de Uso 5: Fazer Revisão	54
Figura 32 – Diagrama de robustez 5: Fazer Revisão	54
Figura 33 – Diagrama de sequência 5: Fazer Revisão	55
Figura 34 – Diagrama de classes 5: Fazer Revisão	55
Figura 35 – Captura de tela 6: Editar Perfil	56
Figura 36 – Caso de Uso 6: Editar Perfil	57

Figura 37 – Diagrama de robustez 6: Editar Perfil	58
Figura 38 – Diagrama de sequência 6: Editar Perfil	58
Figura 39 – Captura de tela 7: Alterar Avatar	59
Figura 40 – Caso de Uso 7: Alterar Avatar	60
Figura 41 – Diagrama de robustez 7: Alterar Avatar	60
Figura 42 – Diagrama de sequência 7: Alterar Avatar	61
Figura 43 – Diagrama de classes 7: Alterar Avatar	61
Figura 44 – Captura de tela 8: Visualizar Ranking	62
Figura 45 – Caso de Uso 8: Visualizar Ranking	63
Figura 46 – Diagrama de robustez 8: Visualizar Ranking	63
Figura 47 – Diagrama de sequência 8: Visualizar Ranking	64
Figura 48 – Captura de tela 9: Alterar Configurações	64
Figura 49 – Caso de Uso 9: Alterar Configurações	65
Figura 50 – Diagrama de robustez 9: Alterar Configurações	66
Figura 51 – Diagrama de sequência 9: Alterar Configurações	66
Figura 52 – Diagrama de classes 9: Alterar Configurações	67
Figura 53 – Caso de Uso 10: Excluir Conta	67
Figura 54 – Diagrama de robustez 10: Excluir Conta	68
Figura 55 – Diagrama de sequência 10: Excluir Conta	69
Figura 56 – Diagrama de classes	70
Figura 57 – Etapas de desenvolvimento dos casos de uso	72
Figura 58 – <i>Codecs</i> suportados pelo <i>Speech-to-Text</i>	74
Figura 59 – Captura de tela da política de privacidade	84
Figura 60 – Captura de tela do formulário de avaliação do aplicativo	85

LISTA DE QUADROS

Quadro 1 – Linguagens de programação e ambientes das plataformas <i>Android</i> e <i>iOS</i> . . .	28
Quadro 2 – Comparativo entre o desenvolvimento Nativo, <i>Web Apps</i> e Híbrido	30
Quadro 3 – Artefatos do processo ICONIX	35
Quadro 4 – Requisitos funcionais do projeto	37
Quadro 5 – Requisitos não funcionais do projeto	37
Quadro 6 – <i>Endpoints</i> disponibilizados na primeira etapa	72
Quadro 7 – <i>Endpoints</i> relacionados à criação e listagem de estímulos	73
Quadro 8 – <i>Endpoints</i> disponibilizados na terceira etapa	77
Quadro 9 – Especificações do servidor	77

LISTA DE ALGORITMOS

Algoritmo 1	Função do aplicativo que solicita o reconhecimento do áudio	74
Algoritmo 2	Trecho de código do <i>SpeechRecognitionProvider</i>	76

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
APK	Android Package
DSM	Manual Diagnóstico e Estatístico de Transtornos Mentais
IBGE	Instituto Brasileiro de Geografia e Estatística
NAR	Nomeação Automática Rápida
RAN	Rapid Automated Naming
REST	Representational State Transfer
RF	Requisitos Funcionais
RNF	Requisitos Não-Funcionais
SDK	Software Development Kit
UML	Unified Modeling Language
XML	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	14
1.2	ESTRUTURA DO TRABALHO	14
2	DISLEXIA	16
2.1	HISTÓRICO	16
2.2	TIPOS DE DISLEXIA	17
2.2.1	Dislexia fonológica	17
2.2.2	Dislexia de superfície	18
2.2.3	Dislexia da nomeação rápida	18
2.2.4	Dislexia de déficit duplo	18
2.3	FATORES NEUROLÓGICOS	18
2.4	CARACTERÍSTICAS	20
2.5	INTERVENÇÃO	21
2.5.1	Consciência fonológica	22
2.5.2	Nomeação Automática Rápida	23
3	TECNOLOGIAS DE DESENVOLVIMENTO	27
3.1	DISPOSITIVOS MÓVEIS	27
3.2	PLATAFORMAS MOBILE	28
3.2.1	Desenvolvimento Nativo	28
3.2.2	Web Apps	29
3.2.3	Desenvolvimento Híbrido	29
3.3	GAMIFICAÇÃO	31
3.4	RECONHECIMENTO E SÍNTESE DE FALA	31
4	METODOLOGIA	33
5	PROJETO	34
5.1	REQUISITOS	36
5.2	MODELO DE DOMÍNIO	37
5.3	DIAGRAMA DE CASOS DE USO	38
5.4	DETALHAMENTO DE CASOS DE USO	39
5.4.1	Caso de Uso 1 - Fazer Login	39
5.4.2	Caso de Uso 2 - Fazer Cadastro	43
5.4.3	Caso de Uso 3 - Realizar NAR de Palavras	45
5.4.4	Caso de Uso 4 - Realizar NAR de Figuras	50

5.4.5	Caso de Uso 5 - Fazer Revisão	52
5.4.6	Caso de Uso 6 - Editar Perfil	55
5.4.7	Caso de Uso 7 - Alterar Avatar	58
5.4.8	Caso de Uso 8 - Visualizar Ranking	61
5.4.9	Caso de Uso 9 - Alterar Configurações	64
5.4.10	Caso de Uso 10 - Excluir Conta	67
5.5	DIAGRAMA DE CLASSES	70
6	IMPLEMENTAÇÃO	71
6.1	CONFIGURAÇÃO DO AMBIENTE DE DESENVOLVIMENTO	71
6.2	DESENVOLVIMENTO DOS CASOS DE USO	71
6.3	PUBLICAÇÃO	77
7	CONSIDERAÇÕES FINAIS	79
	REFERÊNCIAS	80
	ANEXO A – POLÍTICA DE PRIVACIDADE	84
	ANEXO B – FORMULÁRIO DE AVALIAÇÃO	85

1 INTRODUÇÃO

Nos primeiros anos escolares, muitas crianças se deparam com obstáculos para aprender ou se desenvolver em tarefas como a leitura e a escrita. Nos casos onde a criança não possui um déficit primário, o causador dessas dificuldades pode ser um transtorno de aprendizagem. A dislexia é um transtorno específico de aprendizagem, caracterizado pela dificuldade no reconhecimento preciso e/ou fluente de palavras, e pela baixa habilidade de decodificação e soletração (ROTTA; OHLWEILER; RIESGO, 2015). O transtorno atinge entre 5% e 17% da população mundial (IDA, 2012). Embora o número seja expressivo, a desinformação prevalece, colaborando com que o dislético sofra emocionalmente com preconceitos e julgamentos precipitados (IDA, 2012).

A dislexia não desaparece com o tempo, contudo, as pessoas disléticas são capazes de desenvolverem diferentes formas de superar suas dificuldades. Devido à sua complexidade, o diagnóstico pode levar anos para ser feito adequadamente. Segundo a Associação Internacional de Dislexia, o tratamento mais eficaz é a detecção precoce. Estudos indicam, que, com a aplicação de um programa de intervenção adequado, as chances de minimizar as dificuldades são maiores, e, problemas emocionais relacionados são mais facilmente evitados (IDA, 2012).

As áreas de intervenção variam de acordo com as características do perfil do indivíduo e faixa etária. Se tratando de aspectos como a fluência da leitura, uma das principais habilidades relacionadas é a Nomeação Automática Rápida (NAR) (DENCKLA; RUDEL, 1976). Essa habilidade está intimamente ligada com a velocidade de acesso à memória de curto prazo, e, à nomeação fonológica, as quais têm influência direta no desenvolvimento da leitura e escrita (JUNIOR *et al.*, 2019). Crianças com dislexia tendem a levar mais tempo nessa tarefa, quando comparadas com crianças que não apresentam o transtorno (CAPELLINI; CONRADO, 2009).

As pesquisadoras Santos e Capellini (2020), elaboraram um programa de remediação de velocidade de acesso ao léxico, utilizando a estratégia da nomeação automática rápida. Os resultados do estudo mostraram ter havido significância clínica nas provas de identificação, quando comparada a pré com a pós-testagem. Desta forma, concluiu-se que o programa pode ser utilizado como ferramenta de intervenção baseado em evidências científicas (SANTOS; CAPELLINI, 2020).

Dentro desse contexto, no presente trabalho foi desenvolvido um protótipo de aplicativo para dispositivos móveis, com a finalidade de auxiliar crianças com dislexia a aprimorarem suas habilidades de leitura, tendo como inspiração o programa de remediação criado por Santos e Capellini (2020). Também foram incluídos elementos de gamificação, a fim de promover incentivos emocionais no uso da ferramenta. A natureza do aplicativo também demandou o desenvolvimento de uma web API, que é responsável por armazenar e fornecer os dados neces-

sários para o funcionamento do aplicativo.

1.1 OBJETIVOS

Desenvolver um protótipo de aplicativo para dispositivos móveis, para o auxílio de crianças com dislexia. Com base no objetivo geral, foram elaborados os seguintes objetivos específicos:

1. Identificar diferentes abordagens de remediação da dislexia.
2. Categorizar as diferentes abordagens de desenvolvimento de aplicativos para dispositivos móveis.
3. Identificar os principais serviços de reconhecimento e síntese de voz.
4. Elaborar um projeto de *software*, compreendido por visões de modelagem e arquitetura.
5. Desenvolver um aplicativo capaz de incentivar crianças com dislexia à treinarem suas habilidades de leitura, baseando-se em um programa de remediação existente.
6. Desenvolver uma web API capaz de servir ao aplicativo todos os dados necessários para o funcionamento do mesmo.

1.2 ESTRUTURA DO TRABALHO

O presente trabalho está organizado da seguinte forma:

- No Capítulo 2 é apresentada uma visão geral da dislexia. A visão geral é constituída pela definição, histórico, tipos, características e intervenção. Entre as diferentes intervenções, destaca-se a da nomeação automática rápida.
- No Capítulo 3 são descritos os principais conceitos e tecnologias relacionadas ao desenvolvimento da aplicação. Inicialmente, são abordados os aspectos do uso de dispositivos móveis no apoio a escolares com dificuldades de aprendizagem. Em seguida, são descritas as principais plataformas de desenvolvimento para dispositivos móveis. Por fim, é apresentada uma breve fundamentação sobre a gamificação, e são descritos os principais serviços de reconhecimento e síntese de voz.
- No Capítulo 4 é descrita a metodologia de desenvolvimento da ferramenta. É feita uma apresentação do processo de desenvolvimento ICONIX, e das etapas e artefatos que a metodologia engloba.

- No Capítulo 5 é apresentado o projeto de *software* da solução proposta. Inicialmente, é contemplada a etapa de análise de requisitos. Em seguida, as funcionalidades do *software* são detalhadas. Por fim, são apresentados os artefatos que ilustram as interações do usuário com a aplicação.
- No Capítulo 6 são descritas as principais etapas do desenvolvimento da aplicação. Inicialmente, a configuração do ambiente de desenvolvimento é detalhada. Em seguida, é descrita a implementação das funcionalidades. Por fim, é descrito o processo de publicação.
- No Capítulo 7 são apresentadas as considerações finais do trabalho.

2 DISLEXIA

A palavra "dislexia" é constituída pelos radicais "dis", que significa distúrbio, e, "lexia", que significa leitura, no latim. Ou seja, o termo dislexia refere-se à dificuldades na leitura (LERNER, 1989). Segundo o pesquisador G. Reid Lyon (1995), dislexia é um distúrbio específico de origem constitucional caracterizado por uma dificuldade na decodificação de palavras simples que, como regra, mostra uma insuficiência no processamento fonológico. A Associação Internacional de Dislexia (2012) segue a mesma linha de Lyon:

A dislexia é um transtorno específico de aprendizagem de origem neurobiológica, caracterizada por dificuldade no reconhecimento preciso e/ou fluente das palavras e pela baixa habilidade de decodificação e soletração. Essas dificuldades normalmente derivam de um déficit no componente fonológico da linguagem, muitas vezes surpreendente, quando comparado a outras habilidades cognitivas e ao acesso à aprendizagem. Também são apresentadas consequências secundárias, que podem incluir dificuldades na compreensão de texto e experiência de leitura reduzida, podendo impedir o desenvolvimento do vocabulário e do conhecimento geral. (IDA, 2012, tradução nossa).

De acordo com a Associação Internacional de Dislexia (2012), a dislexia está presente entre 5% e 17% da população mundial. Embora o número seja expressivo, o disléxico ainda sofre constantemente com os problemas emocionais causados pelo distúrbio, que, no ambiente escolar, fica mais evidente. Diante da inabilidade de leitura, o disléxico pode desenvolver um sentimento de fobia e rejeição ao ambiente escolar. Braggio (2012), aponta que muitas crianças disléxicas acabam abandonando a escola, por terem que conviver e lidar com os modelos de ensino antiquados, exigências burocráticas, humilhações sofridas, e, principalmente, as notas.

É possível observar que a dislexia está presente em muitas discussões na área da Educação. Ler e escrever são competências básicas para o ser humano. Desta forma, estas habilidades tornam-se o alicerce de qualquer aprendizado. A seguir, são descritos os principais aspectos acerca do transtorno.

2.1 HISTÓRICO

Em 1887, o oftalmologista alemão, Rudolf Berlin, utilizou pela primeira vez o termo dislexia, para caracterizar pacientes que observou durante alguns anos com dificuldades de leitura e escrita. Em 1896, o médico britânico, W. Pringle Morgan, publicou no *British Medical Journal* (BMJ) o caso de um jovem de 14 anos, que, apesar da boa avaliação cognitiva, tinha incapacidade para ler. Morgan denominou esse caso peculiar de "cegueira verbal". Em 1907, Stevenson, relata seis casos de "cegueira verbal" em uma família, já dando indícios do aspecto genético (ROTTA; OHLWEILER; RIESGO, 2015).

Em 1917, a expressão dislexia ressurgiu, com o oftalmologista James Hinshelwood. Hinshelwood encontrou uma paciente, que, apesar da escolaridade avançada, tinha dificuldade para ler e escrever. Posteriormente, dedicou anos de estudo à cegueira verbal, e, concluiu que a causa mais provável desse distúrbio de leitura era um defeito congênito no cérebro, afetando a memória visual de palavras e letras. Todavia, as observações subsequentes feitas por oftalmologistas concluíram que a dificuldade não estava nos olhos, mas sim no funcionamento de áreas de linguagem no cérebro (ROTTA; OHLWEILER; RIESGO, 2015).

No ano de 1928, após se dedicar ao estudo de transtornos de aprendizagem, o neuropsiquiatra, Samuel T. Orton, publicou um trabalho descrevendo as dificuldades perceptivo-linguísticas específicas em crianças com pouca habilidade de leitura. Somente em 1950, Hallgren publicou o primeiro estudo clínico e genético que chamou de "dislexia específica", substituindo a expressão "cegueira verbal" (ROTTA; OHLWEILER; RIESGO, 2015).

O desenvolvimento de tecnologias de neuroimagem, nas décadas de 1980 e 1990, possibilitou que pesquisas sobre a dislexia tivessem grandes avanços (FIEZ; PETERSEN, 1998). Em 2001, Temple et al. (2001) sugeriram que a dislexia poderia ser caracterizada por interrupções neurais de processos fonológicos e ortográficos importantes para a leitura. Em 2003, Ziegler e colegas, publicaram uma pesquisa apoiando a ideia de que a origem da dislexia é principalmente biológica (ZIEGLER *et al.*, 2003). De 2003 até os dias atuais, os modelos de relação entre o cérebro e a dislexia geralmente focam em alguma forma de amadurecimento cerebral defeituoso ou atrasado (COLLINS; ROURKE, 2003).

2.2 TIPOS DE DISLEXIA

Atualmente, ainda nenhuma instituição criou uma lista oficial de tipos de dislexia. Contudo, especialistas vêm pesquisando sobre possíveis formas de subdividir o transtorno. Acredita-se que a genética desempenhe um dos papéis mais importantes nas diferenciações dos casos. Diferentes dificuldades de leitura podem estar ligadas a combinações específicas de genes, e também às diferentes experiências de aprendizagem de uma criança. Esses fatores colaboram para explicar o motivo de não existirem duas pessoas com dislexia exatamente iguais (UNDERSTOOD, 2020).

Os especialistas divergem sobre como a dislexia deve ser dividida em tipos ou subtipos. Todavia, na literatura é possível encontrar algumas definições de tipos mais comumente mencionadas, sendo elas, descritas a seguir.

2.2.1 Dislexia fonológica

Os especialistas acreditam que esse é o tipo mais comum de dislexia. Os principais sinais são dificuldades em compreender e separar os sons da linguagem, e combiná-los com

símbolos escritos. O processo de pronúncia e decodificação de palavras é o mais afetado (UNDERSTOOD, 2020).

2.2.2 Dislexia de superfície

Quando a pessoa não tem problemas com a pronúncia de palavras, mas, tem dificuldades em reconhecer uma palavra comum no primeiro momento em que a vê, é dito que o indivíduo possui uma dislexia de superfície. Essa dificuldade ocorre pois o cérebro possui uma dificuldade em memorizar a aparência das palavras (UNDERSTOOD, 2020).

Muitas pessoas podem ter a dislexia fonológica combinada com a dislexia de superfície, pois, problemas com a decodificação podem atrapalhar o reconhecimento visual das palavras (UNDERSTOOD, 2020).

2.2.3 Dislexia da nomeação rápida

Nesse tipo de dislexia, as pessoas têm dificuldades em nomear rapidamente símbolos, como, letras, números e cores, quando os vêem. Esse problema está ligado à velocidade com que o cérebro acessa a representação fonológica do símbolo. Isso pode ocasionar uma leitura mais pausada, e, lenta (UNDERSTOOD, 2020).

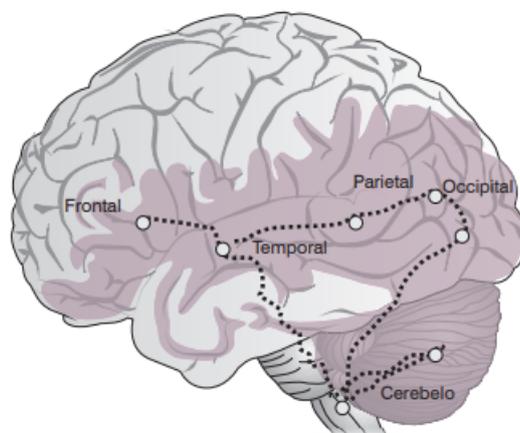
2.2.4 Dislexia de déficit duplo

As pessoas com dislexia de déficit duplo enfrentam desafios em dois aspectos da leitura. Muitas vezes, esse tipo é usado para agrupar pessoas com dificuldades em identificar os sons das palavras, e, dificuldades com a velocidade de nomeação. Esse tipo de dislexia é considerado um dos mais graves (UNDERSTOOD, 2020).

2.3 FATORES NEUROLÓGICOS

Técnicas de neuroimagem, como, a imagem por ressonância magnética funcional e a tomografia por emissão de pósitrons, possibilitaram que os estudos neurológicos sobre a dislexia tivessem um grande avanço (WHITAKER, 2010). Com essas técnicas também foi possível a identificação do padrão de ativação no cérebro do leitor. Os estudos apontam consistentemente para a ativação da região temporal esquerda do cérebro durante a execução de tarefas de linguagem. Além disso, as regiões parietais inferiores esquerdas estão implicadas no processamento fonológico normal e recuperação de palavras, e, os circuitos do cerebelo-frontal, estão envolvidos com a nomeação automática rápida (ROTTA; OHLWEILER; RIESGO, 2015). Na Figura 1 estão representadas as áreas encefálicas envolvidas na leitura.

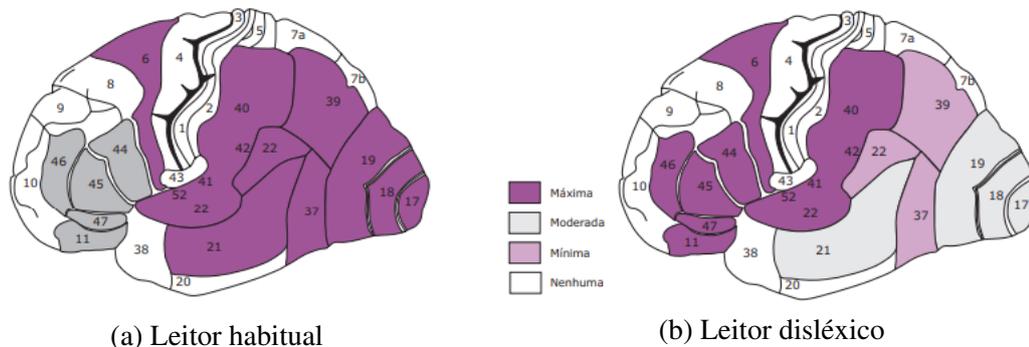
Figura 1 – Áreas encefálicas envolvidas na leitura



Fonte: ROTTA; OHLWEILER; RIESGO (2015).

A partir de tarefas de leitura, verificou-se que há uma diferença na ativação cerebral de pessoas com dislexia e pessoas sem dislexia. A Figura 2 mostra essa diferença. Na Figura 2a é mostrada a ativação cerebral de um leitor habitual, e na Figura 2b de um leitor disléxico.

Figura 2 – Ativação cerebral durante a leitura



Fonte: ROTTA; OHLWEILER; RIESGO (2015).

Para compensar essa dificuldade, regiões envolvidas na produção de sons da fala são mais ativadas em comparação com não disléxicos, assim como a região inferior-frontal, e as áreas do hemisfério direito, que fornecem pistas visuais (RICHLAN, 2012). O leitor disléxico pode tentar pronunciar uma palavra desconhecida usando somente a informação visual, sem vincular as letras aos sons. Isso explica por que alguns disléxicos parecem estar tentando adivinhar as palavras quando lêem. Acredita-se que ocorre uma adaptação cerebral, devido às dificuldades no sistema de decodificação (KEARNS *et al.*, 2019).

2.4 CARACTERÍSTICAS

O Manual Diagnóstico e Estatístico de Transtornos Mentais (DSM) (APA, 2014) propõe que as características centrais da dislexia são observadas quanto à precisão na leitura de palavras, a velocidade ou fluência de leitura, e a compreensão leitora.

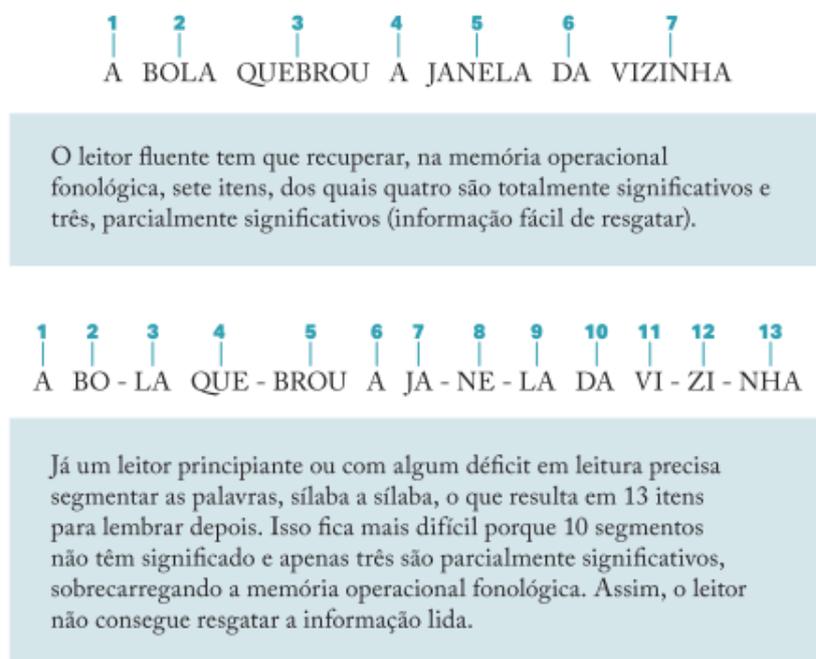
Uma das principais características do dislético é a dificuldade na decodificação e no reconhecimento de palavras. A decodificação é o processo no qual o leitor transforma as retas e curvas que compõem as letras em sons oralizados, ou, numa imagem mental. Por si só, o processo de decodificação é complexo, devido à relação não biunívoca entre sons e letras, onde cada letra não é sempre pronunciada da mesma forma. Por exemplo, no caso da palavra “banana”, temos três ocorrências da letra “A”, mas, em cada uma delas essa letra representa um som diferente. O déficit para decodificar e reconhecer palavras é uma característica central da dislexia, pois, acarreta em muitas outras dificuldades (MOUSINHO *et al.*, 2020).

Devido à dificuldade de decodificação, e, a imprecisão no acesso lexical, surgem problemas na precisão de leitura. Segundo Mousinho *et al.* (2020), é comum observar:

1. Substituições: palavra CAVALO lida como CAFALO;
2. Transposições: palavra PRAIA lida como PARIA;
3. Aproximações lexicais: palavra BARRIL lida como BRASIL;
4. Regularizações: palavra DUREX lida com o som típico de X e CH.

Outra característica presente nos disléticos é a leitura lenta. Geralmente, o indivíduo com o transtorno lê de forma segmentada, e, com maior número de pausas. Isso ocorre pois a leitura não é feita de forma automatizada, conforme mostra a Figura 3.

Figura 3 – Diferença na leitura de um disléxico e de um não disléxico



Fonte: MOUSINHO *et al.* (2020).

Como consequência dos comprometimentos principais que são causados pelo transtorno, o sujeito disléxico normalmente também apresenta dificuldade para compreender o que lê. Isso ocorre pois a falta de automatismo acaba causando uma sobrecarga de memória operacional, impossibilitando que uma compreensão do que é lido seja feita. Além dos déficits fonológicos, os indivíduos com dislexia também apresentam dificuldades de percepção visual. Os problemas referentes aos movimentos oculares podem ser indicadores de dificuldades de compreensão, e, por esse motivo, se observa a existência de frequentes movimentos de regressão e de inúmeras e longas fixações para ler uma única palavra (PRADO; DUBOIS; VALDOIS, 2007).

2.5 INTERVENÇÃO

A dislexia é um transtorno crônico, ou seja, não desaparece com o tempo. Porém, uma pessoa com dislexia é capaz de aprender novas formas de se expressar à medida que evolui. O tratamento mais eficaz para a dislexia é a detecção precoce. Com a aplicação de um programa de intervenção adequado, as chances de minimizar as dificuldades no desenvolvimento são maiores, e problemas emocionais relacionados são mais facilmente evitados.

Segundo a Associação Internacional de Dislexia (2012), é importante que a abordagem de tratamento para disléxicos seja multissensorial e baseada na estruturação da linguagem. Também é importante, que, durante o apoio, haja o estímulo de vários sentidos ao mesmo tempo,

como, ver, ouvir e tocar. Disléxicos geralmente precisam de muita prática e *feedback* corretivo imediato para desenvolver habilidades de reconhecimento automático de palavras.

As áreas de intervenção variam de acordo com as características do perfil do indivíduo e faixa etária. Em idades mais precoces, são recomendadas estratégias de aprimoramento da consciência fonológica, e, em idades mais avançadas, é mais relevante intervir em aspectos como a compreensão e fluência (CADIN, 2020). Também salienta-se a importância dos indivíduos receberem estímulos por todas as suas conquistas na aprendizagem. Desta forma, há uma busca pelo progresso e aumento da confiança. A seguir, são apresentadas duas áreas de possível intervenção.

2.5.1 Consciência fonológica

Se tratando de estimular o desenvolvimento fonológico em crianças com dislexia, vários métodos são encontrados na literatura. Adams et al. (2018), propõe que o estímulo da consciência fonológica deve ser feito através dos seguintes pontos:

1. Estímulo da percepção auditiva, atenção e concentração;
2. Uso de rimas para introduzir os sons das palavras;
3. Desenvolvimento da consciência de que a fala é composta por uma sequência de palavras;
4. Desenvolvimento da capacidade de analisar as palavras em sílabas;
5. Desenvolvimento da consciência de que as palavras contém fonemas;
6. Introdução da relação entre grafema/fonema;
7. Introdução gradativa das letras e da escrita.

O Instituto ABCD¹, instituto referência em dislexia no Brasil, propõe algumas fichas de atividades de ajuda no processo de alfabetização, que podem ser usadas no estímulo da consciência fonológica. A Figura 4 é um exemplo de atividade proposta, onde o aluno deve ouvir a sílaba final falada pelo instrutor, e assinalar a figura correspondente na ficha.

¹ <<https://www.institutoabcd.org.br/>>

Figura 4 – Ficha de atividade usada no estímulo da consciência fonológica



Fonte: ABCD (2020).

2.5.2 Nomeação Automática Rápida

Quanto aos aspectos de fluência da leitura, uma das habilidades de maior importância é a velocidade com que se realiza a nomeação de um estímulo visual linguístico. A velocidade de nomeação encontra-se intimamente relacionada à velocidade de acesso à memória de curto prazo, e, à nomeação fonológica, as quais influenciam diretamente no desenvolvimento da leitura e da escrita (JUNIOR *et al.*, 2019). Para avaliar a velocidade de nomeação, pesquisadores criaram um teste chamado *Rapid Automatized Naming (RAN)* (DENCKLA; RUDEL, 1976). Originalmente, o teste consiste na apresentação de pranchas (tradicionalmente cartões ou folhas de papel) contendo estímulos visuais que devem ser nomeados pelos disléxicos o mais rápido possível. Esses estímulos podem ser letras, palavras, números, figuras ou cores. A tarefa é então cronometrada, permitindo avaliar o tempo que a criança leva para olhar os estímulos e acessar sua representação fonológica e significado na memória (DENCKLA; RUDEL, 1976).

Pesquisas apontam que crianças com dislexia tendem a levar mais tempo na realização do teste de nomeação automática rápida, comparado com crianças que não apresentam alteração

na leitura (CAPELLINI; CONRADO, 2009). Um estudo internacional desenvolveu um programa de remediação para escolares com dificuldades de leitura através de um treinamento da consciência fonológica e do ensino explícito das regras de correspondência grafo-fonêmicas. O programa utilizou a nomeação automática rápida como uma das medidas de pré e pós-intervenção. Ao final do estudo, é indicado que a associação entre as habilidades metafonológicas favorece a realização do teste de nomeação rápida, melhorando assim, a velocidade de acesso ao léxico e a fluência de leitura (WOLFF, 2014).

Apesar do uso da nomeação automática rápida como estratégia de intervenção ainda ser escasso na literatura nacional e internacional, Santos e Capellini (SANTOS; CAPELLINI, 2020) elaboraram um programa de remediação de velocidade de acesso ao léxico, utilizando essa estratégia. O programa que foi elaborado tem como base o teste original, citado anteriormente, e é constituído por pranchas com estímulos linguísticos (letras, palavras reais e pseudopalavras) e não linguísticos (cores e figuras). Para a criação do programa, as pesquisadoras realizaram uma seleção de estímulos a partir de um banco de palavras. Dentre 600 palavras obtidas, selecionaram apenas os substantivos, pois podem ser representados por figuras. Em seguida, foram selecionadas apenas as palavras em que o padrão silábico (CV, CCV) estava presente com alta frequência. Na elaboração das pranchas, cinco estímulos foram distribuídos em 10 fileiras, de modo que cada estímulo se repita 10 vezes, como pode ser observado na Figura 5.

Figura 5 – Prancha de nomeação rápida de palavras

bolsa	cinto	filme	metro	tempo
cinto	filme	metro	tempo	bolsa
filme	bolsa	tempo	metro	cinto
tempo	cinto	filme	bolsa	metro
metro	filme	tempo	cinto	bolsa
bolsa	metro	cinto	filme	tempo
tempo	cinto	bolsa	metro	filme
metro	bolsa	cinto	filme	tempo
tempo	cinto	bolsa	metro	filme
tempo	bolsa	metro	filme	cinto

Fonte: SANTOS; CAPELLINI (2018).

No método de aplicação proposto pelas pesquisadoras são realizadas 6 sessões, de modo que o grau de complexidade dos estímulos aumente a cada 2 sessões. Cada sessão deve ser realizada pelo aplicador seguindo os seguintes passos:

1. Primeiramente, a criança deverá realizar o reconhecimento dos estímulos contidos em cada prancha da sessão. Caso houver algum estímulo desconhecido, o aplicador poderá explicar ao indivíduo antes do início da primeira tomada de tempo;
2. Em seguida, a criança é orientada a iniciar o procedimento de nomeação rápida dos estímulos. O tempo que a mesma leva para fazer a nomeação é anotado e chamado de T1;
3. Após a primeira tomada de tempo, é realizada uma intervenção, para que haja a identificação dos estímulos que não foram reconhecidos em um primeiro momento;
4. Por fim, a criança é orientada a realizar novamente a nomeação rápida dos estímulos, e o tempo novamente anotado, chamado de T2.

Os resultados do estudo de Santos e Capellini mostraram ter havido significância clínica nas provas de identificação, quando comparada a pré com a pós-testagem. Desta forma, as pesquisadoras concluíram que o programa se mostrou eficaz e com aplicabilidade, podendo ser utilizado como instrumento de intervenção baseado em evidências científicas para escolares com dislexia (SANTOS; CAPELLINI, 2020).

O estudo referenciado deu origem ao ProNAR-LE (SANTOS; CAPELLINI, 2018), um programa comercializado, utilizado por fonoaudiólogos e outros profissionais da área da Educação. O programa tem como objetivo o auxílio a escolares com problemas de leitura na melhora da decodificação, precisão, velocidade e fluência de leitura, utilizando a estratégia da nomeação automática rápida. O ProNAR-LE é composto por dois cadernos, sendo um de pranchas para avaliação e um de pranchas para a intervenção. O método de aplicação segue os mesmos passos do estudo base, descrito anteriormente. O caderno de pranchas de intervenção é composto por 22 pranchas, que são distribuídas por sessões de acordo com o grau de complexidade. As 22 pranchas são descritas a seguir:

- 1 prancha de figuras com estímulos monossilábicos;
- 3 pranchas de figuras com estímulos dissilábicos;
- 2 pranchas de figuras com estímulos trissilábicos;
- 1 prancha de figuras com estímulos polissilábicos de quatro sílabas;
- 4 pranchas de pseudopalavras (monossilábicos, dissilábicos, trissilábicos e polissilábicos com quatro sílabas);
- 3 pranchas de palavras reais monossilábicas;
- 3 pranchas de palavras reais dissilábicas;
- 2 pranchas de palavras reais trissilábicas;

- 1 prancha de palavras reais polissilábicas com quatro sílabas;
- 2 pranchas de cores, sendo uma prancha com estímulos dissilábicos e uma prancha contendo estímulos dissilábicos, trissilábicos e polissilábicos com quatro sílabas;

Na Figura 6 observa-se um exemplo de prancha do ProNAR-LE, composta por figuras que representam palavras de apenas duas sílabas.

Figura 6 – Prancha do ProNAR-LE composta por figuras com estímulos dissilábicos



Fonte: SANTOS; CAPELLINI (2018).

3 TECNOLOGIAS DE DESENVOLVIMENTO

Este capítulo apresentará uma descrição das tecnologias relacionadas ao desenvolvimento do projeto. Na Seção 3.1 são apresentados dados e estudos acerca do uso de dispositivos móveis. Na Seção 3.2 são apresentadas as principais plataformas *mobile* da atualidade. Na Seção 3.3 são abordados aspectos envolvendo o uso da gamificação no incentivo de estudantes com dificuldades de aprendizagem. Por fim, na Seção 3.4 são apresentados os principais serviços de reconhecimento e síntese de voz.

3.1 DISPOSITIVOS MÓVEIS

A Pesquisa Nacional por Amostra de Domicílios Contínua sobre Tecnologia da Comunicação e da Informação (PNAD Contínua TIC 2018), divulgada pelo Instituto Brasileiro de Geografia e Estatística (IBGE), mostra que 79,3% dos brasileiros com 10 anos ou mais têm aparelhos celulares para o uso pessoal, com ou sem internet (IBGE, 2020). Em outra pesquisa, realizada pelo site de notícias *Mobile Time* e a empresa de soluções de pesquisas *Opinion Box*, foi constatado que, no Brasil, 74% das crianças entre 10 e 12 anos possuem um aparelho celular próprio (TIME, 2019). Com a popularização dos celulares, cresce também o uso dos aplicativos móveis. De acordo com um dos principais relatórios sobre o tema no mundo, da consultoria *App Annie*, o *download* anual de aplicativos no Brasil cresceu 40% nos últimos três anos, atingindo cerca de 5 bilhões em 2019 (BRASIL, 2020).

O avanço dessas tecnologias proporciona o surgimento de novas oportunidades de uso, e, dentre elas, o emprego de aplicativos móveis em processos de ensino e aprendizagem (DAUD; ABAS, 2013). As tecnologias digitais podem facilitar a aprendizagem de alunos, aumentando sua motivação, promovendo a auto-competição, aumentando sua confiança e autoestima (THOMAS apud SKIADA *et al.*, 2014). O uso de tecnologias digitais como ferramentas de apoio a estudantes com dificuldades pode permitir o desenvolvimento conforme o tempo e necessidades necessárias (SKIADA *et al.*, 2014). Estudantes com dificuldades na leitura e escrita, como é o caso da dislexia, podem tirar proveito desses benefícios.

Segundo SKIADA *et al.* (2014), os aplicativos móveis incentivam as crianças com dislexia a realizarem um maior número de exercícios, assim como tirar proveito dos estímulos audiovisuais na melhoria de suas habilidades. Os pesquisadores Saleh e Alias (2012 apud SKIADA *et al.*, 2014) desenvolveram um aplicativo móvel de aprendizagem inspirado em revistas em quadrinhos, para crianças com dislexia. O estudo evidenciou o potencial uso das ferramentas de apoio digitais em tal contexto. Em outro estudo, Rello, Bayari e Gorriz (2012 apud SKIADA *et al.*, 2014) desenvolveram um aplicativo chamado *Dyseggia*. O aplicativo é constituído por exercícios e jogos com palavras, e, recebeu um *feedback* positivo das crianças disléxicas que

utilizaram a ferramenta.

3.2 PLATAFORMAS MOBILE

Atualmente, a indústria de aplicativos para dispositivos móveis é dominada por dois sistemas operacionais: *Android* e *iOS* (AXELSSON; CARLSTRÖM, 2016). Embora o *Android* tenha a maior participação de mercado, com mais de 70% (STATCOUNTER, 2020), é essencial para empresas e desenvolvedores que seu produto seja entregue para ambas as plataformas. Cada plataforma tem sua própria maneira de desenvolver aplicações. Deste modo, um problema comum que a indústria enfrenta é a necessidade de desenvolver o mesmo produto em dois aplicativos separados, que, por sua vez, necessitam de manutenção paralela e um maior gasto na contratação de desenvolvedores especializados. Uma solução para esse problema é o desenvolvimento de uma aplicação híbrida, onde o mesmo código é reaproveitado na geração do aplicativo para as diferentes plataformas. Outra alternativa é o desenvolvimento de *web apps*. A seguir, são descritas e comparadas cada uma das três diferentes categorias de desenvolvimento de aplicativos móveis.

3.2.1 Desenvolvimento Nativo

Uma aplicação nativa é desenvolvida especificamente para uma determinada plataforma, utilizando suas próprias ferramentas de desenvolvimento e linguagens de programação suportadas (MONTAN; SANTOS, 2018). O Quadro 1 mostra as linguagens suportadas e o ambiente de desenvolvimento para cada plataforma.

Quadro 1 – Linguagens de programação e ambientes das plataformas *Android* e *iOS*

Plataforma	Linguagem	Ambiente de desenvolvimento
Android	Java / Kotlin	Android Studio
iOS	Objective-C / Swift	XCode

Fonte: MONTAN; SANTOS (2018).

O Google, empresa por trás do sistema operacional *Android*, utilizou o *Java* como a principal linguagem de programação para o desenvolvimento de seus aplicativos. O *Android Software Development Kit (SDK)*, contém todos os pacotes e bibliotecas que o desenvolvedor necessita para criar um aplicativo. Quando o projeto *Android* é criado, o desenvolvedor pode compilar o código *Java* usando o *Android SDK*. Esse processo irá resultar em um *Android Package (APK)*, um pacote compactado da composição do código do aplicativo, que pode ser instalado e executado em um dispositivo real ou virtual. Posteriormente, com o intuito de tornar o desenvolvimento mais rápido e seguro, o Google colaborou com a criação da linguagem *Kotlin*, que também pode ser utilizada no desenvolvimento de aplicativos nativos para *Android*. A implementação da interface de usuário no *Android* utiliza componentes e conceitos da imple-

mentação de interfaces do *Java*. A estilização dos componentes da interface é feita através de um arquivo *Extensible Markup Language (XML)* (MEIER, 2012).

Do outro lado, a Apple utilizou o *Objective-C* durante vários anos como a principal linguagem de programação para suas aplicações. Em 2014, houve uma mudança gradual para a linguagem *Swift*, por conta de ser uma linguagem mais recente, e demandar um tempo menor de implementação. O *Swift* emprega conceitos modernos de programação, oferecendo uma sintaxe mais simples. Conceitos do *Objective-C*, como, protocolos e categorias, são encontrados no *Swift*, porém, podem ser aplicados em outras estruturas da linguagem. O compilador do *Swift* é distribuído junto ao *Xcode*, ambiente de desenvolvimento da Apple (REBOUÇAS *et al.*, 2016).

3.2.2 Web Apps

As *web apps* são aplicações desenvolvidas com linguagens de programação e marcação utilizadas na web (HTML, CSS e *JavaScript*). Esse tipo de aplicação pode ser acessado através do navegador do dispositivo móvel, ou, distribuído como um aplicativo instalável, através de uma *WebView*. A *WebView* é um utilitário existente no *Android* e no *iOS*, que permite a exibição de páginas da web como o *layout* de um aplicativo. Por padrão, a *WebView* não inclui recursos de um navegador convencional, e sua única função é mostrar uma página da web (MEIER, 2012).

Esse tipo de aplicação apresenta algumas desvantagens em relação às outras categorias. Nesse caso, o acesso aos recursos do dispositivo é limitado. A interface da aplicação não utiliza os recursos nativos dos dispositivos, afetando a experiência do usuário. Outro ponto negativo é a necessidade de utilizar o aplicativo conectado à internet (MONTAN; SANTOS, 2018).

3.2.3 Desenvolvimento Híbrido

Os aplicativos híbridos podem ser executados em diversos sistemas operacionais, sem a necessidade do desenvolvimento ser feito em mais de uma linguagem ou ambiente. Essa abordagem combina as categorias de desenvolvimento web e nativo, uma vez que o aplicativo é desenvolvido utilizando técnicas da web, mas, é executado, renderizado e exibido como um aplicativo nativo (AXELSSON; CARLSTRÖM, 2016).

A aplicação híbrida pode ser desenvolvida utilizando diferentes tipos de abordagens. Uma delas, é a abordagem de tradução de código. Nessa abordagem, o código fonte é compilado para *byte-code* ou código de máquina. Após o código ser compilado, componentes nativos da plataforma podem ser usados, transmitindo uma sensação de aplicação nativa para o usuário (HANSSON; VIDHALL, 2016). Alguns exemplos de *frameworks* que utilizam essa abordagem são o *Xamarin* (HERMES, 2015), utilizando a linguagem C#, e mais recentemente o *Flutter*, que utiliza a linguagem *Dart* (ZAMMETTI, 2019).

Outra abordagem é a de tempo de execução. Nesse caso, os aplicativos utilizam um interpretador, que executa o código durante o tempo de execução para chamar alguma *Application*

Programming Interface (API) nativa do dispositivo. Nessa abordagem, o código pode ser escrito em qualquer linguagem *script* que pode ser interpretada por um dispositivo. Os *frameworks React Native* (EISENMAN, 2015) e *Native Script* (ANDERSON, 2016), são dois exemplos de utilização da abordagem de tempo de execução, usando a linguagem *JavaScript*. Como o interpretador é utilizado para fazer chamadas às APIs nativas, todos os componentes do dispositivo também podem ser usados nesse tipo de abordagem. No entanto, esse é um fator que pode causar perda de desempenho em comparação com a abordagem nativa. O Quadro 2 apresenta um comparativo de vantagens e desvantagens entre as diferentes categorias de desenvolvimento *mobile*. Para a construção do quadro, foram considerados os principais critérios e aspectos encontrados nos trabalhos citados acima.

Quadro 2 – Comparativo entre o desenvolvimento Nativo, *Web Apps* e Híbrido

Critério	Nativo	Web Apps	Híbrido
Multiplataforma	Não	Sim	Sim
Custo de desenvolvimento	Alto	Baixo	Médio
Experiência do usuário	Boa	Ruim	Média
Desempenho	Alto	Baixo	Médio / Alto
Acesso aos recursos do dispositivo	Alto	Baixo	Alto
Facilidade de manutenção	Baixa	Alta	Baixa
Reconhecimento comercial	Alto	Baixo	Alto

Fonte: O Autor (2021).

É possível observar que as aplicações híbridas constituem uma solução interessante e atendem bem a maioria dos requisitos. Também são encontrados estudos comparativos onde os autores concluem que as soluções híbridas estão cada vez mais próximas das soluções nativas no quesito desempenho (BRITO *et al.*, 2018).

Geralmente, os aplicativos híbridos atuam em conjunto com uma API REST. Uma API é um serviço responsável por fornecer e manipular dados permitindo a integração entre sistemas (MASSE, 2011). O *Representational State Transfer (REST)* é um estilo de arquitetura que define um conjunto de padrões a serem usados na criação de serviços web (FIELDING, 2000). Nos dias atuais, a arquitetura REST é amplamente aplicada na construção de APIs. No caso da integração de uma API REST com um aplicativo híbrido, a API é responsável por fornecer os dados armazenados em um banco de dados remoto, para o acesso do aplicativo. Desse modo, o aplicativo somente exerce o papel de interface para o usuário.

Dentre as tecnologias que podem ser utilizadas na construção de APIs web, estão o *Java*, *Ruby* e o *Node.js*. O *Node.js* foi lançado em 2009, e proporciona um ambiente de execução *JavaScript* para a construção de aplicações web, a nível de cliente ou servidor (TEIXEIRA, 2012).

3.3 GAMIFICAÇÃO

Pesquisas mostram que estudantes com dislexia mostram menos motivação em aspectos de aprendizagem, quando comparados com outros estudantes sem o transtorno (ZISIMOPOULOS; GALANAKI, 2009). Desta forma, destaca-se a importância da aplicação de estratégias de ensino motivacionais no ensino da criança disléxica. A gamificação é um conceito recente, e, se refere ao uso de mecânicas e características de jogos (*games*) para motivar comportamentos e facilitar o aprendizado de pessoas em situações reais (DETERDING *et al.*, 2011).

Denny (2013) realizou um estudo analisando 1000 estudantes universitários, e o uso de uma ferramenta de aprendizagem gamificada. O estudo observou que os estudantes que usaram a versão gamificada da plataforma responderam mais questões e usaram o sistema mais frequentemente que os estudantes que usaram a versão não gamificada. O pesquisador também apontou que os estudantes gostaram bastante do sistema de recompensa, e manifestaram uma preferência pela inclusão deste na versão original da ferramenta.

Gooch et al. (2015) analisou o uso de uma plataforma gamificada chamada *ClassDojo* por estudantes com dislexia dos primeiros anos escolares. O estudo mostrou que o uso da plataforma trouxe benefícios para os estudantes, e, destacou a importância do professor em adaptar o modo de uso da plataforma, dependendo do grau de severidade do transtorno em cada aluno.

Um estudo de Francheschini et al. (2013), mostrou que jogos de ação podem ajudar crianças com dislexia a lerem mais rápido e eficientemente. Segundo os autores, o aprimoramento das habilidades de atenção visuo-espacial podem impactar diretamente em uma melhor leitura em crianças com o transtorno.

3.4 RECONHECIMENTO E SÍNTESE DE FALA

O reconhecimento de fala é uma área de linguística computacional, cujo objetivo é permitir o reconhecimento e a transcrição de linguagem falada de maneira automática. A síntese de fala é o processo de produção artificial de fala humana (JUANG; RABINER, 2005). Atualmente, diversas empresas disponibilizam serviços de reconhecimento e síntese de fala para desenvolvedores que queiram integrar esse tipo de funcionalidade à seus aplicativos. Geralmente, esses serviços são disponibilizados por meio de APIs.

Uma dessas empresas é o Google, que oferece tais serviços através da *Google Cloud Platform*¹. A *Google Cloud Platform* é uma suíte de computação em nuvem, que fornece um conjunto de ferramentas incluindo armazenamento de dados, análise de dados e inteligência artificial. Dois dos serviços que utilizam a tecnologia de inteligência artificial do Google, são o *Speech-to-Text* e o *Text-to-Speech*.

O recurso *Speech-to-Text* permite a transcrição de uma fala, em texto. Com essa API,

¹ <<https://cloud.google.com/>>

é possível que a transcrição seja feita em tempo real, ou, por meio de um áudio armazenado em um arquivo. Segundo a página oficial do produto, a API é compatível com mais de 125 idiomas e variantes, incluindo o Português do Brasil. Utilizando algoritmos de redes neurais, o *Speech-to-Text* faz com que a precisão de transcrição seja um fator de destaque do serviço. Além disso, é possível fornecer dicas para personalizar o reconhecimento de fala, com o objetivo de transcrever termos específicos e palavras raras. O *Text-to-Speech* permite a conversão de texto, em voz. Segundo a página do produto, a voz é gerada com entonação similar à humana. É possível escolher entre um grupo de mais de 220 vozes em mais de 40 idiomas e variantes. Além disso, o *Text-to-Speech* permite a personalização do tom e da velocidade da voz selecionada.

Da mesma forma que o Google, a IBM oferece através da *IBM Cloud*², ferramentas de transcrição de áudio para texto, e de texto para áudio, sendo elas o *Watson Speech to Text*, e o *Watson Text to Speech*. Outra empresa que oferece serviços similares é a Microsoft, através do *Microsoft Azure Cognitive Services*³.

² <<https://cloud.ibm.com/>>

³ <<https://azure.microsoft.com/>>

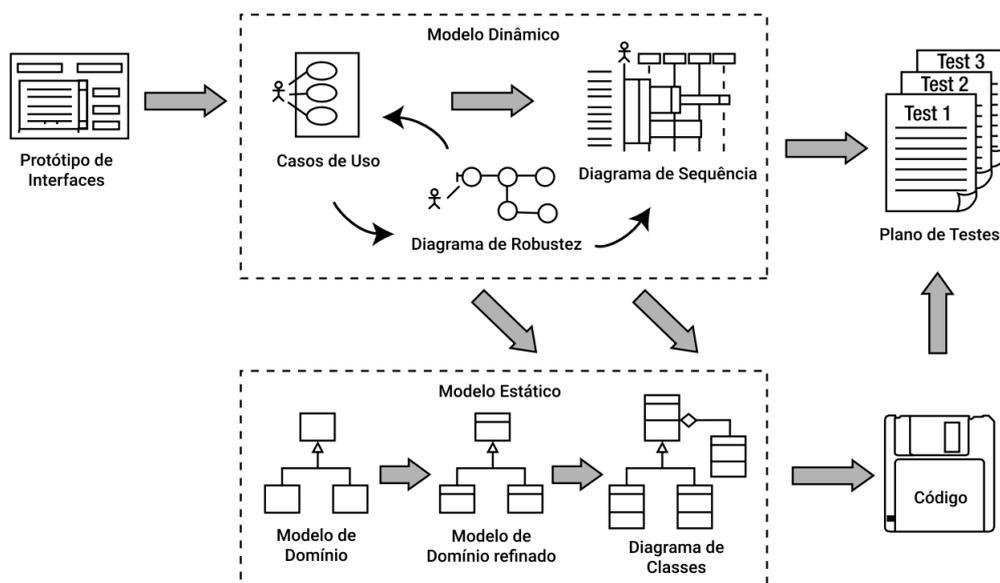
4 METODOLOGIA

No desenvolvimento do projeto foi utilizada a metodologia ICONIX (ROSENBERG; STEPHENS; COLLINS-COPE, 2005). Segundo os seus criadores, o ICONIX é um processo minimalista de modelagem de *software*, orientado a casos de uso. Através de uma maneira simples e iterativa, o principal objetivo do ICONIX é chegar no estágio de escrita de código o mais rápido possível, sem perder os benefícios de uma análise inicial e de um processo de desenvolvimento. Desta forma, essa metodologia beneficia projetos de médio ou pequeno porte, e com equipes reduzidas, como é o caso do presente trabalho.

O ICONIX utiliza um subconjunto de diagramas e técnicas do padrão *Unified Modeling Language (UML)* (RUMBAUGH; JACOBSON; BOOCH, 1999). Os artefatos são divididos em duas visões de projeto, o modelo estático e o modelo dinâmico, que podem ser desenvolvidos paralelamente. O modelo estático é constituído pelo modelo de domínio e o diagrama de classes, que representam a modelagem e o funcionamento do sistema sem nenhuma interação com o usuário. O modelo dinâmico leva em consideração as interações entre o usuário e o sistema, sendo representadas pelo diagrama de casos de uso e o diagrama de sequência. Ao decorrer do desenvolvimento do modelo dinâmico, o modelo estático é incrementalmente refinado.

A Figura 7 mostra uma visão geral da metodologia ICONIX e seus principais artefatos.

Figura 7 – Visão geral do processo ICONIX



Fonte: Adaptado de ROSENBERG; STEPHENS; COLLINS-COPE (2005).

5 PROJETO

Neste capítulo será apresentado o projeto da solução proposta, que é constituída por um aplicativo para dispositivos móveis, com a finalidade de auxiliar crianças com dislexia a aprimorarem suas habilidades de leitura, e uma web API REST, capaz de fornecer os dados necessários para o funcionamento do aplicativo. O programa de remediação que serviu como inspiração para a criação das principais funcionalidades foi o criado por Santos e Capellini (SANTOS; CAPELLINI, 2020).

Como descrito na seção 2.5, o programa proposto por Santos e Capellini é composto por diferentes pranchas de nomeação. As pranchas são compostas por 5 estímulos visuais que podem ser cores, figuras, letras, palavras e números, ordenados em 10 fileiras. Cada prancha é composta por somente um tipo de estímulo e nível de dificuldade, de acordo com seu tipo silábico. No caso da aplicação proposta, as pranchas foram adaptadas de modo que o indivíduo visualize e faça a nomeação de um estímulo por vez. Ao final da primeira etapa de nomeação, é feita uma revisão dos estímulos nomeados corretamente e incorretamente, assim como é descrito no programa original. Em seguida, é feita a segunda etapa, onde os mesmos estímulos da primeira etapa devem ser nomeados novamente. Os estímulos escolhidos são compostos por palavras monossilábicas, dissilábicas, trissilábicas e polissilábicas.

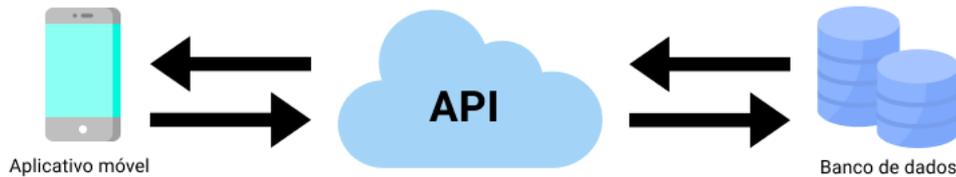
As duas principais funcionalidades do aplicativo são a atividade de nomeação rápida de palavras, e a atividade de nomeação rápida de figuras. Para a realização e correção das nomeações feitas pelo usuário, a aplicação conta com a integração de um serviço de reconhecimento de voz. Também é necessário a disponibilização de instruções em formato de áudio, para os usuários com um maior grau de dificuldade de leitura. Desta forma, também deve ser realizada a integração com um serviço de síntese de voz. Os serviços escolhidos são o *Speech-to-Text*, e, o *Text-to-Speech*, do Google. As escolhas basearam-se na popularidade das ferramentas do Google, na clareza e qualidade da documentação, na quantidade de materiais auxiliares para guiar a integração com as tecnologias de desenvolvimento, e no suporte para o idioma Português do Brasil.

Também foram desenvolvidas funcionalidades secundárias, como, um perfil para o usuário, onde é possível alterar informações como o nome e a senha, além da escolha de um avatar. Também foi desenvolvido um sistema de *ranking* de acordo com o nível de cada usuário. O nivelamento é de acordo com uma pontuação, obtida pelo usuário ao completar cada atividade de nomeação. As funcionalidades secundárias vão de encontro com os aspectos emocionais envolvidos na dislexia, que, como descrito na seção 3.3 do capítulo 3, são fatores importantes para a superação das dificuldades impostas pelo transtorno.

Algumas funcionalidades, como, o *ranking*, implicam no desenvolvimento de uma apli-

cação capaz de realizar o armazenamento e o controle de dados remotamente. Desta forma, também foi desenvolvida uma web API REST, utilizando a tecnologia *Node.js*. Na Figura 8, a comunicação entre o aplicativo móvel, a API e o banco de dados remoto é ilustrada.

Figura 8 – Comunicação da solução proposta



Fonte: O Autor (2021).

A tecnologia escolhida para o desenvolvimento do aplicativo móvel é o *React Native*. A escolha de cada tecnologia é feita de acordo com as vantagens descritas no capítulo 3, a familiaridade e o conhecimento técnico do autor do trabalho, e também devido a boa interoperabilidade que é oferecida, pois, desta forma, o código é escrito em uma única linguagem de programação (*JavaScript*).

A seguir, são abordadas todas as etapas da metodologia ICONIX. Primeiramente, é elaborada a etapa de análise de requisitos. Essa etapa é constituída pela apresentação dos requisitos funcionais e não funcionais do sistema, modelo de domínio e o diagrama de casos de uso. Após essa etapa, é feito um detalhamento dos principais casos de uso da aplicação, incluindo diagrama de robustez, diagrama de sequência e diagrama de classes. Também foram elaborados protótipos de interface para o aplicativo, porém, a apresentação dos protótipos é substituída por capturas de tela, que foram obtidas após a ferramenta ser desenvolvida. Todas as nomenclaturas que serão encontradas nos diagramas foram traduzidas para o idioma inglês, para corresponderem com sua representação no código da aplicação. O Quadro 3 descreve brevemente cada artefato englobado pelo processo ICONIX.

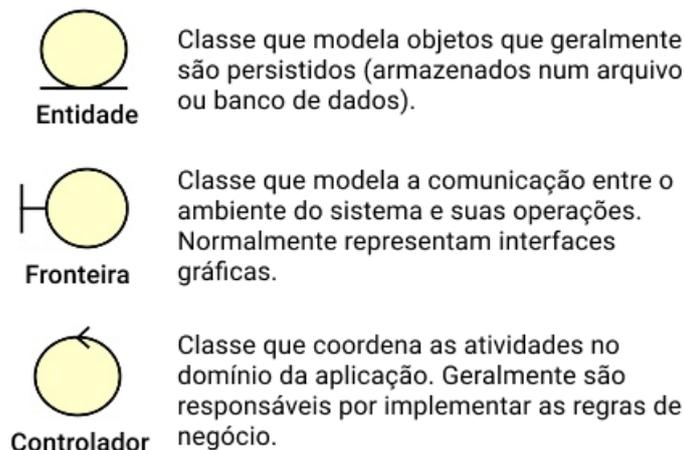
Quadro 3 – Artefatos do processo ICONIX

Artefato	Descrição
Modelo de Domínio	Representado por classes que identificam objetos de domínio do mundo real.
Casos de Uso	Define os requisitos comportamentais da aplicação.
Diagrama de Robustez	Identifica o conjunto de objetos que irão estar envolvidos com um determinado caso de uso.
Diagrama de Sequência	Define uma sequência de comportamentos para a realização de um determinado caso de uso.
Diagrama de Classes	Conclui o modelo estático e representa a estrutura das classes da aplicação, juntamente com suas relações e atributos.

Fonte: O Autor(2021).

Em cada diagrama mencionado, são encontrados diferentes tipos de objetos que compartilham um mesmo comportamento dentro da aplicação. O modelo que representa esses objetos é denominado de classe. A Figura 9 apresenta uma descrição de cada tipo de classe, de acordo com o padrão UML (RUMBAUGH; JACOBSON; BOOCH, 1999).

Figura 9 – Tipos de classes do padrão UML



Fonte: O Autor (2021).

5.1 REQUISITOS

Os requisitos são as capacidades e condições às quais a aplicação deve atender. Existem duas classificações de requisitos, são elas: Requisitos Funcionais (RF) e Requisitos Não-Funcionais (RNF). Os requisitos funcionais compreendem as necessidades ou solicitações a serem realizadas pelo *software*. Também pode-se dizer, que, os requisitos funcionais tratam do que o sistema deve fazer, ou, permitir. Por outro lado, os requisitos não funcionais referem-se aos critérios que qualificam os requisitos funcionais. Tais critérios podem se referir à qualidade do *software*, restrições técnicas, entre outros (LARMAN, 2004).

O Quadro 4 apresenta os requisitos funcionais do projeto.

Quadro 4 – Requisitos funcionais do projeto

RF	Descrição
RF-001	A aplicação deve permitir o login de usuários.
RF-002	A aplicação deve permitir o cadastro de usuários.
RF-003	A aplicação deve permitir a realização da nomeação de palavras.
RF-004	A aplicação deve permitir a realização da nomeação de figuras.
RF-005	A aplicação deve exibir um tutorial introdutório na primeira vez em que o usuário iniciar uma atividade de nomeação.
RF-006	A aplicação deve atribuir uma pontuação para o usuário, a cada atividade de nomeação realizada.
RF-007	A aplicação deve permitir a realização de uma revisão durante as atividades de nomeação.
RF-008	A aplicação deve permitir a visualização do perfil do usuário.
RF-009	A aplicação deve informar um nível de pontuação de cada usuário.
RF-010	A aplicação deve permitir a alteração do nome do usuário.
RF-011	A aplicação deve permitir a alteração da senha do usuário.
RF-012	A aplicação deve permitir a alteração do avatar do usuário.
RF-013	A aplicação deve permitir a visualização de um ranking de usuários.
RF-014	A aplicação deve permitir a alteração da fonte dos textos do aplicativo.
RF-015	A aplicação deve permitir a alteração da velocidade de fala das instruções por áudio.
RF-016	A aplicação deve permitir a escolha de privacidade do perfil do usuário.
RF-017	A aplicação deve permitir a exclusão da conta do usuário.

Fonte: O Autor(2021).

O Quadro 5 apresenta os requisitos não funcionais do projeto.

Quadro 5 – Requisitos não funcionais do projeto

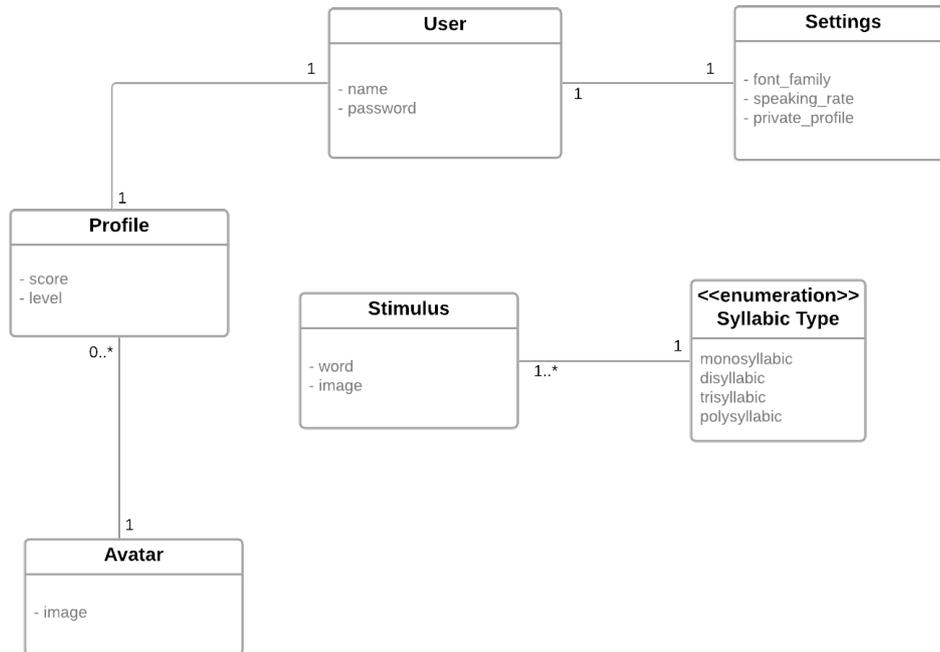
RF	Descrição
RNF-001	O dispositivo móvel deve estar conectado à internet.
RNF-002	O serviço de sintetização de voz deve ser utilizado.
RNF-003	O serviço de reconhecimento de voz deve ser utilizado.

Fonte: O Autor(2021).

5.2 MODELO DE DOMÍNIO

O modelo de domínio compreende uma estimativa preliminar do modelo estático (diagrama de classes), baseado especificadamente em classes que envolvem o domínio do problema. Ou seja, o modelo de domínio é somente uma forma de representar visualmente coisas do mundo real, e conceitos relacionados ao problema que a aplicação se propõe a resolver (ROSENBERG; STEPHENS; COLLINS-COPE, 2005). Nesse modelo, as classes conceituais que representam o mundo real são apresentadas sem qualquer método ou função de sistema. Apenas são demonstradas as associações entre elas, e, opcionalmente, alguns de seus atributos. A Figura 10 apresenta o modelo de domínio do projeto.

Figura 10 – Modelo de domínio

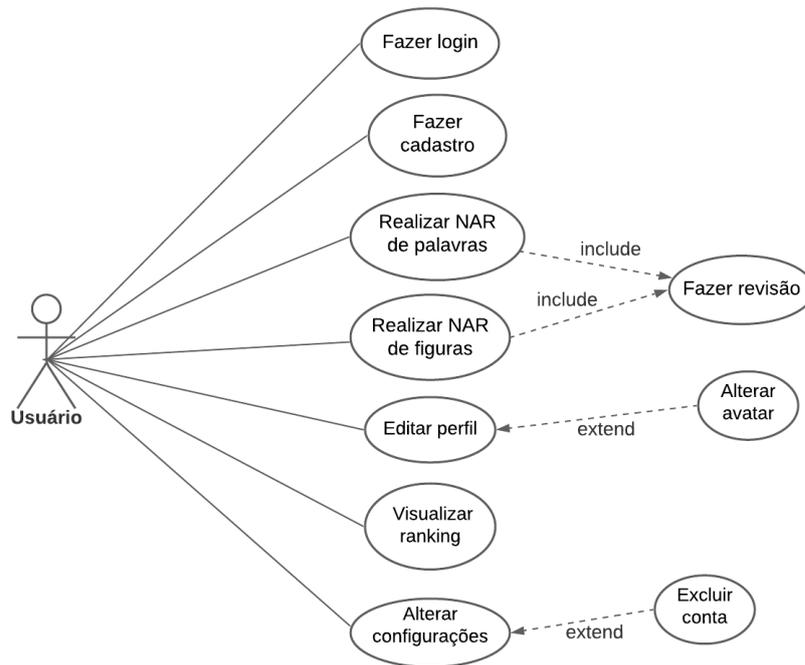


Fonte: O Autor (2021).

5.3 DIAGRAMA DE CASOS DE USO

O diagrama de casos de uso é utilizado para demonstrar um conjunto de casos de uso, atores e suas associações (LARMAN, 2004). Os atores representam os papéis dos usuários do sistema. No presente projeto apenas um ator foi identificado, que é o usuário do aplicativo. Os casos de uso, propriamente ditos, são representações dos requisitos funcionais do sistema, e suas ligações com os atores são expressas através dos relacionamentos (LARMAN, 2004). Na Figura 11, o diagrama de casos de uso do projeto pode ser visto.

Figura 11 – Diagrama de casos de uso



Fonte: O Autor (2021).

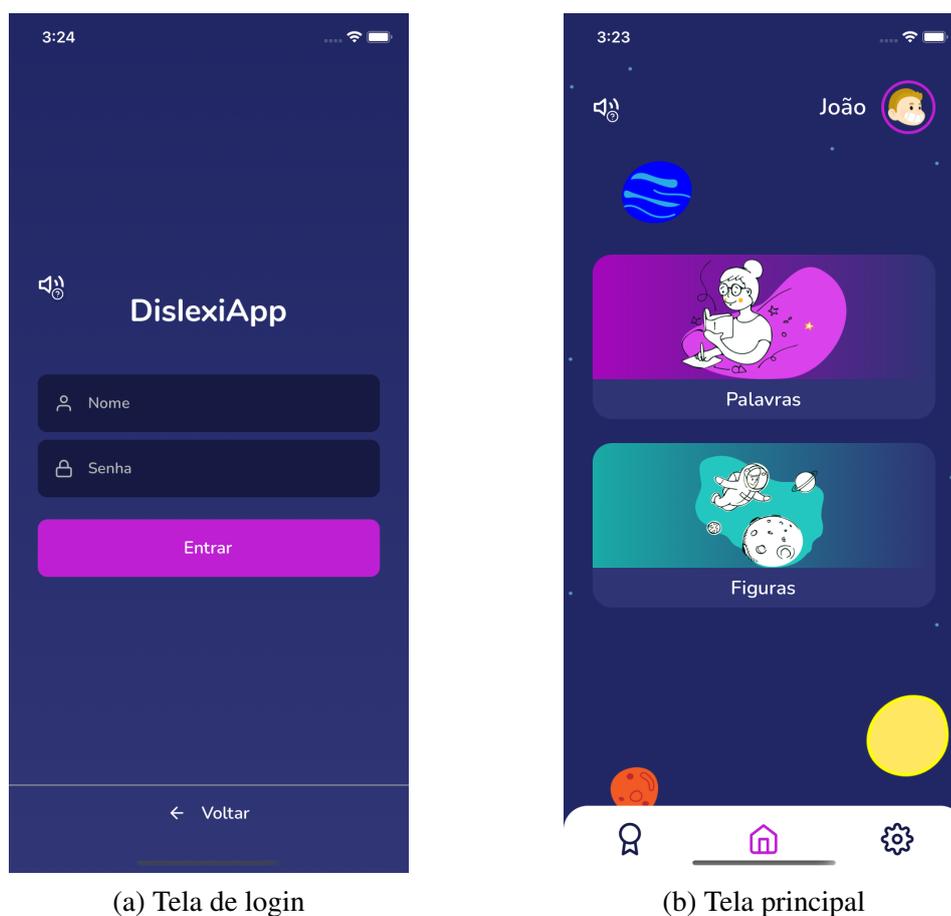
5.4 DETALHAMENTO DE CASOS DE USO

Nesta seção, cada caso de uso representado na Figura 11 será detalhado, conforme proposto pela metodologia ICONIX.

5.4.1 Caso de Uso 1 - Fazer Login

O caso de uso "Fazer Login" atende ao requisito funcional RF-001. Na Figura 12 são apresentadas capturas de tela do aplicativo, onde a ação de realizar login é realizada. Na Figura 12a é mostrada a tela de login, e na Figura 12b é mostrada a tela principal do aplicativo, mostrada ao usuário após a autenticação ser bem sucedida.

Figura 12 – Capturas de tela 1: Fazer Login

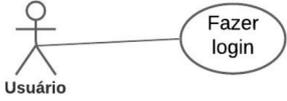


Fonte: O Autor (2021).

A partir da tela de login, o usuário pode se autenticar informando seu nome de usuário e senha, ou, ir para a tela de cadastro caso não possuir uma conta. Caso a autenticação seja bem sucedida, o usuário é redirecionado para a tela principal. Todas as telas contam com um botão de ajuda que aciona a reprodução de uma instrução por áudio, para que os usuários com maiores dificuldades possam entender o que pode ser feito naquela tela.

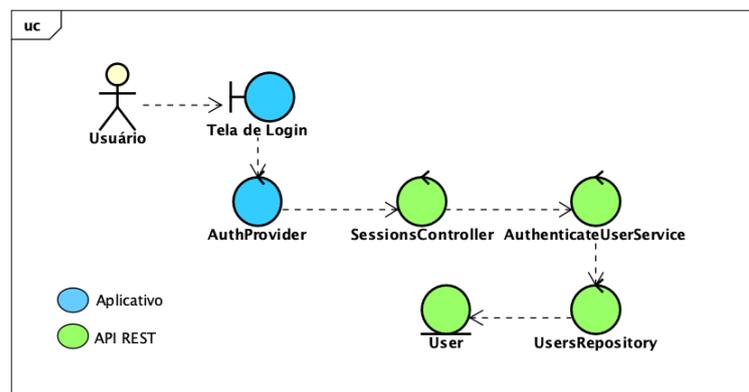
Na Figura 13 é apresentado o detalhamento, e na Figura 14 é apresentado o diagrama de robustez do caso de uso "Fazer Login".

Figura 13 – Caso de Uso 1: Fazer Login

			
Versão	1.0	Data Criação	03/11/2020
Autor	Anderson Begossi	Data Atualização	03/11/2020
Descrição	O aplicativo deve permitir que o usuário faça sua autenticação.		
RF Associados	001		
RNF Associados	001 e 002.		
Atores	Usuário		
Pré-condições	O usuário deve possuir uma conta.		
Pós-condições	O usuário é redirecionado para a tela principal.		
Fluxo principal	<ol style="list-style-type: none"> 1. Aplicativo solicita nome e senha. 2. Usuário fornece nome e senha. 3. Usuário pressiona o botão Entrar. 4. Aplicativo valida os dados: nome - obrigatório senha - obrigatório 5. Aplicativo redireciona o usuário para a tela principal. 		
Fluxos alternativos	<ol style="list-style-type: none"> 4. a) Campos obrigatórios. <ol style="list-style-type: none"> 1. Aplicativo informa que o nome é obrigatório. 2. Aplicativo informa que a senha é obrigatória. b) Login inválido. <ol style="list-style-type: none"> 1. Aplicativo informa que os dados estão incorretos. 		
Requisitos especiais			

Fonte: O Autor (2021).

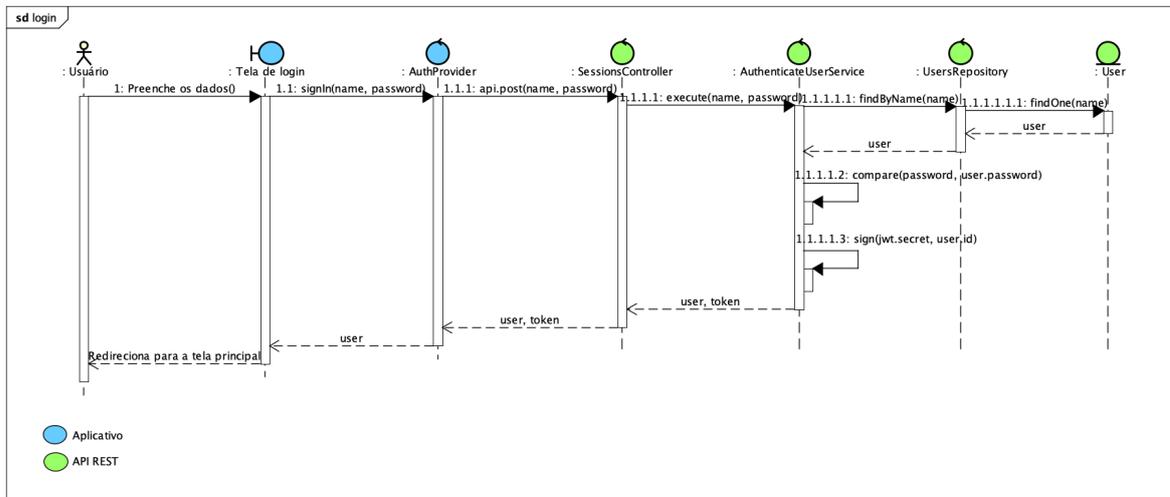
Figura 14 – Diagrama de robustez 1: Fazer Login



Fonte: O Autor (2021).

Na Figura 15 é apresentado o diagrama de sequência.

Figura 15 – Diagrama de sequência 1: Fazer Login

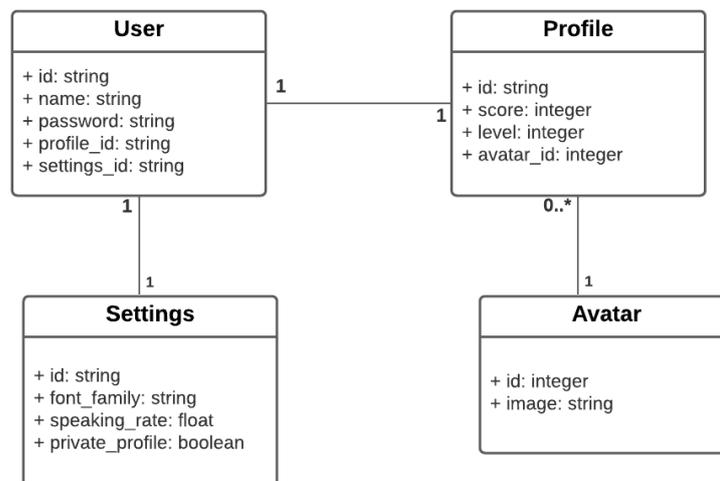


Fonte: O Autor (2021).

Conforme pode ser visto nos diagramas acima, as classes na cor azul são referentes ao aplicativo, e as classes na cor verde são referentes à web API REST. Após o usuário preencher os dados na tela de login, o aplicativo realiza uma requisição para a API. A API recebe a requisição, se encarrega de autenticar os dados e devolver uma resposta adequada para o aplicativo.

Na Figura 16 é apresentado o diagrama das classes envolvidas no caso de uso. As classes Profile, Settings e Avatar não foram incluídas no diagrama de sequência e no diagrama de robustez pois estão envolvidas implicitamente no retorno do objeto User.

Figura 16 – Diagrama de classes 1: Fazer Login



Fonte: O Autor (2021).

5.4.2 Caso de Uso 2 - Fazer Cadastro

O caso de uso "Fazer Cadastro" atende ao requisito funcional RF-002. Na Figura 17 é apresentada a captura de tela onde o caso de uso é realizado pelo usuário.

Figura 17 – Captura de tela 2: Fazer Cadastro



Fonte: O Autor (2021).

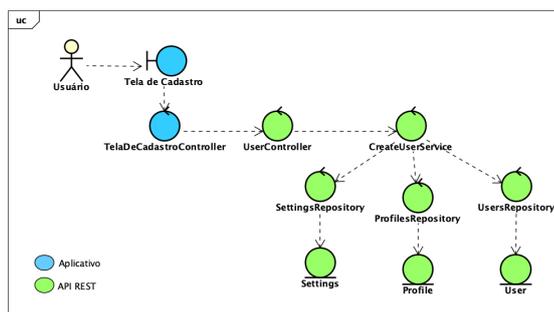
A partir da tela de cadastro, o usuário pode criar sua conta para se autenticar no aplicativo. Na Figura 18 é apresentado o detalhamento, e na Figura 19 é apresentado o diagrama de robustez do caso de uso "Fazer Cadastro".

Figura 18 – Caso de Uso 2: Fazer Cadastro

			
Versão	1.0	Data Criação	10/11/2020
Autor	Anderson Begossi	Data Atualização	10/11/2020
Descrição	O aplicativo deve permitir que o usuário crie sua conta.		
RF Associados	002		
RNF Associados	001 e 002.		
Atores	Usuário		
Pré-condições			
Pós-condições	A conta de usuário é criada e o usuário é redirecionado para a tela de login.		
Fluxo principal	<ol style="list-style-type: none"> 1. Aplicativo solicita nome, senha e confirmação de senha. 2. Usuário fornece nome, senha e confirmação de senha. 3. Usuário pressiona o botão Cadastrar. 4. Aplicativo valida os dados: nome - obrigatório senha - mínimo de 6 dígitos, obrigatório confirmação de senha - correspondente ao campo senha 5. Aplicativo exibe uma mensagem informando que a conta foi criada com sucesso. 6. Aplicativo redireciona o usuário para a tela de login. 		
Fluxos alternativos	<ol style="list-style-type: none"> 4. a) Campos obrigatórios. <ol style="list-style-type: none"> 1. Aplicativo informa que o nome é obrigatório. 2. Aplicativo informa que a senha é obrigatória. 3. Aplicativo informa que a confirmação de senha é obrigatória. b) Campos inválidos. <ol style="list-style-type: none"> 1. Aplicativo informa que a senha deve ter pelo menos 6 dígitos. 2. Aplicativo informa que as senhas não correspondem. 5. a) Nome já existente. <ol style="list-style-type: none"> 1. Aplicativo informa que o nome digitado já existe. 		
Requisitos especiais			

Fonte: O Autor (2021).

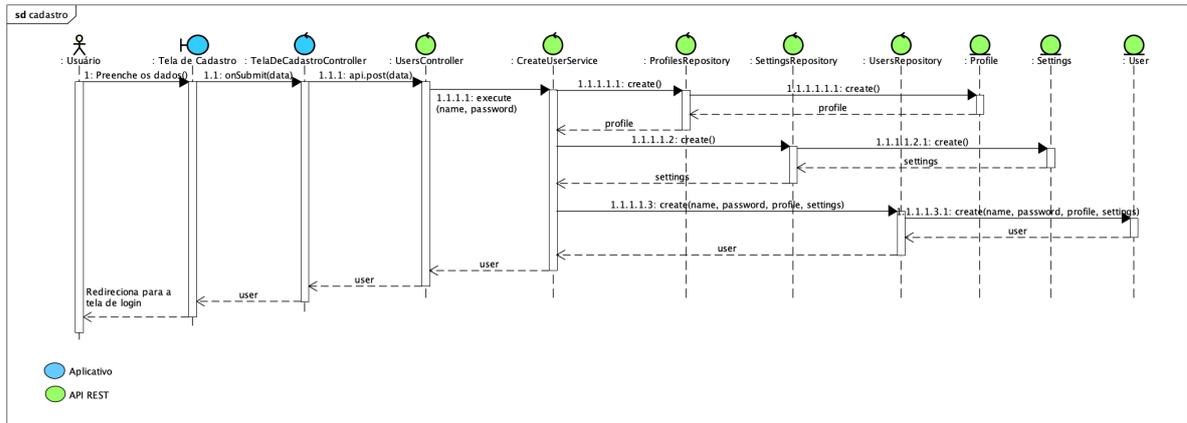
Figura 19 – Diagrama de robustez 2: Fazer Cadastro



Fonte: O Autor (2021).

Na Figura 20 é apresentado o diagrama de sequência do caso de uso "Fazer Cadastro".

Figura 20 – Diagrama de seqüência 2: Fazer Cadastro



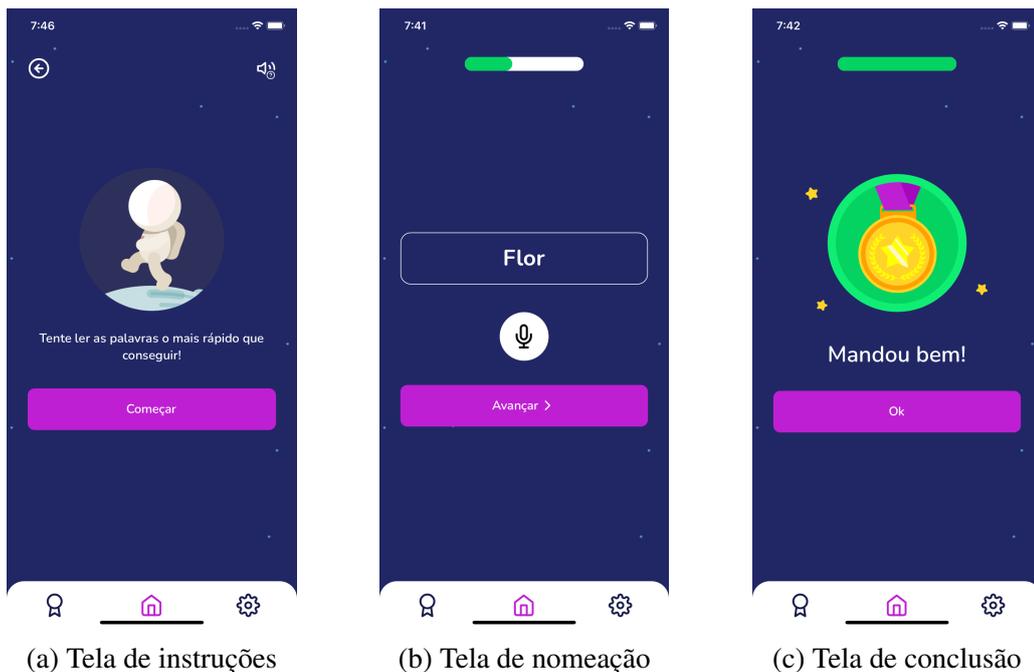
Fonte: O Autor (2021).

O diagrama de classes do caso de uso "Fazer Cadastro" é o mesmo que o apresentado na Figura 16.

5.4.3 Caso de Uso 3 - Realizar NAR de Palavras

O caso de uso "Realizar NAR de Palavras" atende aos requisitos funcionais RF-003, RF-005 e RF-006. Na Figura 21 são apresentadas as capturas de tela. A Figura 21a apresenta a tela de instruções, a Figura 21b apresenta a tela de nomeação e a Figura 21c apresenta a tela que é mostrada ao usuário ao concluir a atividade.

Figura 21 – Capturas de tela 3: Realizar NAR de Palavras



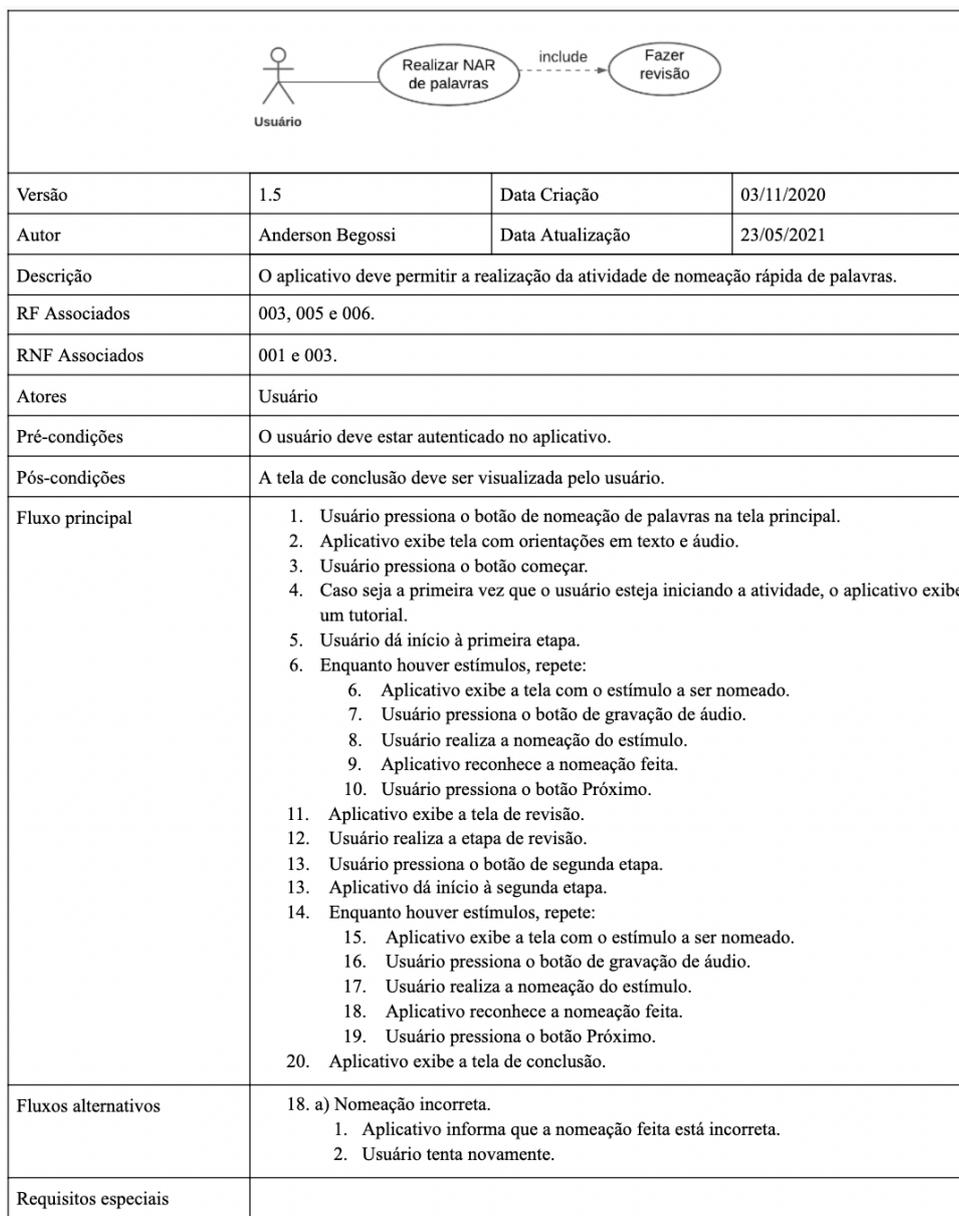
Fonte: O Autor (2021).

A realização da atividade de nomeação é feita em 2 etapas, incluindo uma revisão, conforme o padrão do programa de remediação de Santos e Capellini (2020). Caso seja a primeira vez que o usuário esteja iniciando a atividade, um tutorial é exibido. Após a visualização do tutorial, uma sequência de 5 estímulos correspondentes ao nível do é será apresentada. Todos os usuários recém cadastrados iniciam com o nível 0. Para cada estímulo, o usuário deverá realizar a nomeação pressionando o botão de gravação de áudio, que é mostrado no centro da tela com um ícone de microfone. O áudio gravado é então enviado para a API, onde é transcrito com a utilização do serviço *Speech-to-Text*. Após a transcrição ser obtida, é definido um resultado através de uma comparação de *strings* entre a transcrição e a palavra que corresponde ao estímulo nomeado.

Após finalizar a primeira etapa, é iniciada a etapa de revisão. Durante a revisão, o usuário pode conferir se nomeou o estímulo corretamente, e também ver a representação do mesmo em forma de figura, palavra e áudio. Ao finalizar a revisão, o usuário pode iniciar a segunda etapa, onde os mesmos estímulos da primeira etapa deverão ser nomeados novamente. Nessa etapa, a correção da nomeação é feita imediatamente após a conclusão da mesma.

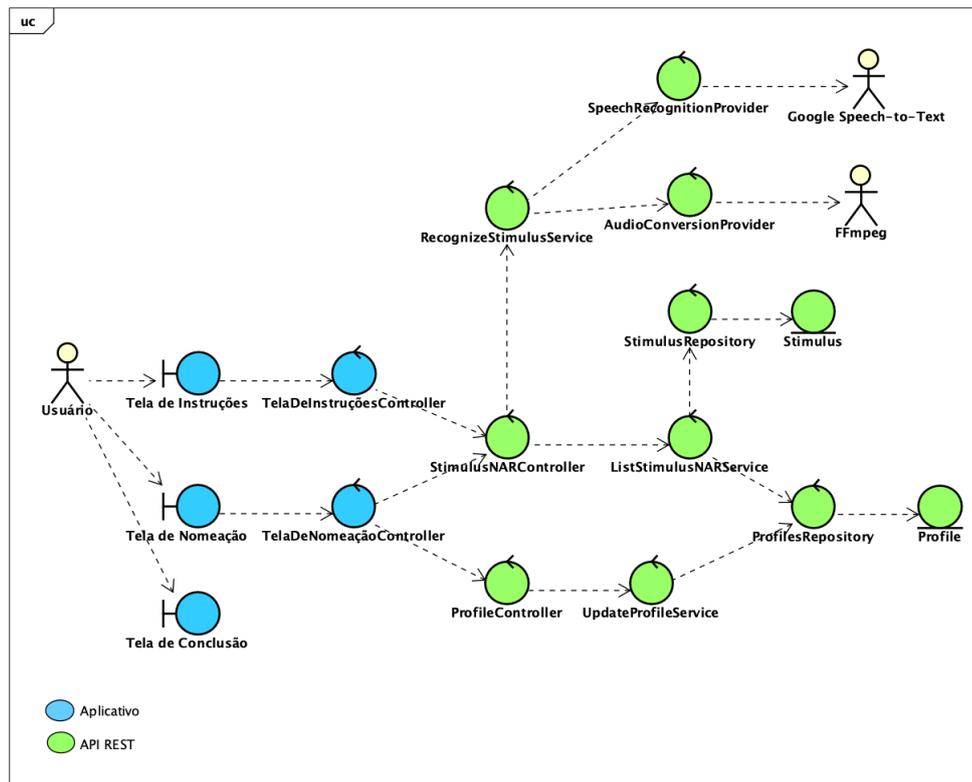
Na Figura 22 é apresentado o detalhamento, e na Figura 23 é apresentado o diagrama de robustez do caso de uso "Realizar NAR de Palavras".

Figura 22 – Caso de Uso 3: Realizar NAR de Palavras



Fonte: O Autor (2021).

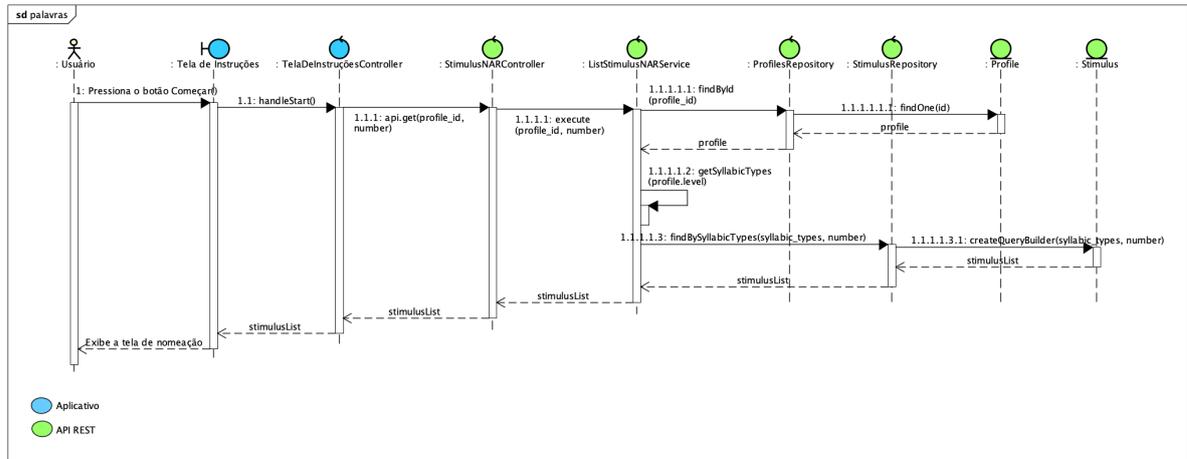
Figura 23 – Diagrama de robustez 3: Realizar NAR de Palavras



Fonte: O Autor (2021).

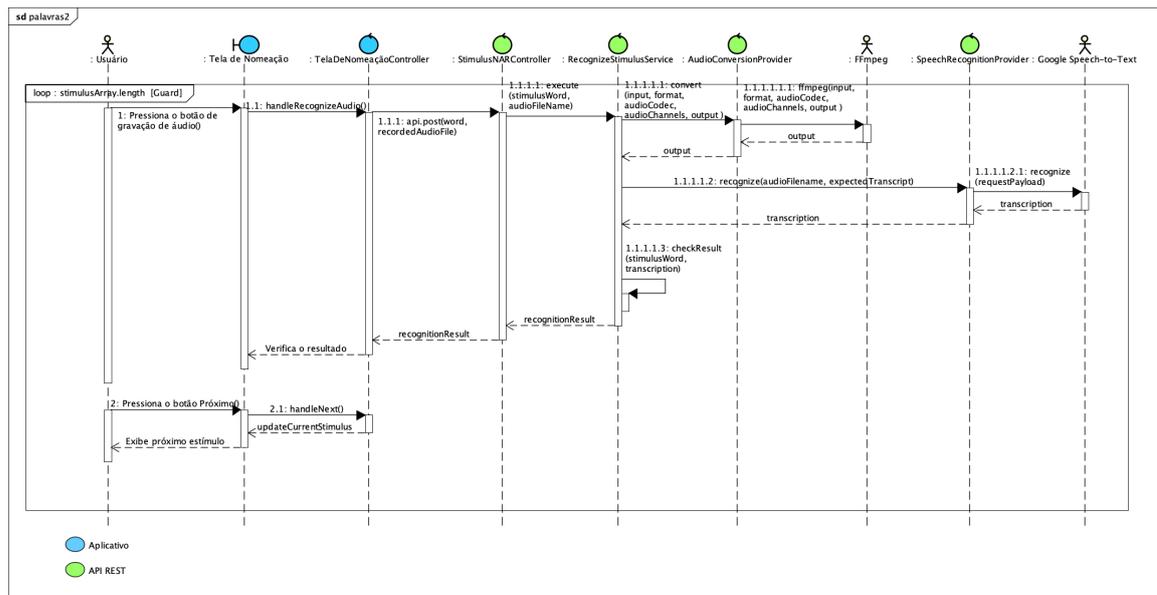
Devido a complexidade do caso de uso, o diagrama de sequência foi dividido em 3 partes. A primeira delas, apresentada na Figura 24, demonstra a sequência de interações para obter a lista de estímulos a ser nomeada pelo usuário. Na segunda parte, apresentada na Figura 25, é representada a sequência para fazer o reconhecimento do estímulo nomeado pelo usuário. Por último, é apresentado na Figura 26, a sequência para realizar a atualização do perfil do usuário com a pontuação obtida ao completar a atividade.

Figura 24 – Diagrama de sequência 3: Realizar NAR de Palavras, parte 1



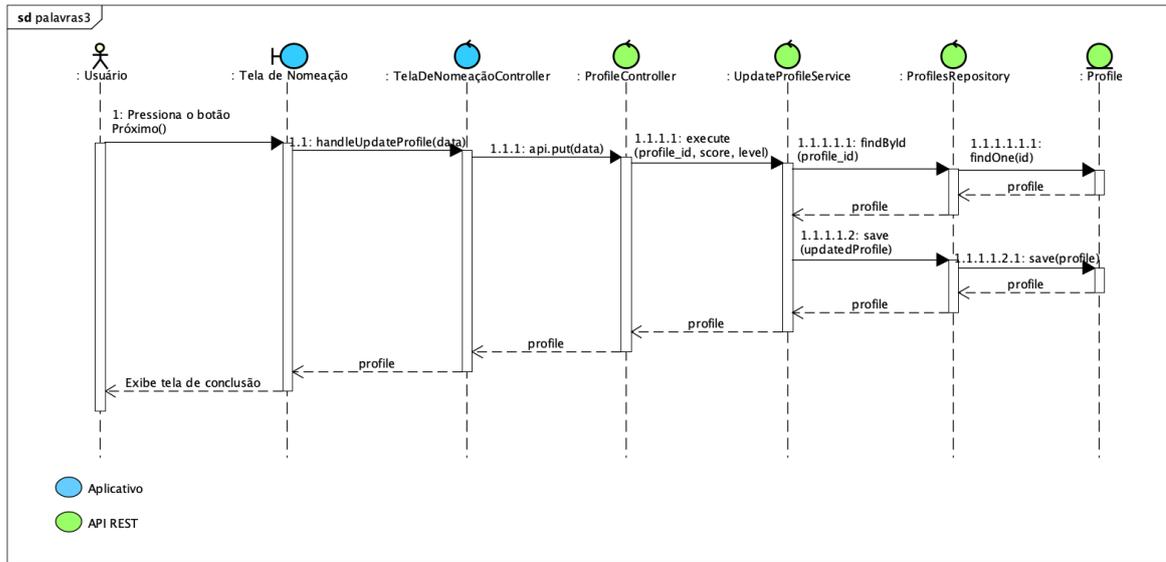
Fonte: O Autor (2021).

Figura 25 – Diagrama de sequência 3: Realizar NAR de Palavras, parte 2



Fonte: O Autor (2021).

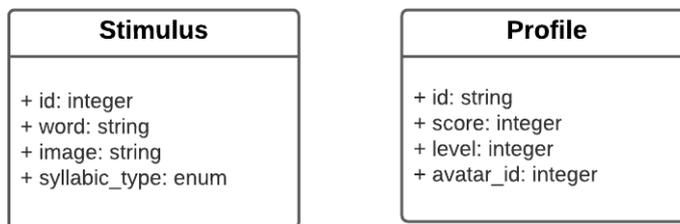
Figura 26 – Diagrama de sequência 3: Realizar NAR de Palavras, parte 3



Fonte: O Autor (2021).

Na Figura 27 é apresentado o diagrama de classes do caso de uso "Realizar NAR de Palavras".

Figura 27 – Diagrama de classes 3: Realizar NAR de Palavras

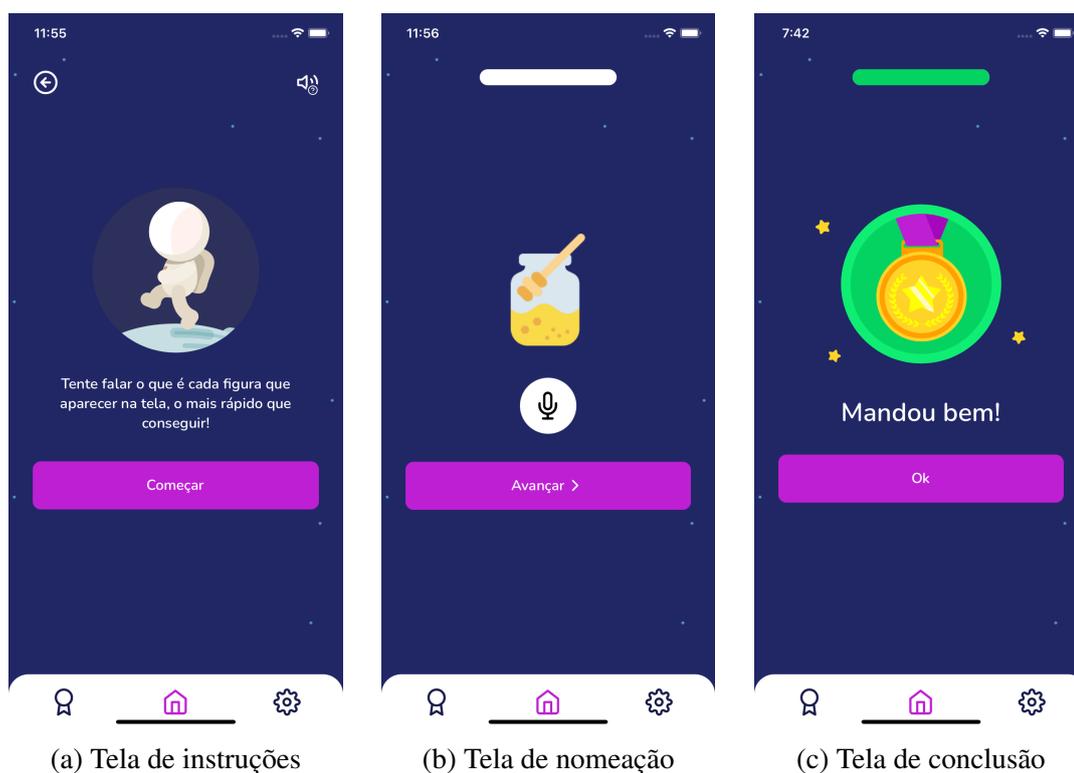


Fonte: O Autor (2021).

5.4.4 Caso de Uso 4 - Realizar NAR de Figuras

O caso de uso "Realizar NAR de Figuras" atende aos requisitos funcionais RF-004, RF-005 e RF-006. Na Figura 28 são apresentadas as capturas de tela. A Figura 28a apresenta a tela de instruções, a Figura 28b apresenta a tela de nomeação e a Figura 28c apresenta a tela de conclusão.

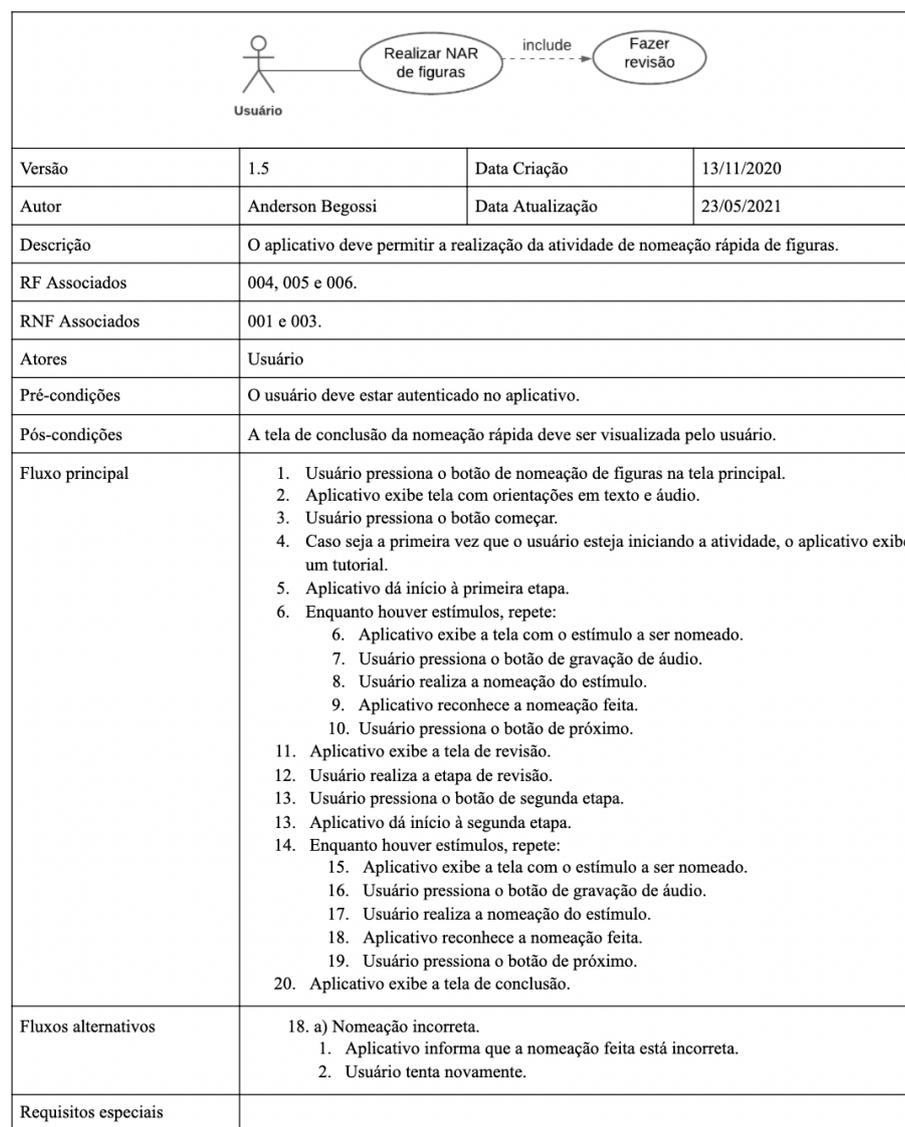
Figura 28 – Capturas de tela 4: Realizar NAR de Figuras



Fonte: O Autor (2021).

A realização da atividade de nomeação de figuras segue as mesmas etapas da nomeação de palavras. As atividades se diferem na apresentação do estímulo para o usuário, sendo esta em forma de figura. Na Figura 29 é apresentado o detalhamento do caso de uso.

Figura 29 – Caso de Uso 4: Realizar NAR de Figuras



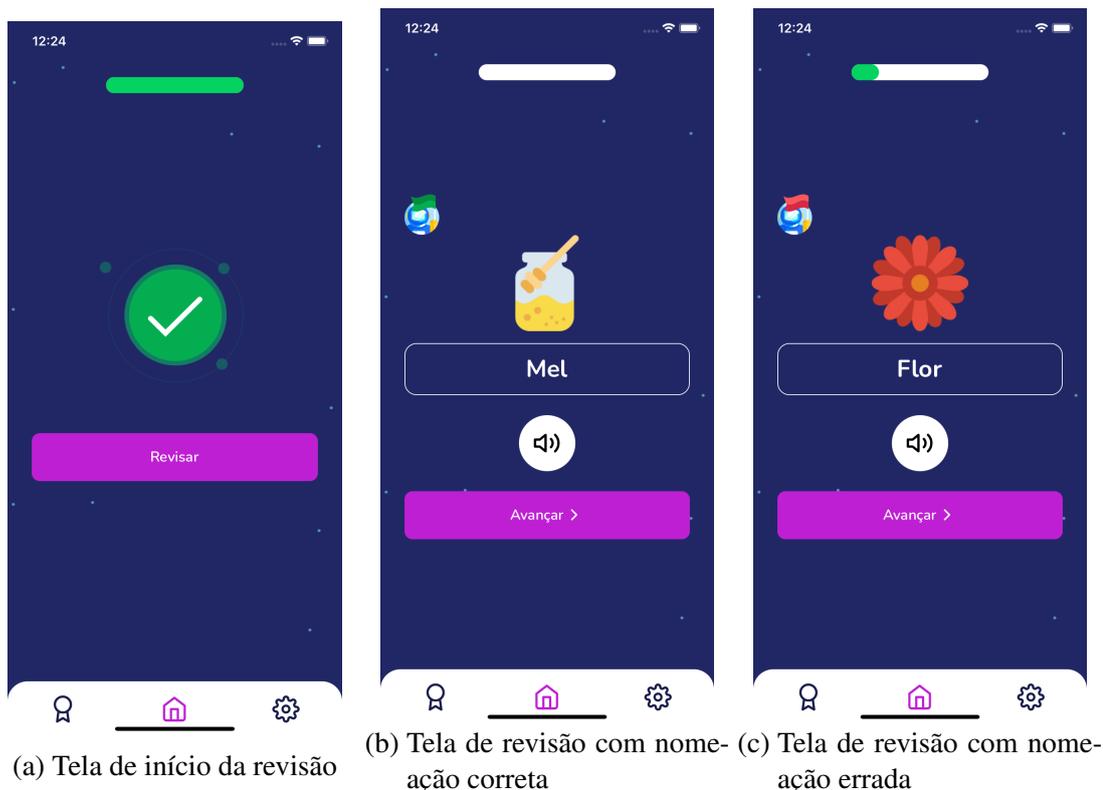
Fonte: O Autor (2021).

O diagrama de robustez do caso de uso "Realizar NAR de Figuras" é o mesmo que o apresentado na Figura 23. O diagrama de sequência e o diagrama de classes são os mesmos do caso de uso "Realizar NAR de Palavras".

5.4.5 Caso de Uso 5 - Fazer Revisão

O caso de uso "Fazer Revisão" atende ao requisito funcional RF-007. Na Figura 30 são apresentadas as capturas de tela. A Figura 30a apresenta a tela de início da revisão, a Figura 30b apresenta a tela de revisão destacando uma nomeação feita corretamente, e a Figura 30c apresenta a tela de revisão quando a nomeação foi feita erroneamente.

Figura 30 – Capturas de tela 5: Fazer Revisão



Fonte: O Autor (2021).

Nesta etapa, é exibida a figura e a palavra correspondente ao estímulo, além de reproduzida a pronúncia do mesmo através de um áudio, com a utilização do serviço *Text-to-Speech*. Também é mostrada a figura de um astronauta no lado esquerdo da tela, que exibe uma bandeira verde caso a nomeação foi realizada corretamente, ou, uma bandeira vermelha em caso contrário. Ao finalizar a revisão, o usuário pode prosseguir para a segunda etapa da atividade.

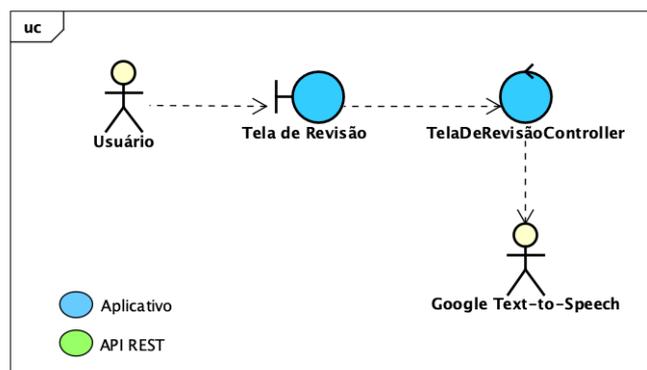
Na Figura 31 é apresentado o detalhamento, e na Figura 32 é apresentado o diagrama de robustez do caso de uso "Fazer Revisão".

Figura 31 – Caso de Uso 5: Fazer Revisão

Versão	1.5	Data Criação	13/11/2020
Autor	Anderson Begossi	Data Atualização	23/05/2021
Descrição	O aplicativo deve permitir a realização de uma revisão durante as atividades de nomeação rápida.		
RF Associados	007		
RNF Associados	001 e 002.		
Atores	Usuário		
Pré-condições	O usuário deve estar autenticado no aplicativo.		
Pós-condições	A tela de conclusão da revisão deve ser visualizada pelo usuário.		
Fluxo principal	<ol style="list-style-type: none"> 1. Usuário pressiona o botão Revisar. 2. Enquanto houver estímulos, repete: <ol style="list-style-type: none"> 4. Aplicativo exibe a tela de revisão do estímulo. 5. Usuário pressiona o botão Próximo. 6. Aplicativo exibe a tela de conclusão da revisão. 		
Fluxos alternativos	<ol style="list-style-type: none"> 5. a) Usuário pressiona o botão de áudio. <ol style="list-style-type: none"> 1. Aplicativo reproduz a palavra correspondente ao estímulo. 		
Requisitos especiais			

Fonte: O Autor (2021).

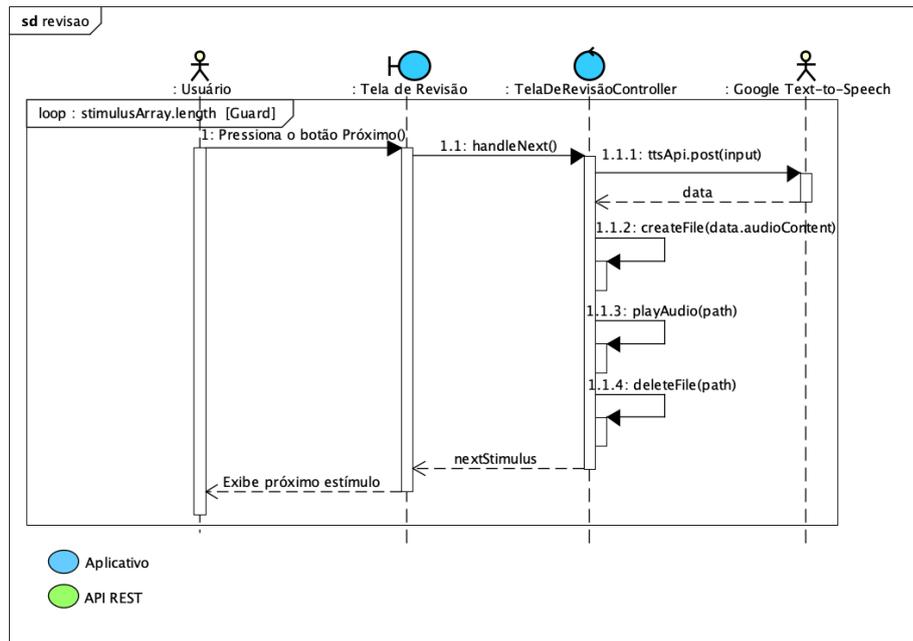
Figura 32 – Diagrama de robustez 5: Fazer Revisão



Fonte: O Autor (2021).

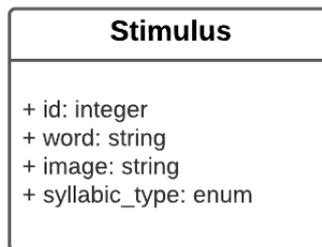
Na Figura 33 é apresentado o diagrama de sequência, e na Figura 34 é apresentado o diagrama de classes.

Figura 33 – Diagrama de sequência 5: Fazer Revisão



Fonte: O Autor (2021).

Figura 34 – Diagrama de classes 5: Fazer Revisão



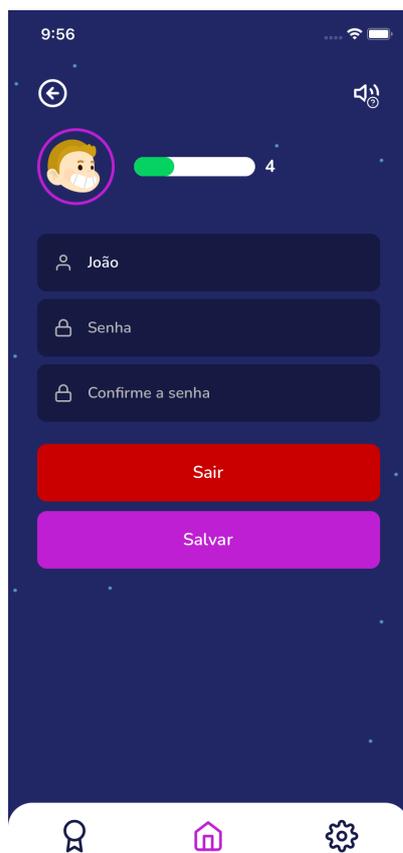
Fonte: O Autor (2021).

Conforme mostra o diagrama acima, somente a classe "Stimulus", referente à um estímulo, é manipulada durante o caso de uso "Fazer Revisão".

5.4.6 Caso de Uso 6 - Editar Perfil

O caso de uso "Editar Perfil" atende aos requisitos funcionais RF-008, RF-009, RF-010 e RF-011. Na Figura 35 é apresentada a respectiva captura de tela.

Figura 35 – Captura de tela 6: Editar Perfil

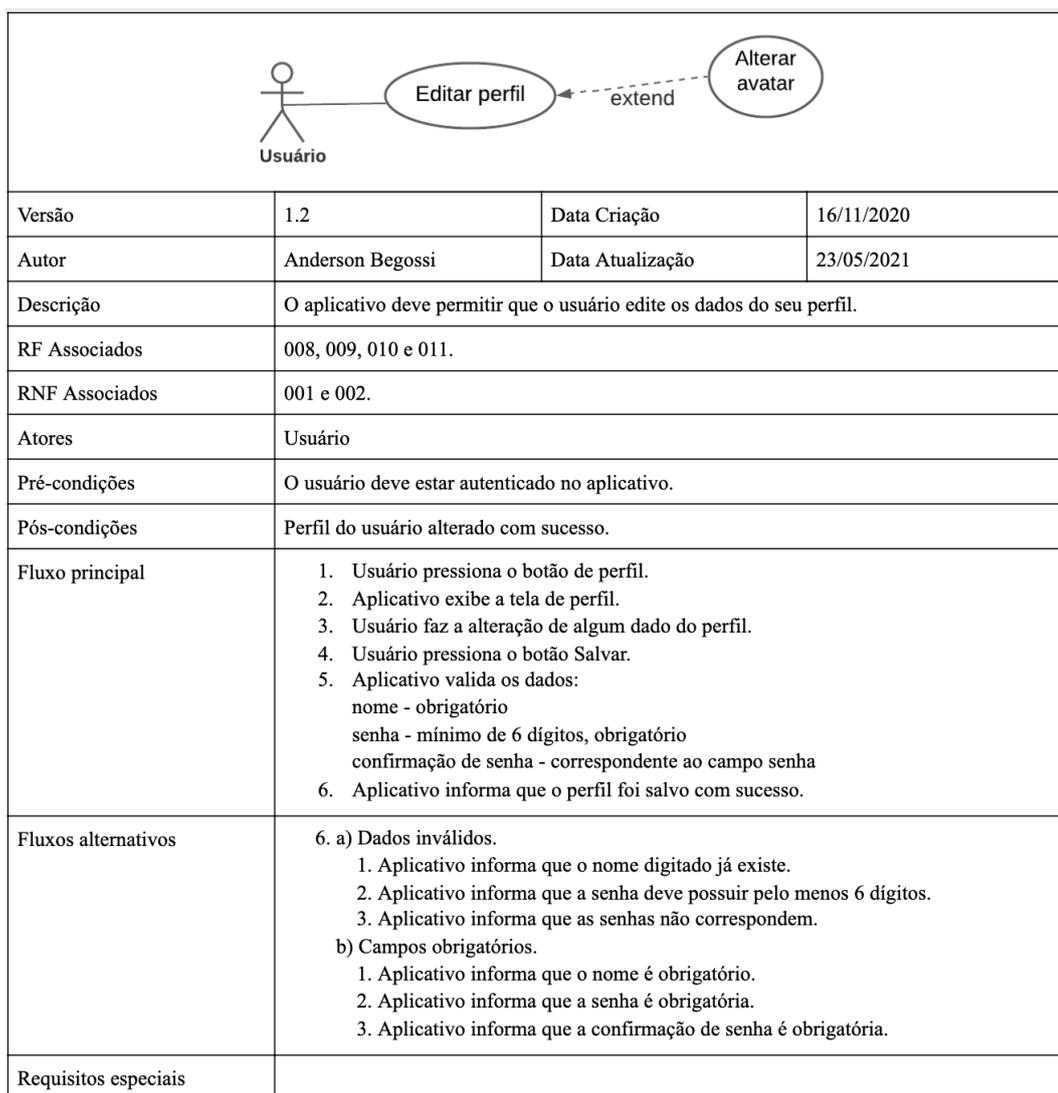


Fonte: O Autor (2021).

Nesta tela, o nome de usuário e a senha poderão ser alterados. A alteração do avatar é feita em outra tela, e corresponde ao caso de uso 7 "Alterar Avatar".

Na Figura 36 é apresentado o detalhamento do caso de uso.

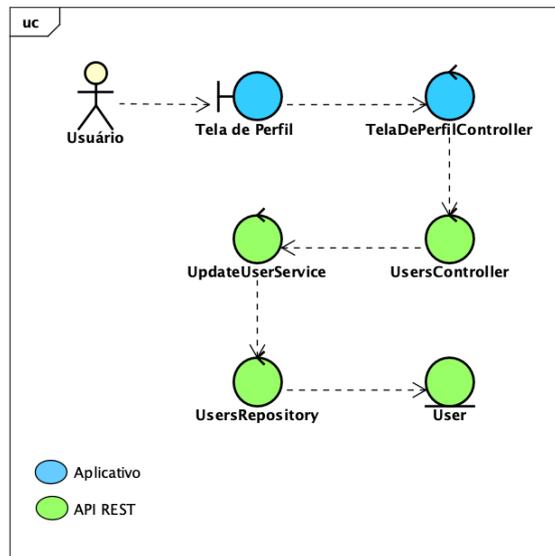
Figura 36 – Caso de Uso 6: Editar Perfil



Fonte: O Autor (2021).

Na Figura 37 é apresentado o diagrama de robustez.

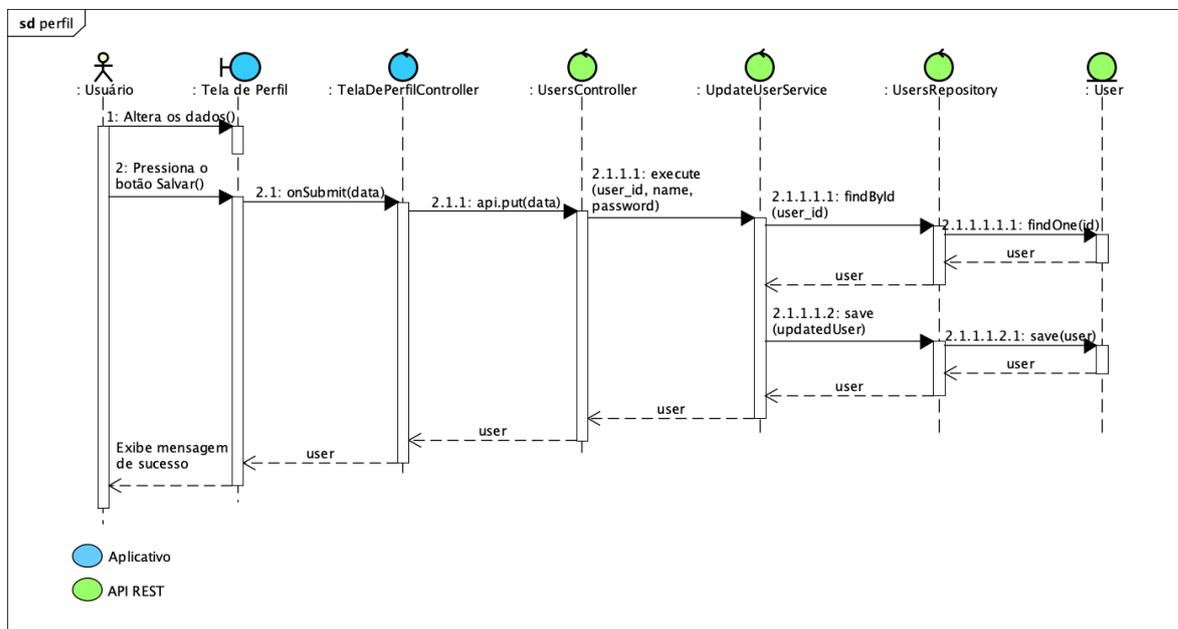
Figura 37 – Diagrama de robustez 6: Editar Perfil



Fonte: O Autor (2021).

Na Figura 38 é apresentado o diagrama de sequência.

Figura 38 – Diagrama de sequência 6: Editar Perfil



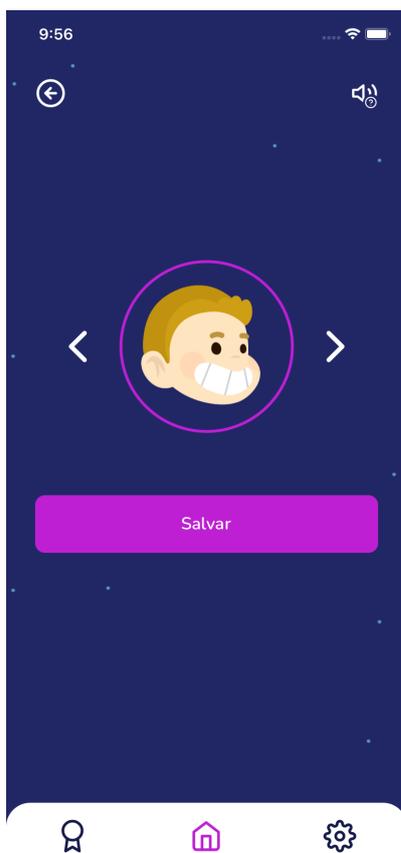
Fonte: O Autor (2021).

O diagrama de classes é o mesmo que o apresentado na Figura 16.

5.4.7 Caso de Uso 7 - Alterar Avatar

O caso de uso "Alterar Avatar" atende ao requisito funcional RF-012. Na Figura 39 é apresentada a captura de tela referente a esse caso de uso.

Figura 39 – Captura de tela 7: Alterar Avatar



Fonte: O Autor (2021).

Conforme mostra a captura, a partir desta tela o usuário pode alterar o avatar de seu perfil. Os avatares disponíveis para a escolha foram retirados de um repositório comunitário de imagens de avatar do *software Figma*¹. Na Figura 40 é apresentado o detalhamento do caso de uso, e na Figura 41 é apresentado o diagrama de robustez.

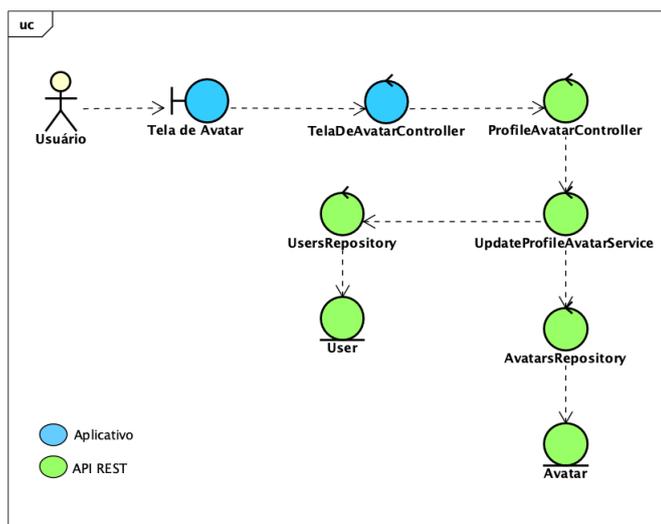
¹ <<https://www.figma.com>>

Figura 40 – Caso de Uso 7: Alterar Avatar

Versão	1.2	Data Criação	16/11/2020
Autor	Anderson Begossi	Data Atualização	23/05/2021
Descrição	O aplicativo deve permitir que o usuário altere o avatar do seu perfil.		
RF Associados	012		
RNF Associados	001 e 002.		
Atores	Usuário		
Pré-condições	O usuário deve estar autenticado no aplicativo.		
Pós-condições	Avatar do usuário alterado com sucesso.		
Fluxo principal	<ol style="list-style-type: none"> 1. Usuário pressiona o avatar na tela de perfil. 2. Aplicativo exibe a tela de alteração de avatar. 3. Usuário escolhe o avatar desejado. 4. Usuário pressiona o botão Salvar. 5. Aplicativo informa que o avatar foi alterado com sucesso. 		
Fluxos alternativos	<ol style="list-style-type: none"> 5. a) Erro ao salvar o avatar. <ol style="list-style-type: none"> 1. Aplicativo informa que houve um erro ao salvar o avatar. 		
Requisitos especiais			

Fonte: O Autor (2021).

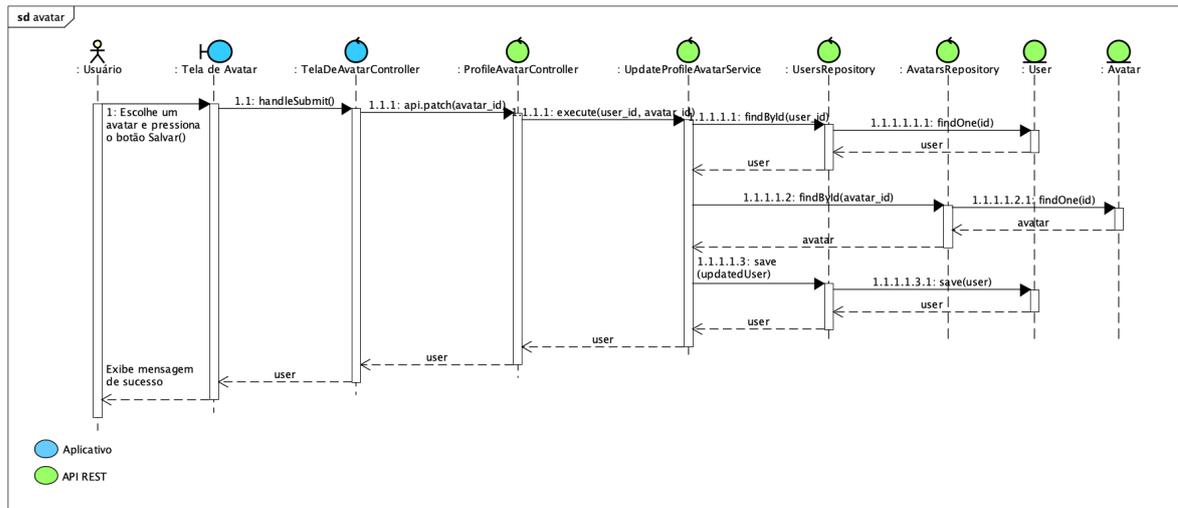
Figura 41 – Diagrama de robustez 7: Alterar Avatar



Fonte: O Autor (2021).

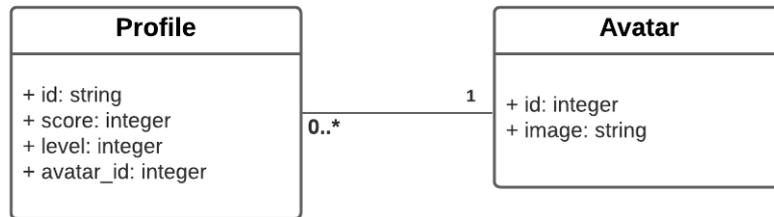
Na Figura 42 é apresentado o diagrama de sequência, e na Figura 43 é apresentado o diagrama de classes.

Figura 42 – Diagrama de sequência 7: Alterar Avatar



Fonte: O Autor (2021).

Figura 43 – Diagrama de classes 7: Alterar Avatar



Fonte: O Autor (2021).

5.4.8 Caso de Uso 8 - Visualizar Ranking

O caso de uso "Visualizar Ranking" atende ao requisito funcional RF-013. Na Figura 44 é apresentado a respectiva captura de tela.

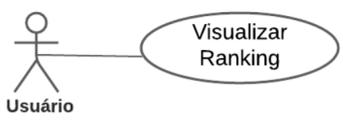
Figura 44 – Captura de tela 8: Visualizar Ranking



Fonte: O Autor (2021).

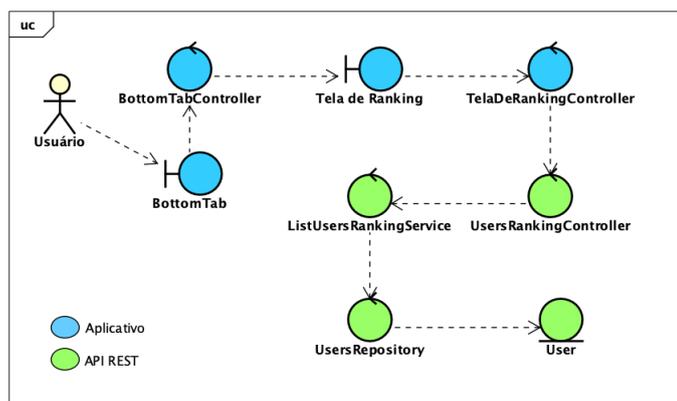
Conforme mostra a captura de tela, o usuário pode visualizar um *ranking* com os usuários do aplicativo, ordenados pelo nível de cada um. Na Figura 45 é apresentado o detalhamento, e na Figura 46 é apresentado o diagrama de robustez do caso de uso "Visualizar Ranking".

Figura 45 – Caso de Uso 8: Visualizar Ranking

			
Versão	1.2	Data Criação	16/11/2020
Autor	Anderson Begossi	Data Atualização	23/05/2021
Descrição	O aplicativo deve permitir que o usuário visualize um ranking do nível dos usuários do aplicativo.		
RF Associados	013		
RNF Associados	001 e 002.		
Atores	Usuário		
Pré-condições	O usuário deve estar autenticado no aplicativo.		
Pós-condições	Ranking visualizado pelo usuário.		
Fluxo principal	<ol style="list-style-type: none"> 1. Usuário pressiona o botão de ranking na barra inferior. 2. Aplicativo exibe a tela de ranking com os usuários do aplicativo, ordenados pelo nível de cada um. 3. Usuário visualiza o ranking. 		
Fluxos alternativos			
Requisitos especiais			

Fonte: O Autor (2021).

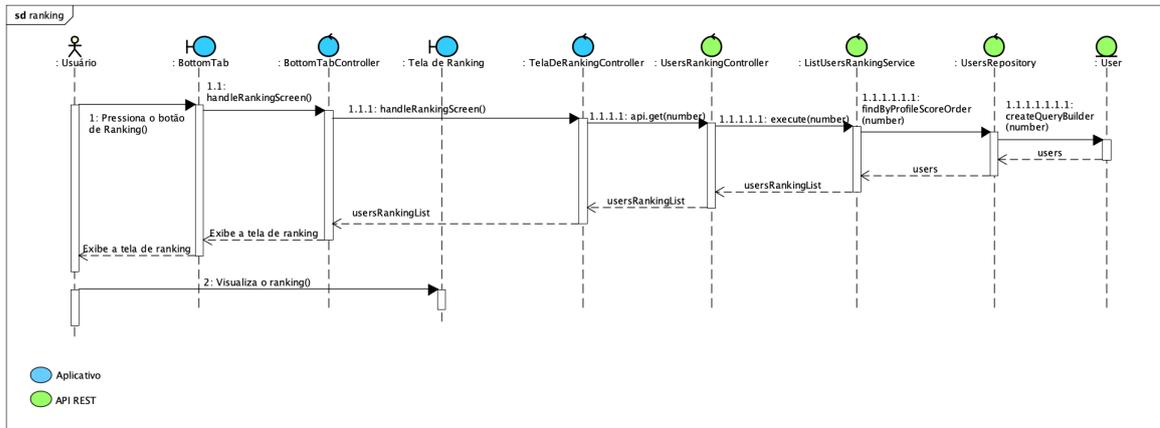
Figura 46 – Diagrama de robustez 8: Visualizar Ranking



Fonte: O Autor (2021).

Na Figura 47 é apresentado o diagrama de sequência.

Figura 47 – Diagrama de sequência 8: Visualizar Ranking



Fonte: O Autor (2021).

O diagrama de classes do caso de uso "Visualizar Ranking" é o mesmo do que o apresentado na Figura 16.

5.4.9 Caso de Uso 9 - Alterar Configurações

O caso de uso "Alterar Configurações" atende aos requisitos funcionais RF-014, RF-015 e RF-016. Na Figura 48, é apresentada a respectiva captura de tela.

Figura 48 – Captura de tela 9: Alterar Configurações



Fonte: O Autor (2021).

Dentre as configurações, o usuário pode alterar a família da fonte do aplicativo, alterar a velocidade de fala das instruções por áudio sintetizadas com o *Text-to-Speech*, e também escolher a privacidade do perfil. A privacidade do perfil define se o perfil será, ou não, mostrado no *ranking* de usuários. A tela de configurações também possui um botão de exclusão de conta, que será representada no caso de uso 10, "Excluir Conta".

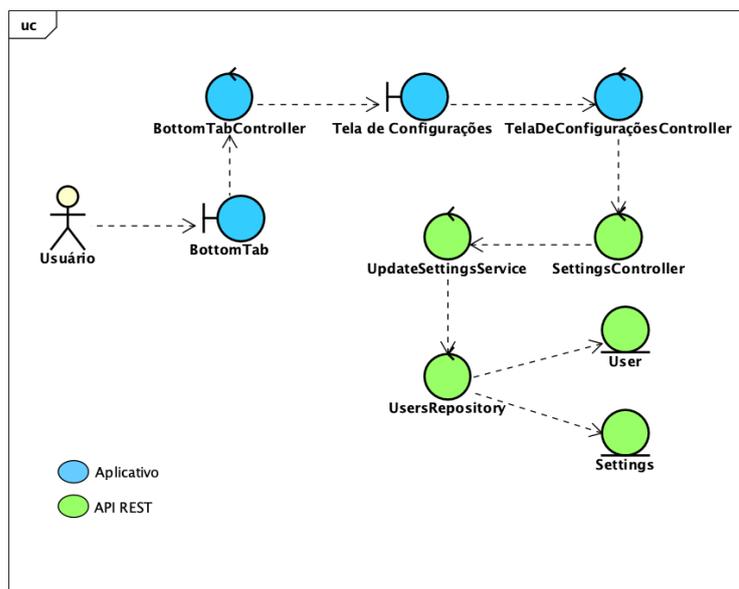
Na Figura 49 é apresentado o detalhamento, e na Figura 50 é apresentado o diagrama de robustez do caso de uso "Alterar Configurações".

Figura 49 – Caso de Uso 9: Alterar Configurações

<p>The diagram shows a stick figure actor labeled 'Usuário' connected to an oval use case labeled 'Alterar configurações'. A dashed arrow labeled 'extend' points from a second oval use case labeled 'Excluir conta' to the 'Alterar configurações' use case.</p>			
Versão	1.5	Data Criação	16/11/2020
Autor	Anderson Begossi	Data Atualização	24/05/2021
Descrição	O aplicativo deve permitir que o usuário altere as configurações do aplicativo.		
RF Associados	014, 015 e 016.		
RNF Associados	001 e 002.		
Atores	Usuário		
Pré-condições	O usuário deve estar autenticado no aplicativo.		
Pós-condições	Usuário vê mensagem dizendo que as configurações foram alteradas com sucesso.		
Fluxo principal	<ol style="list-style-type: none"> 1. Usuário pressiona o botão de configurações. 2. Aplicativo exibe a tela de configurações. 3. Usuário faz a alteração de alguma configuração. 4. Usuário pressiona o botão Salvar. 5. Aplicativo informa que as configurações foram salvas com sucesso. 		
Fluxos alternativos	<ol style="list-style-type: none"> 5. a) Erro ao salvar as configurações. <ol style="list-style-type: none"> 1. Aplicativo exibe mensagem informando que houve um erro ao salvar as configurações. 		
Requisitos especiais			

Fonte: O Autor (2021).

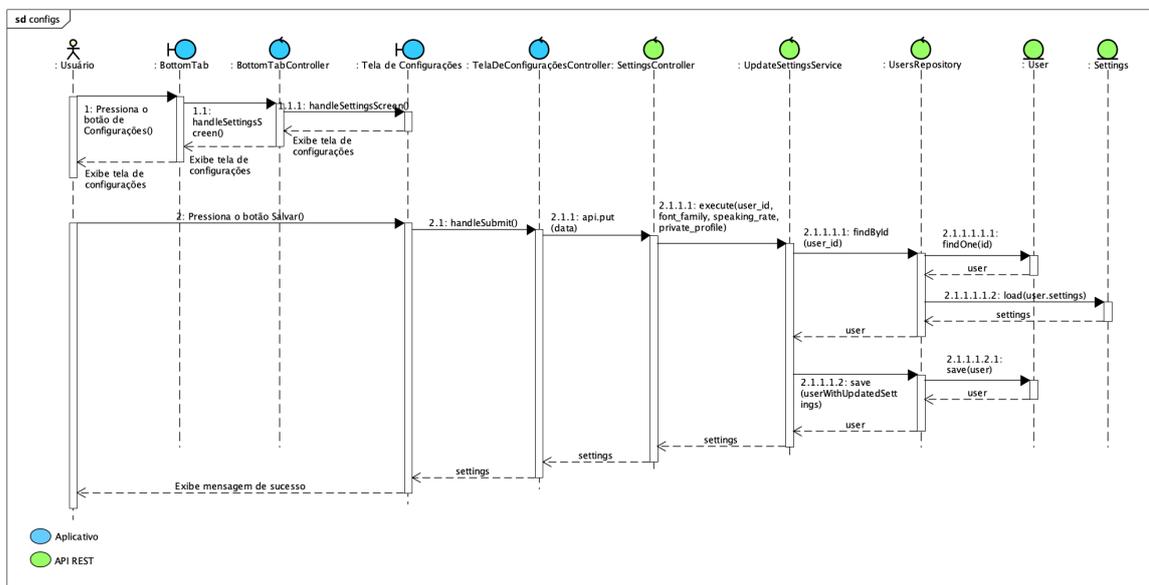
Figura 50 – Diagrama de robustez 9: Alterar Configurações



Fonte: O Autor (2021).

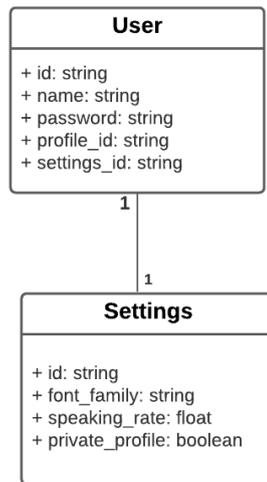
Na Figura 51 é apresentado o diagrama de sequência, e na Figura 52 é apresentado o diagrama de classes.

Figura 51 – Diagrama de sequência 9: Alterar Configurações



Fonte: O Autor (2021).

Figura 52 – Diagrama de classes 9: Alterar Configurações



Fonte: O Autor (2021).

5.4.10 Caso de Uso 10 - Excluir Conta

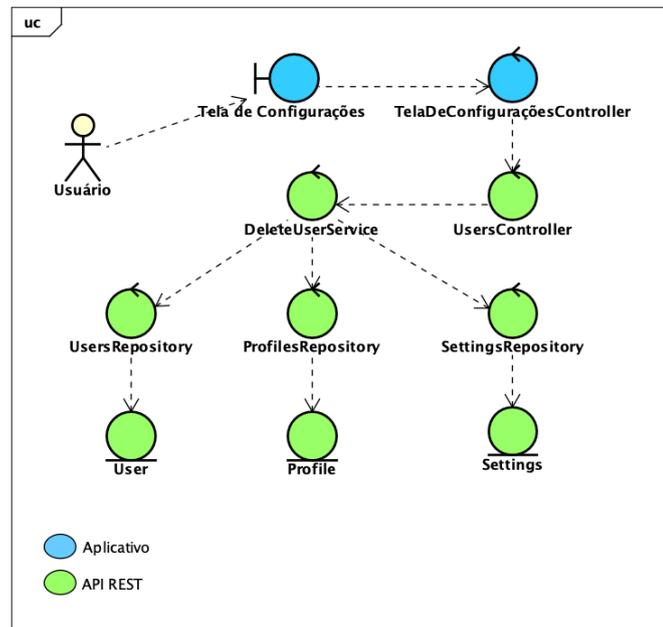
O caso de uso "Excluir Conta" atende ao requisito funcional RF-015. A captura de tela é a mesma que a apresentada na Figura 48, pois, a exclusão de conta é feita através da tela de configurações. Na Figura 53 é apresentado o detalhamento, e na Figura 54 é apresentado o diagrama de robustez do caso de uso "Excluir Conta".

Figura 53 – Caso de Uso 10: Excluir Conta

Versão	1.2	Data Criação	17/11/2020
Autor	Anderson Begossi	Data Atualização	24/05/2021
Descrição	O aplicativo deve permitir que o usuário faça a exclusão de sua conta.		
RF Associados	017		
RNF Associados	001 e 002.		
Atores	Usuário		
Pré-condições	O usuário deve estar autenticado no aplicativo.		
Pós-condições	Conta excluída com sucesso e visualização da tela de login.		
Fluxo principal	<ol style="list-style-type: none"> 1. Usuário pressiona o botão Excluir Conta. 2. Aplicativo pede se o usuário tem certeza que deseja excluir a conta. 3. Usuário confirma. 4. Aplicativo informa que a conta foi excluída com sucesso. 5. Aplicativo redireciona o usuário para a tela de login. 		
Fluxos alternativos	<ol style="list-style-type: none"> 4. a) Erro ao excluir a conta. <ol style="list-style-type: none"> 1. Aplicativo exibe mensagem informando que houve um erro ao excluir a conta. 		
Requisitos especiais			

Fonte: O Autor (2021).

Figura 54 – Diagrama de robustez 10: Excluir Conta



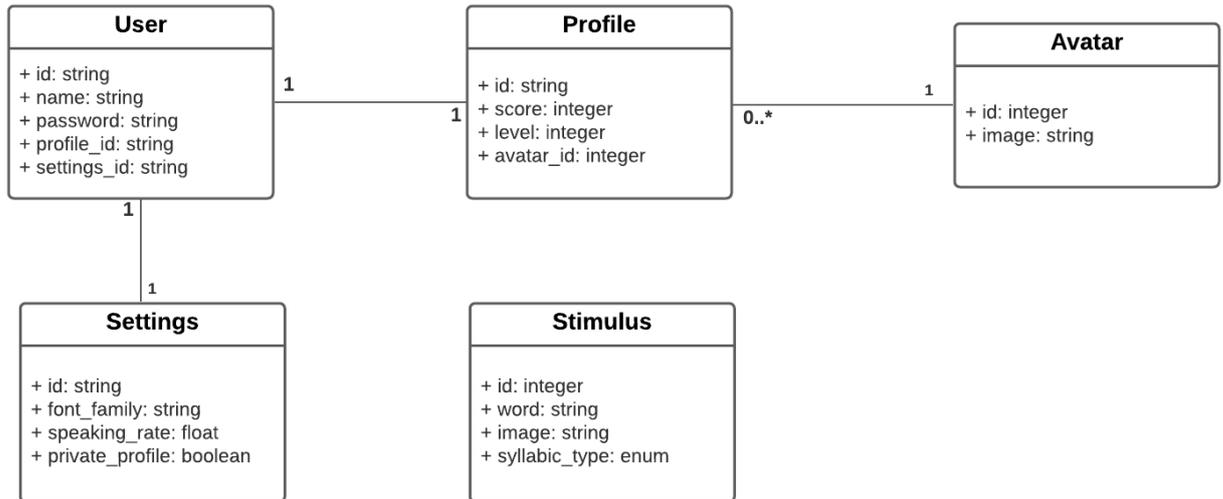
Fonte: O Autor (2021).

Na Figura 55 é apresentado o diagrama de sequência do caso de uso "Excluir Conta".

5.5 DIAGRAMA DE CLASSES

Na Figura 56 é apresentado o diagrama de classes completo do projeto. O diagrama de classes compreende o modelo estático do processo ICONIX, e, foi refinado a partir do modelo de domínio apresentado na seção 5.2.

Figura 56 – Diagrama de classes



Fonte: O Autor (2021).

6 IMPLEMENTAÇÃO

Este capítulo descreve como foi realizada a implementação da aplicação. A Seção 6.1 descreve a configuração do ambiente de desenvolvimento. A Seção 6.2 descreve a etapa de desenvolvimento dos casos de uso. Por fim, a Seção 6.3 descreve o processo de publicação.

6.1 CONFIGURAÇÃO DO AMBIENTE DE DESENVOLVIMENTO

A máquina utilizada durante o desenvolvimento foi um *MacBook Air* (M1, 2020), com 8GB de memória RAM, e sistema operacional *macOS Big Sur*, versão 11.3. O sistema operacional da Apple é necessário para que a compilação do aplicativo para a plataforma *iOS* seja possível.

Primeiramente, foi realizada a instalação do *Node.js*, que proporciona um ambiente de execução *JavaScript* para a implementação da API. Além disso, o *React Native*, *framework* que foi utilizado na construção do aplicativo, também requer o *Node.js* para a compilação de arquivos e execução de processos na máquina utilizada durante o desenvolvimento. A versão do *Node.js* instalada foi a 15.11.0. Em seguida, foi instalado um gerenciador de pacotes para *JavaScript* chamado *Yarn*¹. A versão instalada foi a 1.22.10. Todo o código foi versionado e mantido através das ferramentas *Git* e *GitHub*².

Para o desenvolvimento da API, optou-se pelo uso do *TypeScript* (BIERMAN; ABADI; TORGERSEN, 2014). O *TypeScript* é capaz de adicionar tipagens estáticas ao código *JavaScript*, prevenindo erros em tempo de desenvolvimento e aumentando a robustez da aplicação. Também foi utilizado um *framework* de desenvolvimento de *web services* de *Node.js*, chamado *Express*³. Para a persistência dos dados da aplicação, foi escolhido o banco de dados *PostgreSQL*⁴.

Para a construção do aplicativo utilizando o *React Native*, foi instalada a interface de linha de comando *react-native-cli*⁵. Além disso, para o *iOS* foi necessária a instalação da ferramenta *Xcode*⁶, e, para o *Android*, o *Java Development Kit* 11.0.10 e o *Android Studio*⁷.

6.2 DESENVOLVIMENTO DOS CASOS DE USO

O processo de desenvolvimento seguiu o fluxo apresentado na Figura 57, baseado nos casos de uso descritos no capítulo anterior.

¹ <<https://yarnpkg.com/>>

² <<https://github.com/>>

³ <<https://expressjs.com>>

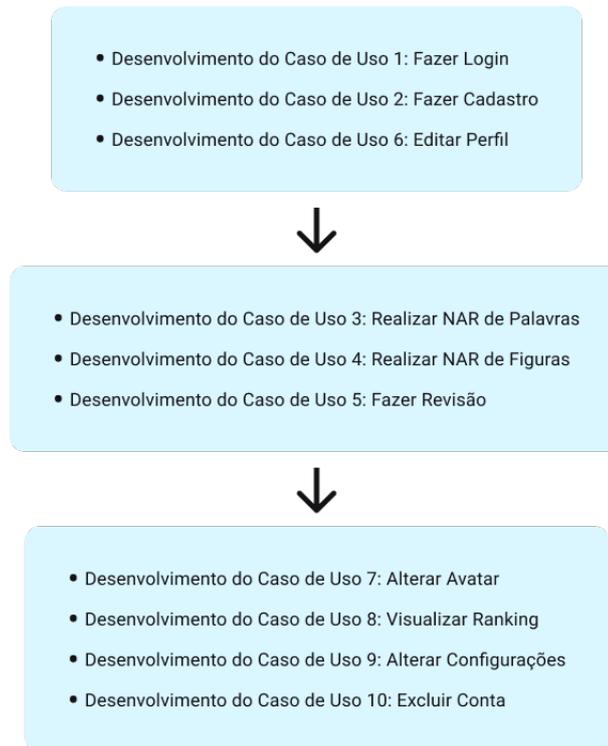
⁴ <<https://www.postgresql.org/>>

⁵ <<https://www.npmjs.com/package/react-native-cli>>

⁶ <<https://developer.apple.com/xcode/>>

⁷ <<https://developer.android.com/studio>>

Figura 57 – Etapas de desenvolvimento dos casos de uso



Fonte: O Autor (2021).

Como pode ser observado, o desenvolvimento foi dividido em 3 etapas. Primeiramente, optou-se pela implementação dos casos de uso que constituem a base do aplicativo, possibilitando que os subsequentes fossem implementados sem nenhum tipo de impedimento. Todas as funcionalidades foram implementadas inicialmente na API. Em seguida, o desenvolvimento foi realizado no aplicativo.

As funcionalidades criadas na API são disponibilizadas através de *endpoints* públicos, que são pontos de acesso de comunicação à uma aplicação. Desta forma, o aplicativo realiza requisições HTTP para solicitar que a API realize um procedimento, ou, obter dados que são necessários para realizar alguma tarefa solicitada pelo usuário. Como resultado do desenvolvimento dos casos de uso da primeira etapa, foram criados os *endpoints* mostrados no Quadro 6.

Quadro 6 – *Endpoints* disponibilizados na primeira etapa

Método	Endpoint	Descrição
POST	/sessions	Realiza a autenticação de um usuário.
POST	/users	Realiza o cadastro de um usuário.
PUT	/users	Realiza a edição de um usuário.
GET	/profile/:profile_id	Retorna um usuário.

Fonte: O Autor (2021).

A autenticação utiliza o padrão RFC 7519, chamado de *JSON Web Token* (IETF, 2015). Após a requisição do aplicativo, se a autenticação for bem sucedida, é retornado um token

JWT, armazenado no aplicativo, e utilizado para autorizar o acesso aos *endpoints* privados da aplicação. No aplicativo, o resultado foi a criação das telas apresentadas nas Figuras 12, 17 e 35, no capítulo anterior. A sequência de interações entre o aplicativo e a API é mostrada nos diagramas de sequência, também apresentados no capítulo anterior.

Na segunda etapa, realizou-se o desenvolvimento das principais funcionalidades do aplicativo, sendo a nomeação de palavras, a nomeação de figuras e a revisão. Primeiramente, foram desenvolvidos *endpoints* na API para a inclusão dos estímulos na base de dados. Após a codificação das classes e controladores relacionados, observou-se a necessidade da integração com um serviço de armazenamento de objetos em nuvem, para armazenar a imagem de cada estímulo adicionado, e, posteriormente, as imagens de avatar. A integração é conveniente devido os seguintes motivos:

1. Armazenar tais imagens no *bundle* do aplicativo aumentaria o tamanho do mesmo, demandando mais espaço no dispositivo do usuário.
2. O custo do armazenamento do banco de dados em serviços de hospedagem geralmente é mais caro do que o de serviços específicos para isso.
3. Armazenar as imagens no banco de dados aumentaria a latência das requisições para a obtenção dos estímulos.
4. A inclusão, remoção e edição de estímulos e avatares pode ser feita a qualquer momento, sem a necessidade de atualizar o aplicativo do usuário.

Optou-se então pela integração com o serviço de armazenamento de objetos AWS S3⁸, oferecido pela empresa Amazon. Após a conclusão, foram disponibilizados os *endpoints* mostrados no Quadro 7, e incluídos 80 estímulos na base de dados.

Quadro 7 – *Endpoints* relacionados à criação e listagem de estímulos

Método	Endpoint	Descrição
POST	/stimulus	Realiza a inserção de um estímulo no banco de dados.
GET	/stimulus	Realiza a listagem de todos os estímulos cadastrados no banco de dados.
GET	/stimulus/:profile_id	Retorna uma lista de estímulos baseado no nível do perfil do usuário.

Fonte: O Autor (2021).

A utilização do *endpoint* que devolve ao aplicativo uma lista de estímulos pode ser observada no diagrama de sequência apresentado na Figura 24. Caso o nível do perfil for menor ou igual a 5, só são retornados estímulos monossilábicos. Caso o nível for maior ou igual a 6, e, menor ou igual a 19, podem ser retornados estímulos monossilábicos e dissilábicos. Caso o nível for maior ou igual a 20, e, menor ou igual a 29, podem ser retornados estímulos monossilábicos, dissilábicos e trissilábicos. Caso o nível for maior ou igual a 30, podem ser retornados

⁸ <<https://aws.amazon.com>>

estímulos monossilábicos, dissilábicos, trissilábicos e polissilábicos. Desta forma, a dificuldade de nomeação é incrementalmente aumentada conforme o progresso do usuário.

Durante a implementação das funcionalidades de nomeação no aplicativo, foi encontrado um empecilho decorrente do serviço escolhido para fazer o reconhecimento da nomeação feita pelo usuário, o *Speech-to-Text*. Tal serviço requer que o arquivo de áudio a ser transcrito esteja em formatos e codificações específicas, não sendo trivial a conversão para um *codec* suportado no dispositivo do usuário. Desta forma, optou-se pela integração do *Speech-to-Text* que inicialmente seria feita no aplicativo, ser feita na API, juntamente com a conversão do arquivo de áudio para um formato suportado. Na Figura 58, os *codecs* suportados pelo *Speech-to-Text* são mostrados. No caso do presente trabalho, foi escolhida a conversão para o *codec* LINEAR16, que não apresenta perda de qualidade durante a compressão do arquivo.

Figura 58 – *Codecs* suportados pelo *Speech-to-Text*

Codec	Nome	Sem perdas	Notas de uso
MP3	Camada de áudio MPEG III	Não	A codificação MP3 é um recurso Beta e está disponível apenas na v1p1beta1. Consulte a documentação de referência de RecognitionConfig para mais detalhes.
FLAC	Codec de áudio livre sem perdas	Sim	16 bits ou 24 bits obrigatórios para o stream
LINEAR16	Modulação por código de pulso linear	Sim	Codificação de modulação de código de pulso linear de 16 bits (PCM)
MULAW	μ -law	Não	Codificação PCM de 8 bits
AMR	Adaptive Multi-Rate banda estreita	Não	A taxa de amostra precisa ser 8.000 Hz
AMR_WB	Adaptive Multi-Rate banda larga	Não	A taxa de amostragem precisa ser 16.000 Hz
OGG_OPUS	Frames de áudio codificados Opus em um contêiner Ogg	Não	A taxa de amostragem precisa ser de 8.000 Hz, 12.000 Hz, 16.000 Hz, 24.000 Hz ou 48.000 Hz
SPEEX_WITH_HEADER_BYTE	Speex banda larga	Não	A taxa de amostragem precisa ser 16.000 Hz
WEBM_OPUS	Opus WebM	Não	

Fonte: GOOGLE (2021).

O fluxo que ocorre quando o usuário realiza a nomeação de um estímulo pode ser observado com detalhes no diagrama de sequência apresentado na Figura 25. Para a conversão do arquivo de áudio, optou-se pela integração com a biblioteca FFmpeg⁹. No Algoritmo 1 é mostrado o código da função do aplicativo que realiza a requisição para a API realizar o reconhecimento do estímulo nomeado.

⁹ <<https://www.ffmpeg.org/>>

Algoritmo 1 – Função do aplicativo que solicita o reconhecimento do áudio

```
1  const handleRecognizeAudio = async path => {
2    const formData = new FormData();
3
4    formData.append('audio', {
5      type: 'audio/aac',
6      name: 'audio.aac',
7      uri: 'file://${path}',
8    });
9
10   formData.append('word', currentStimulus.word);
11
12   try {
13     const { data } = await api.post('/stimulus/recognize', formData);
14
15     const result = {
16       stimulus: currentStimulus,
17       recognized: data.recognized,
18       isCorrect: data.recognition ? data.recognition.isCorrect : false,
19     };
20
21     addNewResult(result);
22   } catch (err) {
23     handleError(err);
24   }
25
26   await deleteFile(path);
27 }
```

Fonte: O Autor (2021).

Como pode ser visto, a função recebe o caminho do arquivo de áudio, cria um objeto FormData adicionando os dados do áudio, e o campo "word" que contém a palavra que representa o estímulo que o usuário tentou nomear. Após isso, o aplicativo realiza uma requisição HTTP do tipo POST para o *endpoint* "/stimulus/recognize" da API, enviando o FormData. Com a resposta obtida, é criado um objeto chamado "result", contendo o estímulo nomeado, um valor chamado "recognized" que informa se o reconhecimento foi feito, e outro valor chamado "isCorrect", que informa se a nomeação está correta. Em seguida, o resultado é adicionado a uma lista de resultados, e o arquivo de áudio é deletado. Também pode ser observado que o arquivo de áudio gravado tem o formato "aac". O formato *Advanced Audio Coding* (AAC) foi utilizado pois é suportado tanto no *Android*, como, no *iOS*, não necessitando de tratamentos especiais para cada plataforma.

No Algoritmo 2, é possível observar o trecho de código do *SpeechRecognitionProvider*, da API, que realiza a requisição para o *Speech-to-Text* transcrever o arquivo de áudio, já

convertido pela biblioteca FFmpeg.

Algoritmo 2 – Trecho de código do *SpeechRecognitionProvider*

```
1  const audioContent = await fs.promises.readFile(originalPath);
2
3  const audioBytes = audioContent.toString('base64');
4
5  const audioFile = {
6    content: audioBytes,
7  };
8
9  const config = speechRecognitionConfig.config.google.recognitionConfig;
10
11  config.speechContexts = [
12    {
13      phrases: [expectedTranscript],
14    },
15  ];
16
17  const request = {
18    audio: audioFile,
19    config,
20  };
21
22  const [response] = await this.client.recognize(request);
23
24  const { results } = response;
25
26  await fs.promises.unlink(originalPath);
27
28  if (results.length === 0) {
29    return null;
30  }
31
32  const transcription = results[0].alternatives[0].transcript;
33
34  return transcription;
```

Fonte: O Autor (2021).

No trecho acima, também é possível notar que o arquivo de áudio é codificado utilizando o método Base64, antes de ser enviado para o serviço de reconhecimento.

Como já mencionado no capítulo anterior, para a funcionalidade de revisão foi realizada a integração com o serviço de sintetização de voz do Google, o *Text-to-Speech*. Essa integração pôde ser feita somente no aplicativo, sem a necessidade da comunicação com a API desenvolvida. O serviço é usado para reproduzir ao usuário a pronúncia da palavra que corresponde a

cada estímulo que é mostrado na tela, durante a etapa de revisão. As telas resultantes podem ser vistas nas Figuras 21, 28 e 30.

Por fim, foi realizada a implementação dos casos de uso faltantes, relacionados à alteração do avatar do usuário, ao *ranking* de usuários, à alteração de configurações e à exclusão da conta do usuário. Nessa etapa, foi feita a inclusão das imagens de avatar no serviço AWS S3. Os *endpoints* resultantes da implementação dessas funcionalidades são mostrados no Quadro 8.

Quadro 8 – *Endpoints* disponibilizados na terceira etapa

Método	Endpoint	Descrição
POST	/avatars	Realiza a inclusão de um avatar.
GET	/avatars	Lista todos os avatares cadastrados no banco de dados.
PATCH	/profile/avatar	Realiza a alteração do avatar de um usuário.
GET	/ranking	Retorna uma lista de usuários, ordenada pelo nível do perfil de cada usuário.
PUT	/settings	Realiza a edição das configurações de um usuário.
DELETE	/users	Realiza a deleção de um usuário.

Fonte: O Autor (2021).

As telas do aplicativo desenvolvidas nesta etapa podem ser vistas nas Figuras 39, 44 e 48.

6.3 PUBLICAÇÃO

Com a finalização do desenvolvimento, iniciou-se a etapa de publicação, com a finalidade de disponibilizar a ferramenta para testes com usuários. Para a hospedagem da API, foi utilizado a plataforma de serviços em nuvem DigitalOcean¹⁰. O servidor contratado foi configurado, e todas as dependências necessárias para manter a API executando foram instaladas. As especificações do servidor são mostradas no Quadro 9.

Quadro 9 – Especificações do servidor

CPU	Memória RAM	Disco	Sistema Operacional	Região
Intel/1vCPU,	1GB	25GB SSD	Ubuntu 20.04	Nova Iorque, Estados Unidos

Fonte: O Autor (2021).

Segundo a DigitalOcean, uma vCPU é uma unidade de processamento correspondente a uma única *hyperthread* em um núcleo de processador (DIGITALOCEAN, 2021).

Também foi necessária a aquisição de um domínio, para que a API fosse disponibilizada em um endereço público acessível pelo aplicativo. O domínio adquirido foi o "dislexi-app.com", através da plataforma *Google Domains*¹¹. Após isso, a API foi publicada no endereço "https://api.dislexiapp.com".

¹⁰ <https://www.digitalocean.com/>

¹¹ <https://domains.google/>

Para o aplicativo, foi realizado o processo de envio para a loja de aplicativos *Google Play Store*¹². O processo demandou diversos esclarecimentos sobre o conteúdo do aplicativo, público alvo, anúncios, além da elaboração de uma política de privacidade, apresentada na Figura 59 (Anexo A), e disponibilizada no endereço "<https://ambegossi.github.io/dislexiapp-privacidade>". Apesar do aplicativo também estar pronto para o envio para a *App Store*¹³, o processo não foi realizado devido ao custo e o tempo necessário para tal.

Embora o aplicativo tenha sido submetido ao processo de publicação na *Google Play Store*, até a data de entrega do presente trabalho, o aplicativo ainda está sendo analisado pela plataforma. Durante a submissão, foi possível observar um aviso informando que o processo de revisão se encontrava mais longo que o normal, devido a ajustes na escala de trabalho do *Google*.

Idealmente, após publicado na loja, o aplicativo seria disponibilizado para um grupo de pais de crianças disléxicas através do Instituto ABCD, instituto referência em dislexia no Brasil, já citado no Capítulo 2. Contudo, por conta da demora no processo de revisão e do tempo disponível para a conclusão deste trabalho, o aplicativo teve de ser disponibilizado no formato APK.

Também foi elaborado e disponibilizado um formulário de avaliação do aplicativo, apresentado na Figura 60 (Anexo B). O formulário foi elaborado através da ferramenta *Google Forms*¹⁴. Até a data de entrega deste trabalho, não foram obtidas respostas.

¹² <<https://play.google.com/>>

¹³ <<https://www.apple.com/br/app-store/>>

¹⁴ <<https://www.google.com/intl/pt-BR/forms/about/>>

7 CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvido um protótipo de aplicativo para dispositivos móveis, para o auxílio de crianças com dislexia. O programa de remediação de Santos e Capellini (2020) foi utilizado como inspiração para a criação das principais funcionalidades do aplicativo. As fundamentações sobre o transtorno, sobre a gamificação e as tecnologias de desenvolvimento, foram essenciais para a idealização do aplicativo.

O projeto da aplicação foi criado seguindo as diretrizes do processo de desenvolvimento de *software* ICONIX. Os artefatos criados, como, protótipos de tela, diagrama de casos de uso, diagramas de sequência e diagramas de robustez, serviram como base documentativa e guiaram a etapa de implementação. As tecnologias escolhidas se mostraram adequadas e nenhuma limitação foi encontrada. A abordagem de desenvolvimento *mobile* híbrida possibilitou que o aplicativo fosse criado para as plataformas *Android* e *iOS* com somente uma linguagem de programação, o que não seria possível através da abordagem nativa. Além disso, esse fator contribuiu para que o desenvolvimento pudesse ser realizado no tempo previsto, e facilitará a manutenção do código futuramente.

Os serviços de reconhecimento e síntese de voz *Speech-to-Text* e *Text-to-Speech*, utilizados no aplicativo, apresentaram resultados satisfatórios. Contudo, uma melhoria futura que pode ser feita é a substituição dos áudios de instruções, que são sintetizados com o *Text-to-Speech*, por instruções narradas por uma voz humana. Apesar de não comprometer a utilização do aplicativo, a voz que é sintetizada pelo serviço possui aspectos robotizados. Nessa mesma linha, outra melhoria é a utilização de *cache* nas requisições do aplicativo para a obtenção das imagens dos estímulos e dos avatares, diminuindo o tempo de espera do usuário. A API criada também pode ser aprimorada, realizando o desenvolvimento e a disponibilização de *endpoints* para a edição e exclusão de estímulos e avatares da base de dados.

Não foram encontradas dificuldades na etapa de disponibilização da API. Na publicação do aplicativo, o processo acabou tomando mais tempo do que o esperado, contribuindo para que a ferramenta não pudesse ser disponibilizada para os usuários através da loja de aplicativos, o que é mais familiar e convencional. Trabalhos futuros podem ser realizados na direção de validar a ferramenta, acompanhar a interação dos usuários com a mesma, verificar os pontos positivos e negativos, e realizar melhorias de acordo com suas opiniões.

Por fim, considera-se que os objetivos do trabalho foram cumpridos, e a ferramenta desenvolvida possui a capacidade de se tornar uma alternativa digital de apoio para crianças disléxicas.

REFERÊNCIAS

- ABCD, I. **Fichas de Atividades - Ouvir e falar melhor**. 2020. Disponível em: <<https://www.institutoabcd.org.br/fichas-de-atividade/>>.
- ADAMS, M. J. **Consciência fonológica em crianças pequenas**. São Paulo: Artmed Editora, 2018.
- ANDERSON, N. J. **Getting Started with NativeScript**. Birmingham: Packt Publishing Ltd, 2016.
- APA. **DSM-5: Manual diagnóstico e estatístico de transtornos mentais**. Porto Alegre: Artmed Editora, 2014.
- AXELSSON, O.; CARLSTRÖM, F. Evaluation targeting react native in comparison to native mobile development. 2016.
- BIERMAN, G.; ABADI, M.; TORGERSEN, M. Understanding typescript. In: SPRINGER. **European Conference on Object-Oriented Programming**. [S.l.], 2014. p. 257–281.
- BRAGGIO, M. A. **A inclusão do disléxico na escola**. 2012.
- BRASIL, A. **Brasil é o 3º país em que pessoas passam mais tempo em aplicativos**. 2020. Disponível em: <<https://agenciabrasil.ebc.com.br/geral/noticia/2020-01/brasil-e-o-3o-pais-em-que-pessoas-passam-mais-tempo-em-aplicativos>>.
- BRITO, H. *et al.* Javascript in mobile applications: React native vs ionic vs nativescript vs native development. In: IEEE. **2018 13th Iberian Conference on Information Systems and Technologies (CISTI)**. [S.l.], 2018. p. 1–6.
- CADIN, B. **Dislexia - Diagnóstico e Intervenção**. 2020. Disponível em: <<https://www.cadin.net/saber-mais-dislexia/154-diagnostico-e-intervencao>>.
- CAPELLINI, S. A.; CONRADO, T. L. B. C. Desempenho de escolares com e sem dificuldades de aprendizagem de ensino particular em habilidade fonológica, nomeação rápida, leitura e escrita. **Revista Cefac**, SciELO Brasil, v. 11, p. 183–193, 2009.
- COLLINS, D. W.; ROURKE, B. P. Learning-disabled brains: A review of the literature. **Journal of Clinical and Experimental Neuropsychology**, Taylor & Francis, v. 25, n. 7, p. 1011–1034, 2003.
- DAUD, S. M.; ABAS, H. 'dyslexia baca' mobile app—the learning ecosystem for dyslexic children. In: IEEE. **2013 International Conference on Advanced Computer Science Applications and Technologies**. [S.l.], 2013. p. 412–416.
- DENCKLA, M. B.; RUDEL, R. G. Rapid 'automatized' naming (ran): Dyslexia differentiated from other learning disabilities. **Neuropsychologia**, Elsevier, v. 14, n. 4, p. 471–479, 1976.
- DENNY, P. The effect of virtual achievements on student engagement. In: **Proceedings of the SIGCHI conference on human factors in computing systems**. [S.l.: s.n.], 2013. p. 763–772.

DETERDING, S. *et al.* From game design elements to gamefulness: defining "gamification". In: **Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments**. [S.l.: s.n.], 2011. p. 9–15.

DIGITALOCEAN. **DigitalOcean, Choosing the Right Droplet Plan**. 2021. Disponível em: <<https://docs.digitalocean.com/products/droplets/resources/choose-plan/>>.

EISENMAN, B. **Learning react native: Building native mobile apps with javascript**. Sebastopol: O'Reilly Media, Inc., 2015.

FIELDING, R. Representational state transfer. **Architectural Styles and the Design of Network-based Software Architecture**, p. 76–85, 2000.

FIGZ, J. A.; PETERSEN, S. E. Neuroimaging studies of word reading. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 95, n. 3, p. 914–921, 1998.

FRANCESCHINI, S. *et al.* Action video games make dyslexic children read better. **Current Biology**, Elsevier, v. 23, n. 6, p. 462–466, 2013.

GOOCH, D.; VASALOU, A.; BENTON, L. Exploring the use of a gamification platform to support students with dyslexia. In: IEEE. **2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)**. [S.l.], 2015. p. 1–6.

GOOGLE. **Google Speech-to-Text**. 2021. Disponível em: <<https://cloud.google.com/speech-to-text>>.

HANSSON, N.; VIDHALL, T. **Effects on performance and usability for cross-platform application development using react native**. 2016.

HERMES, D. **Xamarin Mobile Application Development: Cross-platform c# and xamarin.forms fundamentals**. Nova Iorque: Apress, 2015.

IBGE. **PNAD Contínua TIC 2018: Internet chega a 79,1% dos domicílios do país**. 2020. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/27515-pnad-continua-tic-2018-internet-chega-a-79-1-dos-domicilios-do-pais>>.

IDA. Dyslexia basics. **Just the facts: Information provided by the International Dyslexia Association**, 2012.

INTERNET ENGINEERING TASK FORCE (IETF). **RFC 7519: Json web token (jwt)**. [S.l.], 2015. 1 p.

JUANG, B.-H.; RABINER, L. R. Automatic speech recognition—a brief history of the technology development. **Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara**, v. 1, p. 67, 2005.

JUNIOR, D. S. *et al.* Versão digital do teste de nomeação automática rápida (ran): uma contribuição para detectar precocemente problemas de leitura em crianças. **Revista CEFAC**, SciELO Brasil, v. 21, n. 1, 2019.

KEARNS, D. M. *et al.* The neurobiology of dyslexia. **TEACHING Exceptional Children**, SAGE Publications Sage CA: Los Angeles, CA, v. 51, n. 3, p. 175–188, 2019.

- LARMAN, C. **Agile and iterative development: a manager's guide**. Boston: Addison-Wesley Professional, 2004.
- LERNER, J. W. **Learning disabilities: Theories, diagnosis, and teaching strategies**. Boston: Houghton Mifflin Harcourt (HMH), 1989.
- LYON, G. R. Toward a definition of dyslexia. **Annals of dyslexia**, Springer, v. 45, n. 1, p. 1–27, 1995.
- MASSE, M. **REST API Design Rulebook: Designing consistent restful web service interfaces**. Sebastopol: O'Reilly Media, Inc., 2011.
- MEIER, R. **Professional Android 4 application development**. Nova Iorque: John Wiley & Sons, 2012.
- MONTAN, J. dos S.; SANTOS, M. C. Avaliação de plataformas híbridas para desenvolvimento de aplicações para dispositivos móveis. **Seminários de Trabalho de Conclusão de Curso do Bacharelado em Sistemas de Informação**, v. 2, n. 1, 2018.
- MOUSINHO, R. *et al.* **Leitura, Escrita e Matemática: do desenvolvimento aos transtornos específicos da aprendizagem**. [S.l.]: ABCD, 2020.
- PRADO, C.; DUBOIS, M.; VALDOIS, S. The eye movements of dyslexic children during reading and visual search: impact of the visual attention span. **Vision research**, Elsevier, v. 47, n. 19, p. 2521–2530, 2007.
- REBOUÇAS, M. *et al.* An empirical study on the usage of the swift programming language. In: IEEE. **2016 IEEE 23rd international conference on software analysis, evolution, and reengineering (SANER)**. [S.l.], 2016. v. 1, p. 634–638.
- RICHLAN, F. Developmental dyslexia: dysfunction of a left hemisphere reading network. **Frontiers in human neuroscience**, Frontiers, v. 6, p. 120, 2012.
- ROSENBERG, D.; STEPHENS, M.; COLLINS-COPE, M. Agile development with iconix process. **New York, Editorial Apress**, Springer, 2005.
- ROTTA, N. T.; OHLWEILER, L.; RIESGO, R. dos S. **Transtornos da aprendizagem: abordagem neurobiológica e multidisciplinar**. Porto Alegre: Artmed Editora, 2015.
- RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. The unified modeling language. **Reference manual**, Addison Wesley Longman Inc., 1999.
- SANTOS, B. d.; CAPELLINI, S. A. **ProNAR-LE - Programa de Remediação com a Nomeação Automática Rápida e Leitura**. Ribeirão Preto: [s.n.], 2018.
- _____. Programa de remediação com a nomeação rápida e leitura para escolares com dislexia: elaboração e significância clínica. **SciELO Brasil**, v. 32, n. 3, 2020.
- SKIADA, R. *et al.* Easylexia: A mobile application for children with learning difficulties. **Procedia Computer Science**, Elsevier, v. 27, p. 218–228, 2014.
- STATCOUNTER. **Mobile Operating System Market Share Worldwide**. 2020. Disponível em: <<https://gs.statcounter.com/os-market-share/mobile/worldwide>>.

TEIXEIRA, P. **Professional Node.js**: Building javascript based scalable software. Indianápolis: John Wiley & Sons, 2012.

TEMPLE, E. *et al.* Disrupted neural responses to phonological and orthographic processing in dyslexic children: an fmri study. **Neuroreport**, LWW, v. 12, n. 2, p. 299–307, 2001.

TIME, P. M. Panorama mobile time/opinion box. **Pesquisas independentes sobre conteúdo e serviços móveis**. [Internet], 2019.

UNDERSTOOD. **Types of Dyslexia: What Researchers Are Studying and Why**. 2020. Disponível em: <<https://www.understood.org/en/learning-thinking-differences/child-learning-disabilities/dyslexia/different-types-of-dyslexia>>.

WHITAKER, H. A. **Concise encyclopedia of brain and language**. Oxford: Elsevier, 2010.

WOLFF, U. Ran as a predictor of reading skills, and vice versa: Results from a randomised reading intervention. **Annals of Dyslexia**, Springer, v. 64, n. 2, p. 151–165, 2014.

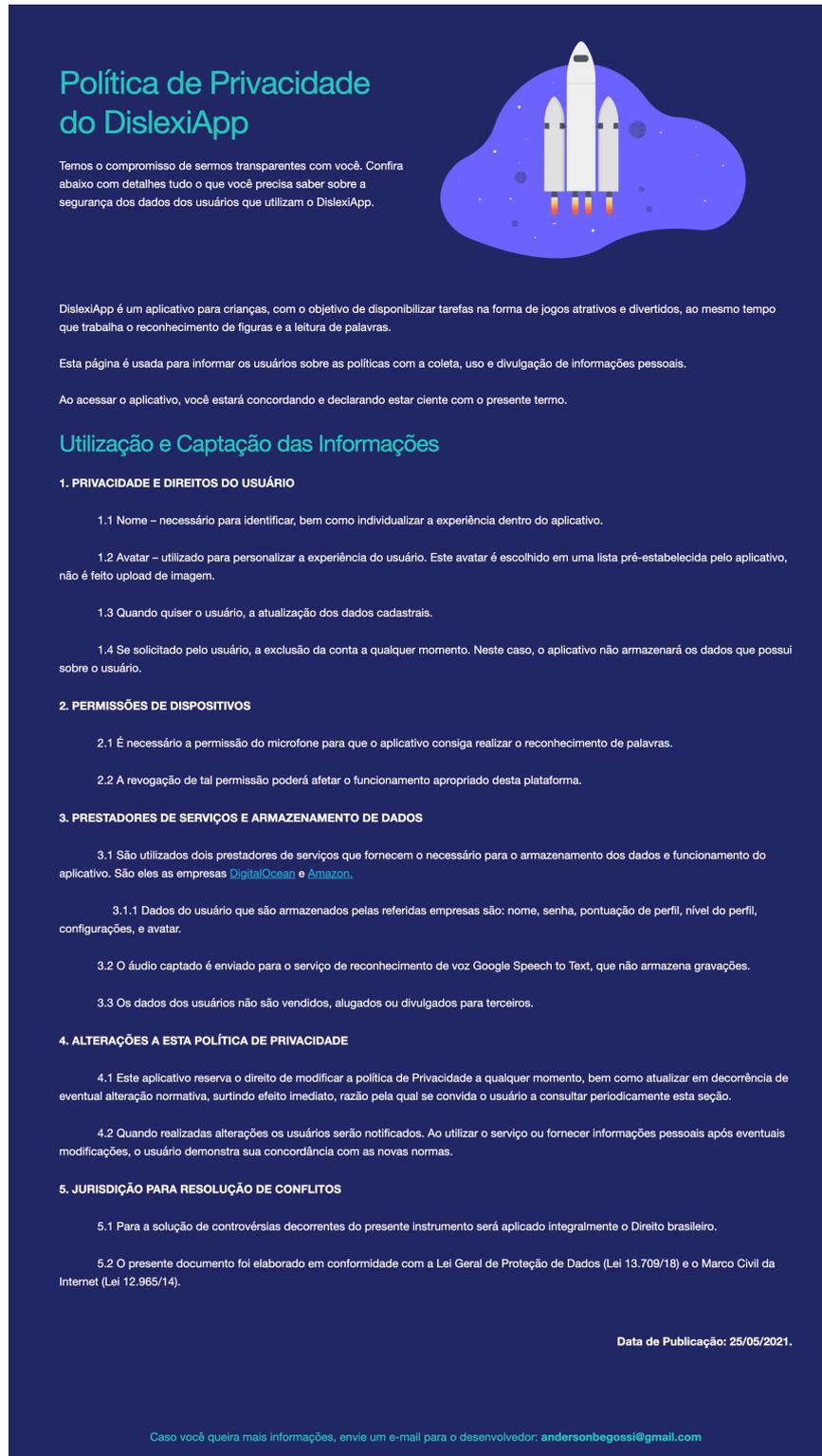
ZAMMETTI, F. **Practical Flutter**: Improve your mobile development with google's latest open-source sdk. California: Springer, 2019.

ZIEGLER, J. C. *et al.* Developmental dyslexia in different languages: Language-specific or universal? **Journal of experimental child psychology**, Elsevier, v. 86, n. 3, p. 169–193, 2003.

ZISIMOPOULOS, D. A.; GALANAKI, E. P. Academic intrinsic motivation and perceived academic competence in greek elementary students with and without learning disabilities. **Learning Disabilities Research & Practice**, Wiley Online Library, v. 24, n. 1, p. 33–43, 2009.

ANEXO A – POLÍTICA DE PRIVACIDADE

Figura 59 – Captura de tela da política de privacidade



Fonte: O Autor (2021).

ANEXO B – FORMULÁRIO DE AVALIAÇÃO

Figura 60 – Captura de tela do formulário de avaliação do aplicativo

Qual a idade da criança que utilizou o aplicativo? *

Menos de 5 anos

5

6

7

8

9

10

11

12

13

Mais de 13 anos

Qual foi o nível de interesse da criança em usar o aplicativo? *

1 2 3 4 5

Muito baixo Muito alto

Qual foi o nível de auxílio que a criança necessitou para usar o aplicativo? *

1 2 3 4 5

Muito baixo Muito alto

Dê uma nota geral para a experiência da criança com o uso do aplicativo. *

1 2 3 4 5

Muito ruim Muito boa

Como você (responsável) avalia o aplicativo? *

1 2 3 4 5

Muito ruim Muito bom

O aplicativo apresentou algum problema? Se sim, qual? *

Sua resposta _____

Descreva como foi a experiência da criança em usar o aplicativo. *

Sua resposta _____

Caso desejar, deixe algum comentário ou sugestão.

Sua resposta _____

Fonte: O Autor (2021).