



Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal Catarinense
Campus Videira

BRENDA ALVES MOURA

**USO DE DETECÇÃO DE OBJETOS EM IMAGENS PARA O
ENSINO DE VOCABULÁRIO ESTRANGEIRO**

Videira
2022

BRENDA ALVES MOURA

**USO DE DETECÇÃO DE OBJETOS EM IMAGENS PARA O
ENSINO DE VOCABULÁRIO ESTRANGEIRO**

Este Trabalho de Curso foi julgado adequado para a obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo curso de graduação em Ciência da Computação do Instituto Federal Catarinense – *Campus* Videira.

Videira (SC), 11 de Fevereiro de 2022

Manassés Ribeiro, D.r
Instituto Federal Catarinense – *Campus* Videira

BANCA EXAMINADORA

Kennedy Ferreira Araújo, M.e
Instituto Federal Catarinense – *Campus* Videira

Mauricio Natanael Ferreira, Esp.
Instituto Federal Catarinense – *Campus* Videira

Dedico este trabalho a todos que se interessam pelo aprendizado de idiomas, principalmente o russo. Também dedico àqueles que encontram dificuldades de aprendizado e precisam de ferramentas que auxiliem nessa jornada.

AGRADECIMENTOS

Agradeço aos meus pais que me deram apoio na estadia tão longe de casa. Agradeço aos meus amigos pelo apoio emocional e agradeço aos professores que, ao longo do curso, auxiliaram da forma como puderam na obtenção do conhecimento.

RESUMO

Com o decorrer dos anos, os dispositivos móveis se tornaram parte do cotidiano da maioria das pessoas. Devido à pandemia, responsável pelo distanciamento da população dos ambientes acadêmicos, as pessoas passaram a olhar os dispositivos móveis não só como uma ferramenta de distração mas também como uma ferramenta que possibilita o aprendizado, reforçando estratégias de aprendizado via dispositivos móveis, conceito este conhecido como *m-learning*. Entretanto, para um bom aprendizado por meio do *m-learning*, é necessário que os aplicativos ofereçam interatividade para o usuário, pois de outro modo, há perda de interesse pelo conteúdo a ser aprendido. Para contornar o problema da interatividade em aplicativos de aprendizado, este trabalho propõe um modelo baseado em tecnologia em nuvem para o reconhecimento de objetos em imagens capturadas pelo próprio usuário, que em seguida será identificada no idioma estrangeiro. A principal hipótese de trabalho é que um aplicativo interativo pode ser concebido a partir de ferramentas de reconhecimento de objetos em nuvem, e que este aplicativo pode ser útil no aprendizado de idiomas. Os passos metodológicos para o desenvolvimento do trabalho passam pela captura da imagem pelo dispositivo mobile, pré-processamentos iniciais, reconhecimento do objeto na imagem, identificação do objeto no idioma desejado e resposta ao usuário. A tradução para o idioma estrangeiro é realizada por meio da *API Google Translate*, sendo o desenvolvimento do aplicativo feito para a plataforma Android, devido a ser a mais utilizada. Experimentos foram conduzidos utilizando as tecnologias em nuvem do Google, conhecida como *Google Cloud Vision*, e da Amazon, conhecida como *Amazon Rekognition*. O objetivo dos experimentos consistiu em entender melhor as tecnologias estudadas e tentar identificar qual delas, eventualmente, poderia melhor se adequar aos objetivos deste trabalho. Como resultado, utilizou-se o *Google Vision* para o desenvolvimento do aplicativo interativo para o ensino-aprendizado de vocabulário estrangeiro utilizando detecção de objetos em imagens em um estudo de caso no idioma russo. Este aplicativo foi nomeado de Bazhenov.

Palavras-chave: Reconhecimento de objetos em imagens. Processamento de imagens na nuvem. Google Cloud Vision, Amazon Rekognition.

ABSTRACT

Over the years, mobile devices have become part of everyday life for most people. Due to the pandemic, responsible for the distancing of the population from academic environments, people started to look at mobile devices not only as a distraction tool but also as a tool that enables learning, reinforcing learning strategies via mobile devices, a concept known as m-learning. However, for good learning through m-learning, it is necessary that the applications offer interactivity to the user, otherwise, there is a loss of interest in the content to be learned. To overcome the problem of interactivity in learning applications, this work proposes a model based on cloud technology for the recognition of objects in images captured by the user, which will then be identified in a foreign language. The main working hypothesis is that an interactive application can be designed from cloud object recognition tools, and that this application can be useful in language learning. The methodological steps for the development of the work go through the capture of the image by the mobile device, initial pre-processing, recognition of the object in the image, identification of the object in the desired language and response to the user. The translation into a foreign language is performed through the API Google Translate, and the development of the application is made for the Android platform, due to being the most used. Experiments were conducted using cloud technologies from Google, known as Google Cloud Vision, and Amazon, known as Amazon Rekognition. The objective of the experiments was to better understand the technologies studied and try to identify which one, eventually, could best suit the objectives of this work. As a result, Google Vision was used to develop an interactive application for teaching and learning foreign vocabulary using object detection in images in a case study in Russian. This app was named Bazhenov.

Keywords: Object recognition in images. Cloud image processing. Google Cloud Vision, Amazon Rekognition.

LISTA DE ILUSTRAÇÕES

Figura 1 – Modelo de um neurônio artificial proposto por Rosenblatt	21
Figura 2 – Modelo de rede neural profunda	23
Figura 3 – Arquitetura <i>Google Vision</i>	25
Figura 4 – Arquitetura de reconhecimento de imagens da Amazon Rekognition .	27
Figura 5 – Fragmento das classes associadas ao cadastro do usuário	38
Figura 6 – Fragmento das classes associadas ao login	39
Figura 7 – Fragmento das classes associadas ao usuário	41
Figura 8 – Imagem enviada para a análise das <i>API'S</i>	48
Figura 9 – Imagem enviada para a análise de ambas as <i>API's</i>	49
Figura 10 – Comparação do percentual total de acertos e erros referente a cada <i>API</i>	53
Figura 11 – Da esquerda para direita: tela inicial, tela de cadastro e tela de login .	56
Figura 12 – Da esquerda para direita: tela principal, tela com resultados de foto da galeria e tela com resultados de imagem tirada com celular	57
Figura 13 – Da esquerda para direita: tela de visualizar perfil, tela para trocar foto e tela com menu lateral aberto	58
Figura 14 – Da esquerda para direita: tela de atualização do <i>email</i> , tela de atuali- zação da senha e tela para deletar o perfil	60
Figura 15 – Resultado do Teste de Usabilidade	61
Figura 16 – Formato DatePicker	70

LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
ML	Machine Learning
AM	Aprendizado de Máquina
UML	Unified Modeling Language
SDK	Software Development Kit
API	Application Programming Interface
QR	Quick Response
AWS	Amazon Web Services
CLI	Command-line Interface
IDE	Interface Development Environment

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivo Geral	13
1.2	Objetivos Específicos	13
2	Fundamentação Teórica	15
2.1	<i>E-learning</i>	15
2.2	<i>M-learning</i>	16
2.3	Aplicativos interativos para aprendizado	18
2.4	Aprendizado de máquina	19
2.5	Redes neurais artificiais	21
2.5.1	Aprendizado profundo	22
2.6	Processamento de imagens na nuvem	24
2.6.1	<i>Google Cloud Vision</i>	25
2.6.2	<i>Amazon Rekognition</i>	26
2.7	Trabalhos Correlatos	29
3	METODOLOGIA	32
3.1	Análise das API's de processamento de imagens na nuvem	32
3.1.1	Protocolo para avaliação das API's	32
3.1.2	Coleta do conjunto de dados	33
3.1.3	Metodologia para o envio das imagens pelo serviço Amazon <i>Re-</i> <i>kognition</i>	33
3.1.4	Metodologia para o envio das imagens pelo serviço Google <i>Vision</i>	35
3.1.5	Avaliação de dados	36
3.2	Metodologia do estudo de caso	37
3.2.1	Requisitos	37

3.2.2	Diagrama de Classes	38
3.2.3	Metodologia referente ao Teste de Usabilidade	42
3.2.4	Metodologia referente à Validação	43
4	EXPERIMENTOS COMPUTACIONAIS	45
4.1	Experimento #1: Avaliação de serviços de processamento de imagens na nuvem	45
4.1.1	Avaliação do modelo	46
4.1.2	Conclusão do experimento	51
4.2	Experimento #2: Estudo de caso	53
4.2.1	Aspectos de implementação na plataforma Android	54
4.2.2	Fluxo das interfaces e escolhas de design	54
4.2.3	Teste de Usabilidade	60
5	CONCLUSÃO E TRABALHOS FUTUROS	62
	REFERÊNCIAS	64
APÊNDICE A	Detalhes de implementação do aplicativo	68
A.1	Técnicas e ferramentas utilizadas	68
A.2	Detalhamento da implementação das Classes	69
A.2.1	Detalhamento das validações	73

1 INTRODUÇÃO

O ensino à distância universal é uma das inovações que a pandemia do coronavírus trouxe às nossas vidas. Escolas e universidades foram forçadas a transferir urgentemente seus programas para o formato *online*, trazendo uma nova visão das pessoas para o aprendizado utilizando celulares e outros dispositivos eletrônicos (PASINI; CARVALHO; ALMEIDA, 2020). A aprendizagem virtual pode ser chamada de *disruptiva e mudança de paradigma*, especialmente quando seu foco muda para a aprendizagem fora da escola tradicional, ou para superar as deficiências percebidas nos currículos existentes e nos métodos de avaliação de sua eficácia. Se antes muitos associavam estes aparelhos à distração, o ensino *online* devido à pandemia trouxe um novo olhar sobre este método de ensino-aprendizado conhecido como *e-learning*, que pode ser definido como a oferta de educação e treinamento usando recursos digitais (SAFANELLI; CATAPAN; ROCHA, 2012), e dentro do *e-learning* surge a categoria *m-learning* (TRAXLER, 2005), relacionada apenas ao ensino-aprendizado via dispositivos móveis. Embora o método não seja novo, sua inclusão em vários setores foi acelerada pela pandemia do novo coronavírus.

A educação móvel, termo cunhado no trabalho de Fluminhan et al. (2020), implica no surgimento de uma ampla gama de novos métodos de ensino-aprendizado baseados na crença de que a interação em uma sala de aula tradicional muitas vezes não é tão eficaz quanto se gostaria (ARKORFUL, 2014). Para compensar a diferença, diversas abordagens de ensino são utilizadas para conquistar a atenção do usuário, como por exemplo, a implementação de elementos interativos como sons, componentes com movimento, realidade virtual, realidade aumentada, e outros elementos que possam captar mais eficientemente a atenção do usuário (CRIOLLO-C et al., 2021). Caso o objetivo de prender a atenção do usuário não seja cumprido, a consequência resultante é um menor aprendizado devido à perda de atenção, podendo levar o usuário a sentir-se desmotivado e abandonar o aprendizado (HARTNETT, 2016).

Os aplicativos móveis são a forma mais comum de expandir o vocabulário (PEGRUM, 2014). Atualmente, muitos aplicativos oferecem a capacidade de expandir o vocabulário durante um jogo. Por exemplo, Quizlet ¹, criado como um jogo de teste, é usado com *flashcards* para ajudar o usuário a testar o vocabulário aprendido. Também está disponível a transcrição de áudio para trabalhar na pronúncia (IFTAKHAR, 2016). Outro exemplo é o Bookwidgets ², o aplicativo inclui modelos para perguntas e jogos (*flashcards*, palavras cruzadas, etc.), que são classificados automaticamente e visualmente mais atraentes do que os materiais de estudo padrão em papel (MENON, 2019).

A abordagem para aprender a gramática é, em muitos aspectos, semelhante à abordagem para consolidar o material de vocabulário em exercícios MALL (do inglês *Mobile Assisted Language Learning*) (ALDA LUCIA E LEFFA, 2014). É o que demonstra o projeto MASELTOV³, que coloca a gramática, o vocabulário e outros aspectos da língua (MASELTOV, 2016).

Atualmente, a literatura contém diversas opções para trabalhar com a interatividade de aplicativos, em particular para reconhecimento de padrões, sendo a interatividade por reconhecimento de padrões sendo aplicada a diversos aplicativos. As tecnologias de informação modernas fornecem uma ampla gama de ferramentas para a implementação de tais algoritmos de reconhecimento de padrões, sendo as mais modernas relacionadas ao processamento de imagens na nuvem. Algumas das ferramentas mais conhecidas neste ramo são o Google Cloud Vision API, IBM Watson Visual Recognition e Amazon Rekognition.

Diante dos expostos, o que se pretende abordar neste trabalho é o desen-

¹ <https://quizlet.com/pt-br>

² <https://www.bookwidgets.com/>

³ Aplicação Móvel para a Integração Social e Empoderamento de Imigrantes por meio de Tecnologias de Aprendizagem de Incentivos e Redes Sociais, ou em inglês *Mobile Application for Social Integration and Immigrant Empowerment through Incentive Learning Technologies and Social Networks*

volvimento de um aplicativo mobile interativo destinado ao aprendizado de idiomas. A ideia é que a interatividade aconteça por intermédio da câmera do dispositivo do usuário, responsável por capturar uma imagem. A hipótese é que a imagem pode ser processada utilizando tecnologias de reconhecimento de objetos em nuvem, que pode ser útil para o desenvolvimento de plataformas interativas de aprendizado de idiomas.

1.1 OBJETIVO GERAL

Propor um aplicativo para a detecção de objetos em imagens utilizando processamento em nuvem para o aprendizado de idiomas.

1.2 OBJETIVOS ESPECÍFICOS

Para melhor entendimento, o objetivo geral é sub-dividido em objetivos específicos, que são:

- Investigar as tecnologias atuais para a detecção de objetos em imagens que utilizem processamento em nuvem;
- Comparar o desempenho das principais tecnologias investigadas no objetivo anterior;
- Sugerir uma metodologia para o desenvolvimento de aplicativos de detecção de objetos em imagens com processamento na nuvem;
- Desenvolver um aplicativo utilizando a metodologia proposta para o aprendizado de idiomas⁴;
- Conduzir um caso de uso para validação do aplicativo no idioma russo.

O restante deste trabalho é composto por três capítulos. O Capítulo 2 apresenta a Fundamentação Teórica utilizada sobre os temas abordados no trabalho como

⁴ Como forma de delimitar o escopo o aplicativo será desenvolvido apenas para o sistema operacional Android.

os conceitos de *m-learning*, redes neurais convolucionais, reconhecimento de imagens, processamento de imagens na nuvem, *API Cloud Vision* e *Amazon Rekognition* e os trabalhos correlatos. No Capítulo 3 são descritos como ocorrerão os passos metodológicos que se pretende seguir neste trabalho, bem como experimentos necessários e relacionados com o processamento de imagens e reconhecimento de padrões na nuvem. No Capítulo 4 constam as informações referentes aos experimentos e resultados. Por fim, no Capítulo 5 são apresentadas as principais conclusões preliminares.

2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo é composto por sete seções e três subseções. Na Seção 2.1 são expostas as concepções sobre *e-learning*. A Seção 2.2 são explicados o conceito de *m-learning*, sua relevância e as diferenças e semelhanças com o modelo *e-learning*. Na Seção 2.4, explana-se o conceito de aprendizado de máquina, utilizado pelas tecnologias de reconhecimento de objetos. Na Seção 2.5, é abordado o assunto de redes neurais artificiais, sendo utilizada também para aprender sobre diversos assuntos, inclusive no reconhecimento e classificação de imagens, tendo por fim a Subseção 2.5.1 tratando do aprendizado profundo. Na Seção 2.3 aborda-se algumas características que dispõem os aplicativos interativos. Na Seção 2.6 é apresentado o conceito de processamento de imagens na nuvem e suas vantagens no cenário atual, após isso, aborda-se duas tecnologias de reconhecimento de imagens em nuvem, sendo a Seção 2.6.1 sobre o serviço *Google Cloud Vision* e a Seção 2.6.2 sobre o serviço *Amazon Rekognition*, tendo por fim a Seção 2.7 que apresenta quatro trabalhos correlatos que contém as características de aprendizado de idiomas ou reconhecimento de imagens na nuvem.

2.1 E-LEARNING

Nos últimos anos, o termo *e-learning* (abreviação de *electronic learning* em inglês), se popularizou. O termo *e-learning* significa não apenas ensino a distância, mas todos os tipos de educação que utilizam tecnologias de informação e comunicação. (BERMAN, 2006).

O *e-learning* vem em muitas formas. Trata-se de uma variedade de plataformas abertas onde qualquer pessoa pode desfrutar de uma série de cursos gratuitos e mais avançados. Exemplos estrangeiros incluem: *Khan Academy*¹ e *Code Academy*²,

¹ Pode ser acessado em www.khanacademy.org

² Pode ser acessado em www.codecademy.com

sendo estes exemplos centros de educação a distância fundados por universidades. Por fim, há o treinamento corporativo utilizando tecnologias de *e-learning*.

Ao mesmo tempo, existem muitas formas de treinamento de *e-learning*, por exemplo (BERMAN, 2006):

- Série de tutoriais em vídeo (por exemplo: ted.com);
- Série de artigos de texto, divididos por etapas / lições, que podem ser complementados com vídeo;
- Discussão especialmente organizada e moderada em fóruns;
- Teste *online* e outras tarefas;
- *Webinars* (do inglês: web + seminário) - transmissão da fala do professor por meio de uma interface especial para os ouvintes com possibilidade de *feedback*, captura de tela, etc;
- Jogos educativos interativos;
- Cursos *online*;

2.2 M-LEARNING

Dentro do campo *e-learning*, encontra-se o conceito de *m-learning* (do inglês *mobile learning*), que significa aprender com a ajuda de aplicativos móveis, está gradualmente entrando na vida de todos (TRAXLER, 2005).

O termo “aprendizagem móvel” tem muitas definições na literatura nacional e estrangeira. Assim, Traxler (2005) caracteriza a aprendizagem móvel como qualquer oferta de educação, onde as únicas tecnologias ou tecnologias dominantes são os dispositivos portáteis, e faz uma reserva de que a aprendizagem móvel pode incluir o uso de telefones, *smartphones*, *tablets* e *laptops*, consoles de jogos, *iPods*. De acordo com Georgiev, Georgieva e Trajkovski (2006), o aprendizado móvel é “um novo estágio no desenvolvimento do *e-learning*”.

A necessidade de estar diretamente na frente do computador e de estar conectado à rede por meio de cabos limitava significativamente a mobilidade dos alunos.

Com o aprendizado móvel, os alunos não estão mais limitados a um computador, eles podem navegar pelos materiais ou se comunicar com um professor enquanto estão em um café ou enquanto esperam por um ônibus em uma parada. Essas situações demonstram o quanto a tecnologia móvel tornou o aprendizado mais flexível (HASHEMI et al., 2011).

Conforme definido por Kambourakis, Kontoni e Sapounas (2004), a aprendizagem móvel “é o ponto em que os dispositivos móveis e o *e-Learning* se cruzam para criar uma experiência de aprendizagem acessível, a qualquer hora e em qualquer lugar”. Tendo considerado as definições de *e-learning* e *mobile learning*, as seguintes conclusões podem ser inferidas sobre as semelhanças e diferenças entre esses dois conceitos:

- *Semelhanças*: a aprendizagem é um componente-chave de ambas as definições. O objetivo principal, independentemente das tecnologias utilizadas, é a assimilação do conhecimento pelo usuário;
- *Diferenças*: o aprendizado móvel é muito mais autônomo do que o *e-learning*. A aprendizagem móvel é quase sempre informal (PEDRO; BARBOSA; SANTOS, 2018). Além disso, o aprendizado móvel oferece aos participantes muito mais liberdade e uma grande variedade de opções de onde e quando aprenderão.

Tendo em vista este fator, os desenvolvedores de aplicativos para dispositivos móveis na área educacional devem levar em consideração os seguintes pontos (HASHEMI et al., 2011):

- Tais aplicativos devem ser capazes de suportar os padrões de comunicação móvel 3G ou 4G;
- Precisam ter interface e navegação amigáveis, pois trabalhar em uma tela pequena requer uma disposição gráfica diferente de símbolos e ícones;
- Minimização de material visual que se distraia do conteúdo educacional

real, o que significa a ausência de um grande número de imagens animadas, janelas *pop-up*, gráficos longos e símbolos multicoloridos;

- A capacidade de criar um aplicativo como um complemento ao livro didático em papel / web / eletrônico;
- Implementação do princípio de *edutainment* (educação + entretenimento, ou seja, aprendizagem através do entretenimento), que aumentará a audiência de interessados no conteúdo (CHING et al., 2018);
- Dependendo da plataforma na qual e para a qual o aplicativo está sendo criado, é necessário levar em consideração a probabilidade de comercialização / monetização do recurso (HSU; LOPEZ; RAJ.S, 2020).

2.3 APLICATIVOS INTERATIVOS PARA APRENDIZADO

Interatividade é um conceito que revela a natureza e o grau de interação entre objetos ou sujeitos. É utilizado nas seguintes áreas: teoria da informação, informática e programação, sistemas de telecomunicações, sociologia, *design*, em particular *design* de interação, entre outros (STROMER-GALLEY, 2004).

Atualmente, não existe uma definição específica estabelecida do significado do termo *interatividade* entre os especialistas nas áreas listadas. No entanto, no geral, a interatividade pode ser definida como o princípio da organização do sistema, em que o objetivo é alcançado pela troca de informações dos elementos desse sistema (YANG; SHEN, 2017).

Os aplicativos interativos móveis são uma variedade de programas desenvolvidos para dispositivos móveis, como *tablets* e *smartphones*, que permitem que seja executado de forma fácil e rápida várias funções no próprio aparelho, como navegação na internet ou interação com outros equipamentos. A principal diferença, em relação a outros aplicativos, é que o próprio usuário pode interagir com o programa e utilizá-lo para realizar determinadas ações e não apenas receber determinadas informações. Existem muitos programas interativos para realizar centenas de tarefas diferentes como treinamento, colaboração com a informação,

orientação, comunicação, criatividade, interação com a tecnologia, dentre outros.

Para Starovoytov (2018) um bom aplicativo interativo para dispositivos móveis se distingue por:

- *Interface amigável*: os princípios de controle e uso do programa devem ser intuitivos para o usuário, a navegação dentro do aplicativo deve ser conveniente;
- *Design de qualidade*: a aplicação deve ser de alta qualidade e moderna, sua aparência deve estar em harmonia com as funções desempenhadas;
- *Boa otimização*: o aplicativo deve iniciar rapidamente e funcionar sem problemas, mesmo nos dispositivos mais fracos;
- *Funcionalidade completa*: o aplicativo deve completar todas as tarefas atribuídas a ele por completo.

2.4 APRENDIZADO DE MÁQUINA

Em 1959, Arthur Samuel cunhou o termo aprendizado de máquina, que pode ser definido como um processo, em que uma máquina é capaz de mostrar um comportamento que não foi explicitamente colocado nela, isto é, que não foi programado diretamente. Samuel desenvolveu um programa que aprendeu a jogar *Damas* melhor do que ele. (BOWLING et al., 2006).

Tom Mitchell, outro conhecido pesquisador de aprendizagem de computação, sugeriu uma definição mais precisa em 1997: é dito que um programa de computador aprende com base na experiência E com respeito a uma certa classe de problemas T e uma medida de qualidade P , se a qualidade de resolução de problemas de T , medida com base em P , melhora com a aquisição de experiência E (MITCHELL, 1997).

As definições acima estabelecem uma meta clara para o aprendizado de máquina (AM), mas não como atingir esta meta. É preciso refinar a definição. Aprendizado de máquina é a prática de usar algoritmos para analisar dados, estudá-los

e, em seguida, definir ou prever algo no mundo.

A seguir são apresentados os principais algoritmos de AM (SATHYA; ABRAHAM, 2013):

- *Aprendizagem supervisionada*: o algoritmo recebe dados de aprendizagem que contêm a “resposta correta” para cada amostra do conjunto de dados. Por exemplo, um algoritmo de aprendizagem supervisionada para detecção de fraudes em cartão de crédito, teria um conjunto de transações anotadas como entrada. Para cada transação, os dados de treinamento conterão um rótulo indicando se é fraudulenta ou não. Assim, o sistema de AM é treinado com antecedência e com a ajuda do “professor”;
- *Aprendizagem não-supervisionada*: o algoritmo não procura por pares objeto-resposta, mas por estrutura (conexões) nos dados de aprendizagem, por exemplo, quais exemplos são semelhantes entre si, e os agrupa em agrupamentos. Nesse caso, o sistema de AM aprende espontaneamente, sem a intervenção de um professor;
- *Aprendizagem semi-supervisionada*: A aprendizagem por algoritmo combina os princípios da aprendizagem supervisionada e não-supervisionada. Por exemplo, ao preparar um algoritmo, apenas parte dos dados de treinamento é marcada e apenas algumas das regras são fornecidas (REDDY; PULABAIGARI; B, 2018).

2.5 REDES NEURAIIS ARTIFICIAIS

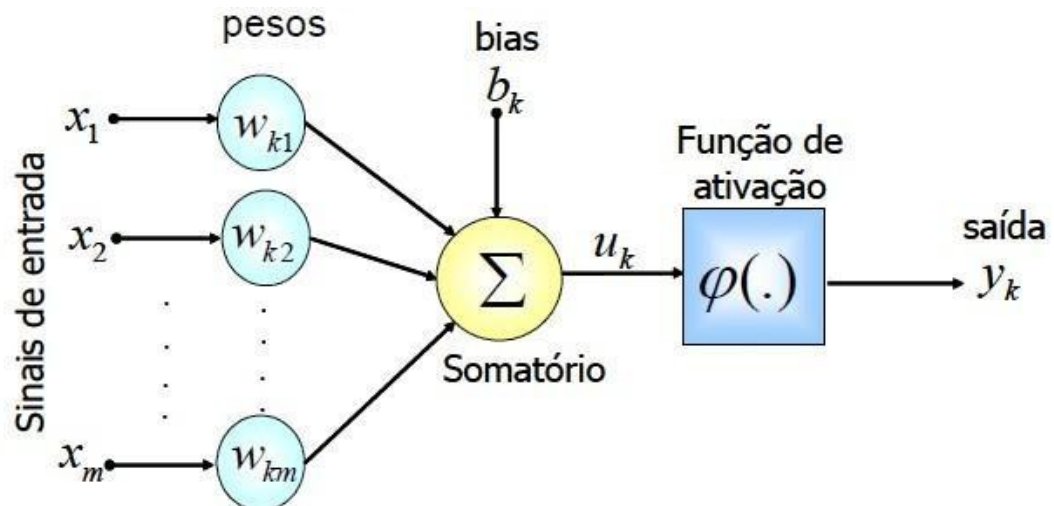
Uma rede neural artificial é um poderoso mecanismo de aprendizagem de máquina que basicamente imita a maneira como o cérebro humano aprende. O cérebro recebe impulsos do mundo externo, executa o processamento de entrada e, em seguida, gera sinais de saída. Usando uma rede neural artificial, estamos tentando mimetizar este comportamento (ZAYEGH; BASSAM, 2018).

Pela primeira vez, o conceito de uma rede neural artificial foi considerado

em 1943 pelos biofísicos W. Pitts e W. McCulloch no trabalho “A logical calculus of the ideas immanent in nervous activity” (MCCULLOCH; PITTS, 1943). Em particular, eles propuseram um modelo de neurônio artificial (SZU; ROGERS, 1992). Na década de 1960, F. Rosenblatt, usando o modelo obtido de McCulloch-Pitts, continuou esse desenvolvimento e, posteriormente, introduziu o *Perceptron* (DU; SWAMY, 2014).

O modelo do *Perceptron* proposto por Rosenblatt (ROSENBLATT, 1958) pode ser visto na Figura 1, onde X corresponde ao sinal de entrada, W é o peso do sinal de entrada, $F()$ a função de ativação e u_k o valor obtido pelo somatório dos sinais de entrada:

Figura 1 – Modelo de um neurônio artificial proposto por Rosenblatt



Fonte: Soares e Silva (2011)

Como pode ser observado na Figura 1, o neurônio recebe sinais por meio dos canais de entrada ($x_1 \dots x_m$) e cada um dos sinais tem um certo peso ($w_{k1} \dots w_{km}$). Cada uma das entradas é pesada e somada no nó. Este procedimento é formalizado pela Equação 2.1:

$$u_k = \sum_{i=1}^n (x_i \times w_{ki}), \quad (2.1)$$

onde x_i representa o sinal de entrada e w_{ki} seu respectivo peso.

O símbolo b é a polarização (*bias*), que é somada às entradas para formar as entradas líquidas. A polarização e os pesos são parâmetros ajustáveis do neurônio, que por sua vez são ajustados usando algumas regras de aprendizagem.

A saída de um neurônio pode variar de $-\infty$ a $+\infty$, pois o neurônio não conhece o limite. Portanto, é preciso fazer uso de um mecanismo de mapeamento entre a entrada e a saída do neurônio. Este mecanismo de mapeamento de entradas para saídas é conhecido como *função de ativação* (S.; ABRAHAM, 2019).

2.5.1 Aprendizado profundo

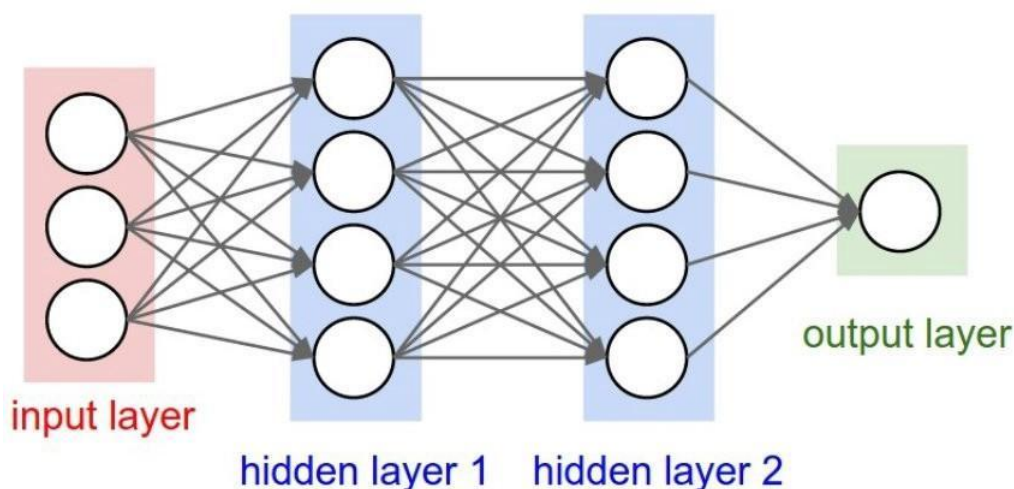
Os métodos tradicionais de AM têm sua capacidade de processar dados em sua forma original limitadas. Por décadas, construir um sistema de reconhecimento de padrão ou AM exigiu um *design* cuidadoso e experiência significativa no projeto de um mecanismo de extração de recursos que transforma dados brutos (como valores de pixel de uma imagem) em uma representação interna adequada, a partir da qual, o mecanismo de aprendizado poderia detectar ou classificar os padrões.

Técnicas de aprendizado profundo (do inglês *deep learning*) são métodos para treinar uma representação com vários níveis de abstração, obtidos pela composição de módulos simples e não lineares, cada um dos quais transforma a representação em um nível (começando pela entrada original) em uma representação em um nível superior, subindo a um nível mais abstrato. Isso é o que torna o aprendizado profundo complexo. Cada nível descreve algum tipo de informação, organiza e transfere para o próximo nível. Usando uma imagem como exemplo, que inicialmente está na forma de uma matriz de valores de pixel, os objetos detectados na primeira camada de representação geralmente referem-se a presença (ou ausência) de bordas em certas orientações e localizações da imagem. A segunda camada geralmente define modelos destacando posições de borda específicas, independentemente de pequenas alterações nas posições de borda. A terceira camada pode coletar modelos em combinações maiores que correspondem a partes de objetos familiares, e as camadas subsequentes detectarão objetos como combinações destas partes (MATHEW; ARUL;

SIVAKUMARI, 2021).

Pode-se considerar o modelo de rede profunda usando o exemplo da Figura 2, onde entre as camadas de entrada (*input layer*) e saída (*output layer*), existem muitas camadas ocultas (*hidden layer* [1..]). Quanto mais houver, mais profunda a arquitetura da rede será. Importante mencionar que a entrada refere-se aos dados a serem processados, e a saída trata-se do resultado da predição da rede.

Figura 2 – Modelo de rede neural profunda



Fonte: Abdelfattah (2017)

O aprendizado profundo está realizando grandes avanços na solução de problemas aos quais há muitos anos as melhores tentativas de Inteligência Artificial (IA) enfrentam. Esta técnica provou ser muito boa na detecção de estruturas complexas em dados multidimensionais e, portanto, é aplicável a muitas áreas da ciência. As técnicas de aprendizado profundo influenciaram muito o estado da arte em reconhecimento de fala e visual (FOUKI; AKNIN; KADIRI, 2019).

2.6 PROCESSAMENTO DE IMAGENS NA NUVEM

O reconhecimento de imagem permite que os computadores reconheçam imagens como humanos. Tradicionalmente, os algoritmos de ML (Machine Learning), em especial as redes neurais convolucionais, têm sido amplamente utilizados para o reconhecimento de imagem e vídeo, inclusive para serviços da web em nuvem. Ao

longo dos anos as técnicas foram aprimoradas melhorando tanto em arquitetura quanto em desempenho de classificação (CUMMAUDO et al., 2020). Atualmente, as *API's* (do inglês *Application Programming Interface*) *Google Cloud Vision* e *Amazon Rekognition* implementam sofisticados algoritmos de reconhecimento de padrões. Com isto, as aplicações podem ter acesso de maneira ágil e fácil a um conjunto de ferramentas e recursos para reconhecimento de padrões em imagens.

A arquitetura em nuvem oferece várias vantagens em relação ao processamento de imagens tradicionais. Alguns desses benefícios incluem (ALTARAWNEH; AL-QAISI, 2019):

- Fácil integração;
- Processamento de imagem com resultados estáveis;
- O fato das ferramentas de processamento na nuvem possuírem modelos pré-treinados para a detecção de diversos tipos de objetos em imagens, como carros, imóveis, animais, etc.

2.6.1 *Google Cloud Vision*

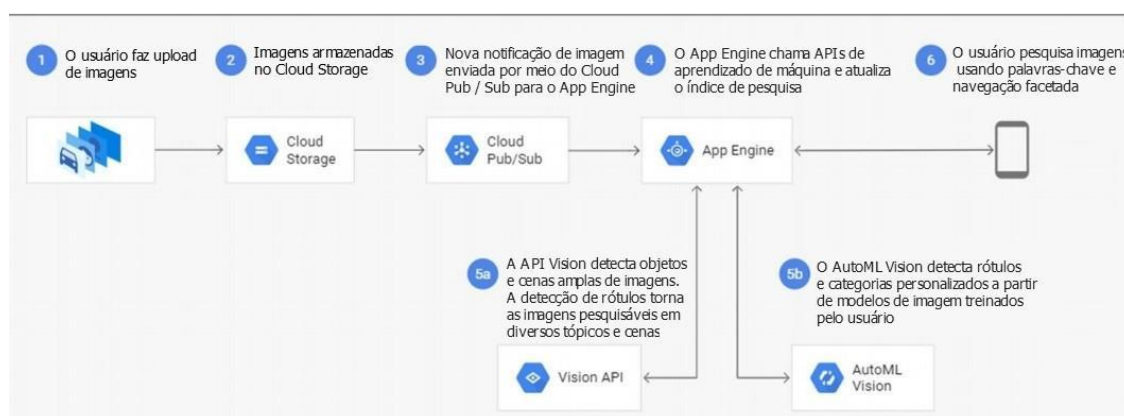
O *Google Cloud Vision* (também conhecido por *Google AutoML* ou *Google Vision Vision AI*)³ oferece modelos de redes neurais profundas pré-treinadas, mas personalizáveis, para tradução de idiomas, classificação de texto, detecção de objetos, classificação de imagens e classificação e rastreamento de objetos de vídeo. Os modelos exigem dados de aprendizagem previamente rotulados, mas não exigem aprendizagem avançada nos conceitos de AM, transferência de conhecimento (*transfer learning*) ou conhecimento crítico de programação.

O *Google Cloud Vision* pode configurar uma rede neural profunda de alta precisão testada pela Google para dados rotulados. O *Vision* treina modelos a partir de dados pré-existentes em vez de começar do zero e implementa aprendizado profundo automático com transferência de conhecimento (ou seja, começando a partir de redes neurais profundas pré-treinadas com base de dados de outros problemas).

Conforme apontado por Franco (2018), relacionado ao fluxo de trabalho

da arquitetura *Google Vision*, o portal *Cloud Vision*, apresenta como exemplo o fluxo de uma aplicação que realiza a pesquisa¹ de imagens com base nos tópicos, cenas e categorias identificadas nas imagens analisadas. Este fluxo é demonstrado na Figura 3, onde o usuário efetua o *upload* das imagens a serem analisadas para o serviço *Cloud Storage*, que aciona as *API's* de AM que realizam a detecção dos objetos e categorias da imagem.

Figura 3 – Arquitetura *Google Vision*



Fonte: Franco (2018)

Em comparação com o treinamento de uma rede neural do zero, a aprendizagem por transferência de conhecimento tem duas vantagens principais: primeiro, requer muito menos dados de treinamento, pois a maioria das camadas da rede são bem treinadas. Em segundo lugar, ele aprende mais rápido porque otimiza apenas a última camada.

2.6.2 Amazon Rekognition

Amazon Recognition (AMAZON, 2021) é uma ferramenta de análise que avalia arquivos de imagem e vídeo e pode reconhecer objetos, rostos e cenas, sendo que avalia e interpreta os resultados com base em determinados critérios. A IA da Amazon pode ser utilizada com o uso de dados antecipadamente fornecidos via

¹³ Link para o site oficial: <https://cloud.google.com/vision>

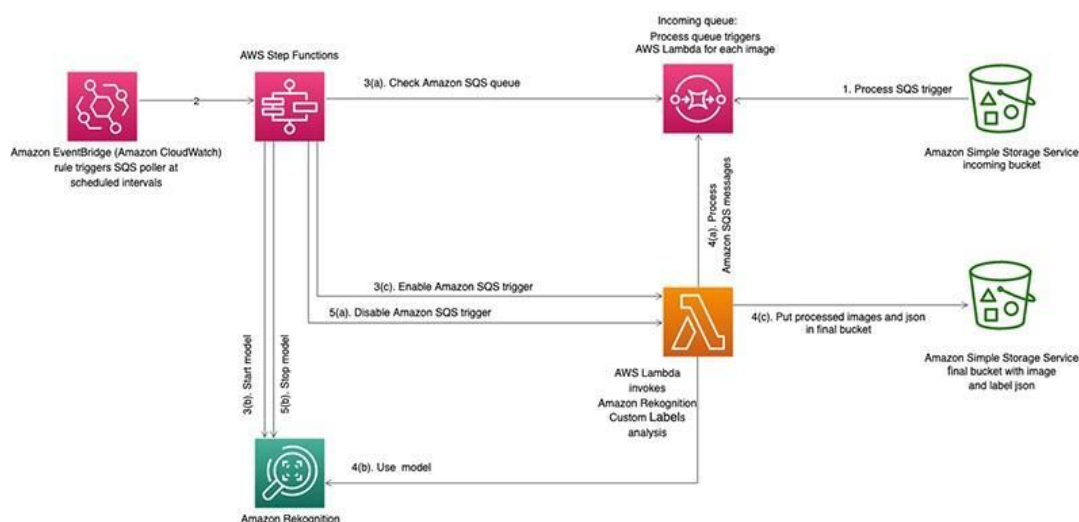
transferência de conhecimento, ou desenvolver o aprendizado de forma independente a partir do zero. O serviço utiliza a tecnologia de aprendizado profundo, altamente escalonável e pensada para ser utilizada sem a necessidade de experiência em AM.

Para reconhecer objetos, o sistema analisa imagens ou vídeos e reconhece o cenário com base em determinados recursos e pode interpretar a situação. Os objetos individuais são vistos e analisados separadamente um do outro. A IA avalia os resultados e os classifica como uma porcentagem com base em um esquema fixo. Desta forma, o sistema decide independentemente se o reconhecimento de um objeto é claro o suficiente para ser capaz de nomeá-lo. Esta função pode ser particularmente interessante para pesquisas de imagens. Os bancos de dados de imagens podem ser filtrados para um termo de pesquisa específico e apenas imagens adequadas são geradas.

Os dados de imagem e vídeo capturados podem ser integrados de forma conveniente em aplicativos e serviços por meio de *APIs*, que pode ser facilmente integrada a sites, aplicativos ou programas. Para permitir que os aplicativos detectem objetos ou cenas em uma imagem, o *Amazon Rekognition* usa rótulos. De acordo com a Amazon, o serviço *Rekognition* suporta milhares de rótulos diferentes para poder reconhecer uma imagem, que são aplicados a objetos, cenas ou conceitos específicos encontrados em uma imagem.

O diagrama de arquitetura na Figura 4 mostra como criar um fluxo de trabalho econômico e altamente escalonável para imagens em lote usando os rótulos do *Amazon Rekognition*. Ele usa serviços da AWS, como *Amazon Simple Queue Service* (Amazon SQS), *Amazon EventBridge*, *AWS Stepping Functions*, *AWS Lambda* e *Amazon Simple Storage Service* (Amazon S3) que são as arquiteturas necessárias para o funcionamento do reconhecimento de imagem e que serão explicados a seguir.

Figura 4 – Arquitetura de reconhecimento de imagens da Amazon *Rekognition*



Fonte: Srivastava (2021)

Esta solução usa uma arquitetura sem servidor e serviços gerenciados, para que possa escalar sob demanda, sem a necessidade de provisionar ou gerenciar servidores. O Amazon SQS melhora a resiliência geral da solução separando a aquisição de imagens do processamento de imagens e garantindo a entrega confiável de mensagens para cada imagem carregada. O *Step Functions* facilita a criação de fluxos de trabalho visuais para organizar uma série de tarefas discretas, como verificar se uma imagem está disponível para processamento e gerenciamento do ciclo de vida do estado do projeto *Amazon Rekognition Custom Labels*. Embora a arquitetura a seguir mostre como criar uma solução de processamento em lote para *Amazon Rekognition Custom Labels* usando *AWS Lambda*, é possível criar uma arquitetura semelhante usando serviços como *AWS Fargate* (AMAZON, 2021).

As etapas a seguir descrevem o fluxo de trabalho geral para uso do *Amazon Rekognition*:

1. Como a imagem é armazenada no *bucket* do Amazon S3, ela dispara uma mensagem que é armazenada na fila do Amazon SQS;
2. O Amazon *EventBridge* é configurado para executar o fluxo de trabalho *AWS Step Functions* em uma frequência específica (1 hora por padrão);

3. Durante a execução do fluxo de trabalho, os seguintes passos são realizados:

- Verifica-se o número de itens na fila do Amazon SQS. Se não houver itens na fila para processar, o fluxo de trabalho termina;
- Se houver itens na fila para processar, o fluxo de trabalho inicia o modelo de tags personalizadas do *Amazon Rekognition*;
- O fluxo de trabalho integra Amazon SQS com uma função AWS *Lambda* para processar essas imagens.

4. Quando a integração é ativada, entre uma fila Amazon SQS e AWS *Lambda* ocorrem os seguintes eventos:

- Realiza a interrupção do modelo de *tags* personalizadas do Amazon *Rekognition*;
- Desativa a integração entre a fila Amazon SQS e a função AWS *Lambda*, desativando o gatilho.
- O AWS *Lambda* começa a processar mensagens de detalhes de imagem do Amazon SQS;
- A função AWS *Lambda* usa o projeto Amazon *Rekognition Custom Labels* para processamento de imagem;
- A função AWS *Lambda* então coloca o arquivo *JSON* contendo os rótulos renderizados no último intervalo. A imagem também se move do segmento original para o último segmento.

5. Depois que todas as imagens foram processadas, o fluxo de trabalho do AWS *Step Functions* faz o seguinte procedimento:

- Realiza a interrupção do modelo de *tags* personalizadas do serviço Amazon *Rekognition*;

- Desativa a integração entre a fila Amazon *SQS* e a função AWS *Lambda*, desativando o gatilho.

2.7 TRABALHOS CORRELATOS

Nesta seção são apresentados os principais trabalhos correlatos à proposta deste trabalho. O foco principal deste levantamento está nos aplicativos mais populares que envolvam tanto o reconhecimento de imagens com processamento em nuvem, quanto aplicativos interativos para o aprendizado de idiomas. Além disso, foram considerados aplicativos para a plataforma *Android*.

O aplicativo *TapTapSee* foi desenvolvido pela *CloudSight Inc* (TAPTAP-SEE, 2021) para ajudar pessoas com deficiência visual a detectar os objetos que encontram em seu cotidiano. O aplicativo utiliza a *API* de reconhecimento de imagem *CloudSight*, com as funções de câmera para fotografar os objetos e *Talkback* (leitor de tela do Google pré-instalado em dispositivos *Android* (GOOGLE, 2021)) para verbalizá-los para o usuário. A ideia do aplicativo é que o usuário possa ativá-lo apenas tocando duas vezes na tela do *smartphone*, que então captura a imagem e a envia para o servidor. Após receber a resposta, o dispositivo *Talkback* relata os resultados ao usuário (TAPTAPSEE, 2021). Embora esse aplicativo reconheça imagens por intermédio do uso de processamento de imagens na nuvem, este aplicativo não tem como objetivo o aprendizado de idiomas, apenas o auxílio de pessoas portadoras de deficiência visual.

Outro aplicativo avaliado é o *Google Lens*, que é um aplicativo de reconhecimento de imagem desenvolvido para obter informações sobre objetos. Funciona com base na análise visual, que é realizada por uma rede neural profunda. Por meio do aprendizado profundo, é obtido melhor desempenho na detecção dos objetos, expandindo assim os recursos do aplicativo (MUREs, AN; OLTEAN, 2018). De modo geral, quando a câmera do *smartphone* é apontada para um objeto, o Google

Lens tentará identificá-lo e, em seguida, exibirá os resultados da pesquisa, páginas da web (caso contenha o objeto em alguma página na internet, o Google *Lens* detecta o site e disponibiliza o link), e informações adicionais. Em sua versão atual, o aplicativo consegue traduzir texto de uma foto, procurar objetos em lojas *online*, reconhecer cardápios e recomendar pratos a partir dele (GOOGLE, 2021). Embora seja um aplicativo com diversas funcionalidades, incluindo a tradução de idiomas, os recursos são apresentados em categorias separadas. O texto é traduzido somente ao apontar a câmera para um texto, sendo a tradução feita em tempo real. Enquanto o reconhecimento de objetos ocorre numa outra categoria, impossibilitando que seja feita a tradução do resultado retornado pelo reconhecimento de objetos.

Um outro trabalho correlato a ser pontuado refere-se à monografia produzida por Franco (2018), onde foi proposto o desenvolvimento de um aplicativo utilizando tecnologia em nuvem para auxiliar pessoas com deficiência visual. Embora este trabalho tenha foco em pessoas com deficiência, as tecnologias utilizadas assemelham-se à proposta deste trabalho, com o diferencial sendo a função do aplicativo. Enquanto o trabalho da monografia citada tem função de auxiliar deficientes, o aplicativo proposto neste trabalho tem como função auxiliar no aprendizado.

Por fim, foi avaliado o aplicativo *EngCards*, que contém cerca de 3.500 cartas na forma de palavras e imagens no idioma inglês com pronúncia profissional de um falante nativo. As palavras são selecionadas, para todos os níveis de conhecimento da língua inglesa, de acordo com os cursos de mesmo nome, *Elementar*, *Pré-intermediário*, *Intermediário* e *Superior Intermediário*, além de bônus na forma de palavras básicas (substantivos, adjetivos, números e verbos) (SHISHAKIN, 2021). A principal característica desta ferramenta são os períodos de repetição de palavras de maneira configurável. A ideia é que se a palavra foi aprendida nos exercícios de “teste” (deve ser feito com 5 estrelas) ou “escreva você mesmo” (deve ser feito com 4 estrelas), todas as palavras corretamente escolhidas e escritas serão consideradas aprendidas. Assim, estas palavras desaparecerão do treinamento por um período de 2

horas, retornando na sequência. Ao memorizar novamente, a palavra desaparecerá por um dia, depois por uma semana, por 2 meses e até por um ano, de acordo com as configurações. A principal vantagem deste aplicativo inclui o suporte para 79 idiomas de um conjunto de palavras padrão, enquanto a desvantagem está relacionada ao fato do usuário não poder adicionar novas palavras à ferramenta. Embora a ferramenta propicie um ambiente para o aprendizado, são utilizados *cards* com palavras que não necessariamente fazem parte do cotidiano do usuário, sendo usadas em poucas ocasiões, problema que favorece o esquecimento da palavra e tira o foco de palavras mais próximas e adaptadas à realidade do usuário.

3 METODOLOGIA

Neste Capítulo é apresentada a proposta de metodologia utilizada na concepção do aplicativo a fim de realizar o desenvolvimento de uma aplicação mobile interativa destinada ao aprendizado de idiomas.

Para tanto, a metodologia utilizada na realização deste trabalho está dividida em duas etapas principais:

- *Análise das API's de processamento de imagens na nuvem*: Contém os processos relacionados ao funcionamento e análise de cada *API*, possui informações sobre a base de dados, o fluxo de implementação de cada plataforma e sobre a captação dos resultados.
- *Metodologia do estudo de caso*: contém a metodologia referente ao estudo de caso, que refere-se ao desenvolvimento do aplicativo. Esta etapa possui detalhes sobre o diagrama de classes, requisitos funcionais e não funcionais, e procedimentos para avaliação e validação.

3.1 ANÁLISE DAS API'S DE PROCESSAMENTO DE IMAGENS NA NUVEM

3.1.1 Protocolo para avaliação das *API's*

O protocolo para a avaliação de desempenho das *API's* segue o seguinte processo:

- *Coleta do conjunto de dados*: as imagens são coletadas de vários conjuntos de categorias para medir o desempenho dos serviços em nuvem propostos, bem como são selecionadas imagens de granulação fina e grossa;
- *Teste de imagens em serviços em nuvem*: para testar os serviços, as imagens contidas no conjunto de dados são enviadas para a *API* e os resultados referente ao nome dos rótulos e pontuação de confiança são capturados para análise;

- *Análise de dados e conclusões*: os resultados das soluções, como rótulos e pontuação de confiança, são comparados com a realidade das imagens. Esse processo forma a base de conclusões sobre a precisão dos serviços em questão. O desempenho de classificação é avaliado considerando as métricas descritas na Seção 3.1.5.

Cada item acima será explanado com mais detalhes nas seções seguintes.

3.1.2 Coleta do conjunto de dados

Para a condução dos experimentos são utilizadas 600 imagens encontradas no Google Imagens. As imagens foram selecionadas manualmente, respeitando os seguintes critérios de seleção:

- Imagens com dimensões distintas com relação às selecionadas;
- Imagens com diferentes resoluções e granulações;
- Prioridade para imagens que contém elementos que fazem parte do cotidiano.

Estes critérios de seleção foram elaborados para simular a diversidade de resoluções e formatos entre as câmeras de dispositivos móveis. Estas 600 imagens estão divididas em 10 categorias, resultando em 60 imagens por categoria¹. Para ambas as *API's*, não é necessário base de treinamento para utilizar o serviço de reconhecimento.

3.1.3 Metodologia para o envio das imagens pelo serviço Amazon *Rekognition*

Os serviços na AWS, como o AWS Lambda, exigem que sejam fornecidas credenciais ao acessá-los para garantir que tenha permissão para acessar os recursos pertencentes a esse serviço. Para fazer isso, utiliza-se o serviço recomendado pela AWS, conhecido como *AWS Identity and Access Management*. Este serviço é configurado na própria plataforma da Amazon.

¹ Não foi possível utilizar quantidade maior de imagens devido à limitação do plano gratuito disponibilizado por ambas plataforma

Após a configuração do serviço na Amazon, inicia-se a configuração do arquivo `serverless.yml` para conceder todas as permissões necessárias para o funcionamento correto da API. A estrutura *Serverless* é uma ferramenta de Command-line interface (CLI) que cria uma abstração sobre o AWS CloudFormation e automatiza todo o processo de configuração da infraestrutura em nuvem exigida pelas funções do projeto. Em uma linguagem muito simples, o arquivo *serverless.yml* é usado para codificar o *template* de acordo com os recursos que queremos criar na conta AWS, podemos definir diferentes tipos de recursos no arquivo `serverless.yml` e também podemos definir diferentes permissões para diferentes recursos. Para este projeto, foram configurados os seguintes recursos:

- Permissões: refere-se às autorizações para que a função *lambda* use o serviço AWS Rekognition para detectar a imagem, concede também permissões de leitura *S3* para ler a imagem carregada para identificação e, por último, permissão para publicar um evento. Dentro de permissões, é necessário configurar os seguintes recursos:
 - *action*: este bloco tem todas as permissões que queremos dar à função *lambda* para realizar diferentes operações. São utilizadas as permissões *s3:PutObject* e *s3:GetObject* para conceder a autorização para o Rekognition colocar e retirar a imagem no bucket. Utiliza-se também a permissão *rekognition:DetectLabels* para usar os recursos de detecção do *Rekognition*.
 - *provider*: onde define-se a configuração, variáveis de ambiente e papéis.
- Funções *lambda*: é necessário apenas uma única função *lambda* para este projeto, que é definida no bloco de funções.
- *Eventos*: o último passo é anexar um evento do *S3* à função *lambda* para que ele seja chamado assim que uma nova imagem for carregada no *bucket S3*, esse é o nome do bucket onde é feito o *upload* de imagens.

Após isso, é feita a configuração para disparar evento para criar um *bucket* chamado *devawsapidetect* no bucket S3.

Quando a função *lambda* é chamada por um evento do S3, ela recebe dados sobre o objeto que foi carregado no bucket S3. Nesta etapa são obtidos os dados do objeto, como o nome do *bucket* para o qual foi carregado e o nome do arquivo também. Na sequência, esses dados são passados para o objeto de parâmetro que é passado para a chamada da *API* do AWS *Rekognition*.

Após conceder as permissões iniciais, é feita a configuração para obter essa imagem no formato *base64*, pois o método da *API DetectLabels* requer um binário da imagem codificado em *base64*, sendo possível ao fazer o *parse* do *JSON*.

Em seguida, são criados dois parâmetros: um que serve para gravar o conteúdo dessa imagem em um bucket do S3, e outro parâmetro que serve para o *Rekognition* identificar uma imagem que vai estar no S3 dentro desse *bucket*.

Na sequência, cria-se uma instância do S3 com uma Promise responsável por gravar o conteúdo do arquivo no bucket criado. Posteriormente, é criada mais uma Promise que chama o *Rekognition*, responsável pela detecção de rótulos passando uma variável que captura no S3 a imagem gravada no processo anterior. O resultado dessa análise vai parar na variável que, por fim, retorna os rótulos detectados no corpo do *JSON*. Para enviar a imagem com facilidade, foi feita uma interface simples no Android Studio que permite o envio das imagens selecionadas na galeria de imagens do dispositivo móvel em questão.

3.1.4 Metodologia para o envio das imagens pelo serviço Google *Vision*

A primeira etapa para utilizar os serviços do Google *Vision* consiste em conectar-se aos serviços do Firebase. Para fazer isso, é necessário entrar no console do Firebase e criar um novo projeto. Em seguida, é adicionado o nome do pacote que se

encontra o projeto mobile que se localiza do arquivo *app/build.gradle* do Android Studio. Após esse procedimento o Firebase gera automaticamente um arquivo chamado *google-services.json*, que deve ser colocado na pasta *app/*.

Em seguida, adiciona-se as seguintes dependências do Google *Vision* ao arquivo gradle, que darão acesso aos recursos da plataforma:

```
implementation 'com.google.firebase:firebase-core:16.0.1'
implementation 'com.google.firebase:firebase-ml-vision:16.0.0'
implementation 'com.google.firebase:firebase-ml-vision-image-label-model:15.0.0'
```

Com estas configurações, obtém-se acesso aos métodos providos pelo próprio Google para enviar as fotos para análise. Após as configurações iniciais, foi feita uma interface simples para capturar fotos e escolher fotos da galeria. Para isso, utilizou-se o método *requestPermissions()* para solicitar as permissões *CAMERA* e *WRITE_EXTERNAL_STORAGE*.

O último passo consiste em converter as imagens em um Bitmap, encaminhar para a plataforma do Google *Vision* e capturar os resultados.

Para encaminhar a imagem utiliza-se o recurso *process*, responsável por processar a imagem escolhida, enquanto para capturar os resultados, utiliza-se o recurso *ImageLabel* fornecido pela plataforma.

Com estes procedimentos, foi possível enviar as imagens para a plataforma e recuperar os resultados obtidos.

3.1.5 Avaliação de dados

Nesta seção são detalhados os métodos de avaliação dos resultados retornados pelas *API's*.

Após a análise feita pelas *API's*, os dados retornados pelo *json* são anotados em planilhas distintas, sendo uma planilha dedicada a anotar o resultado obtido pela *API* do serviço Google *Vision* e a outra planilha contendo os resultados do serviço Amazon *Rekognition*. O serviço do Google Sheets foi utilizado para este fim.

Logo após, é feita a comparação entre os dados nas planilhas e a imagem na qual os dados se referem. Neste trabalho, o desempenho das diferentes soluções é definido como uma métrica agregada que avalia se um rótulo foi produzido corretamente. Todos os rótulos das imagens que resultaram de uma solução são comparados com a imagem enviada. Se uma correspondência for encontrada, um ponto é adicionado na pontuação de acertos dessa imagem. Se um rótulo não for detectado ou detectar erroneamente, um ponto é acrescentado na pontuação de erros. Para cada imagem, é feita a soma da quantidade total de acertos e posteriormente é feita a soma da quantidade de erros obtidos nos rótulos de cada imagem. Para cada imagem, também é feita a soma do total de rótulos retornados pela *API*.

Com estes resultados, é possível calcular a porcentagem de erros e acertos obtidos em cada categoria neste experimento, assim como a quantidade total de rótulos por categoria. A porcentagem de acertos é calculada realizando a divisão da quantidade total de acertos contidos naquela categoria pelo total de rótulos encontrados na mesma categoria em questão, enquanto a porcentagem de erros utilizou o mesmo processo de divisão, porém considerando a quantidade total de erros em vez de acertos. Após isso, é calculada a soma e o percentual total de erros e acertos para cada *API*.

Por fim, foram comparados aspectos específicos de comportamento de cada *API*.

3.2 METODOLOGIA DO ESTUDO DE CASO

3.2.1 Requisitos

Abaixo, são apresentados os requisitos funcionais (RF) e não funcionais (RNF) referentes a este projeto:

1. permitir o cadastro de um novo usuário (RF);
2. permitir a captura de uma foto com a câmera do dispositivo móvel (RF);
3. permitir a escolha de uma imagem armazenada no dispositivo (RF);

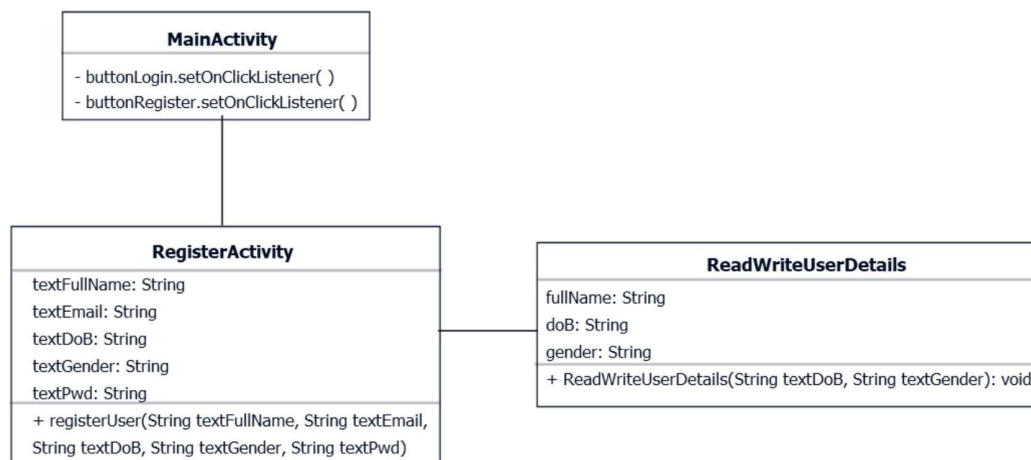
4. transmitir e processar a imagem selecionada utilizando a API mais adequada à este aplicativo de acordo com os experimentos realizados (RF);
5. exibir o resultado da análise da imagem ao usuário (RF);
6. traduzir os resultados retornados da *API* para o português e para o russo, conforme apontado nos objetivos específicos (RF);
7. utilizar a linguagem de programação Java para implementação, devido sua popularidade de utilização na plataforma Android (RNF);
8. utilizar a plataforma Firebase do Google para realizar a autenticação e cadastro do usuário (RNF).

3.2.2 Diagrama de Classes

O diagrama de classes permite visualizar como as classes do aplicativo estão estruturadas, assim como o seu relacionamento. Esta seção descreve as classes do aplicativo desenvolvido, nomeado de *Bazhenov*.

A primeira classe a ser acionada trata-se da *MainActivity*, esta classe oferece as opções do usuário logar ou se cadastrar. Caso o usuário opte pelo cadastro, a classe *RegisterActivity* é chamada. Nesta classe, o método responsável pelo cadastro trata-se do *registerUser*, que passa como parâmetro as informações declaradas pelo usuário nos campos de dados, que são captadas pela classe *ReadWriteUserDetails*. Abaixo, na Figura 5, encontra-se a imagem do fragmento da classe de registro de usuário.

Figura 5 – Fragmento das classes associadas ao cadastro do usuário



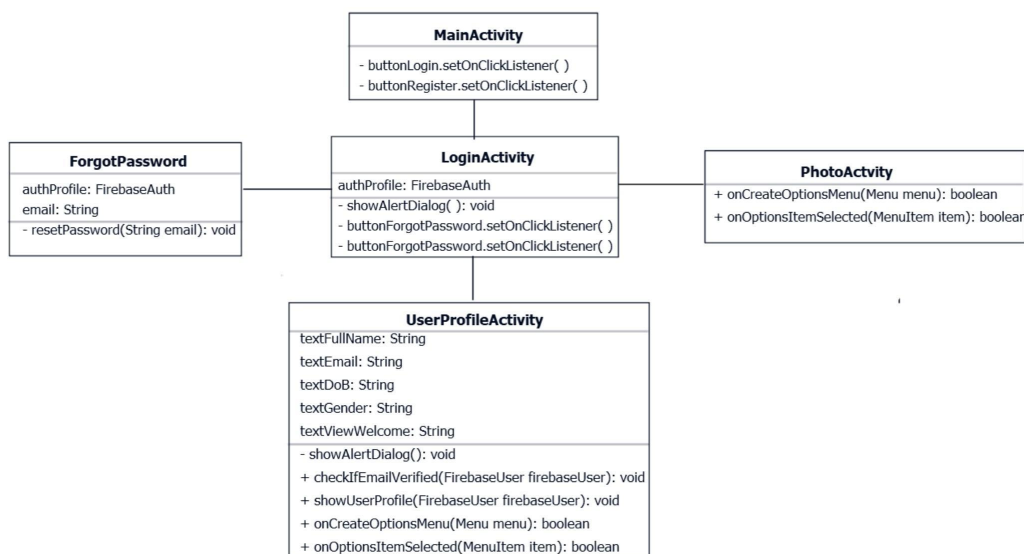
Fonte: Autor, 2022

A classe *LoginActivity*, que pode ser chamada a partir da *MainActivity*, é usada para o usuário se autenticar no sistema, sendo necessário o *email* e senha cadastrados para acessar a página principal do aplicativo. Após o login, a classe *PhotoActivity* é chamada, redirecionando o mesmo para a *Activity* principal do aplicativo.

A classe *ForgotPasswordActivity* é chamada para o fim de trocar a senha do usuário caso ele tenha esquecido da senha, opção encontrada na classe *LoginActivity*. Esta classe é responsável por mandar uma mensagem com um link para o *email* cadastrado pelo usuário.

Na Figura 6 abaixo encontra-se a imagem do fragmento relacionado à classe *LoginActivity*.

Figura 6 – Fragmento das classes associadas ao login



Fonte: Autor, 2022

Em seguida ao login, a classe *PhotoActivity* é chamada, essa classe é responsável apenas por configurar o *recyclerView*, que tem como função listar os resultados quando feita a identificação da imagem. Após isso, a classe *ImageClassificationActivity* é chamada, nesta classe o usuário poderá escolher se deseja capturar uma foto com a câmera ou se deseja escolher uma foto da galeria. Posteriormente, a foto é enviada para a *API*, que faz a análise e retorna os dados no *recyclerView*.

A classe *UserProfileActivity* é responsável pela exibição de algumas informações cadastradas pelo usuário, como nome, email, data de nascimento e gênero. Ainda na seção *UserProfileActivity* é possível atualizar a foto de perfil. Para isso é chamada a classe *UploadProfilePictureActivity*, responsável por esta ação.

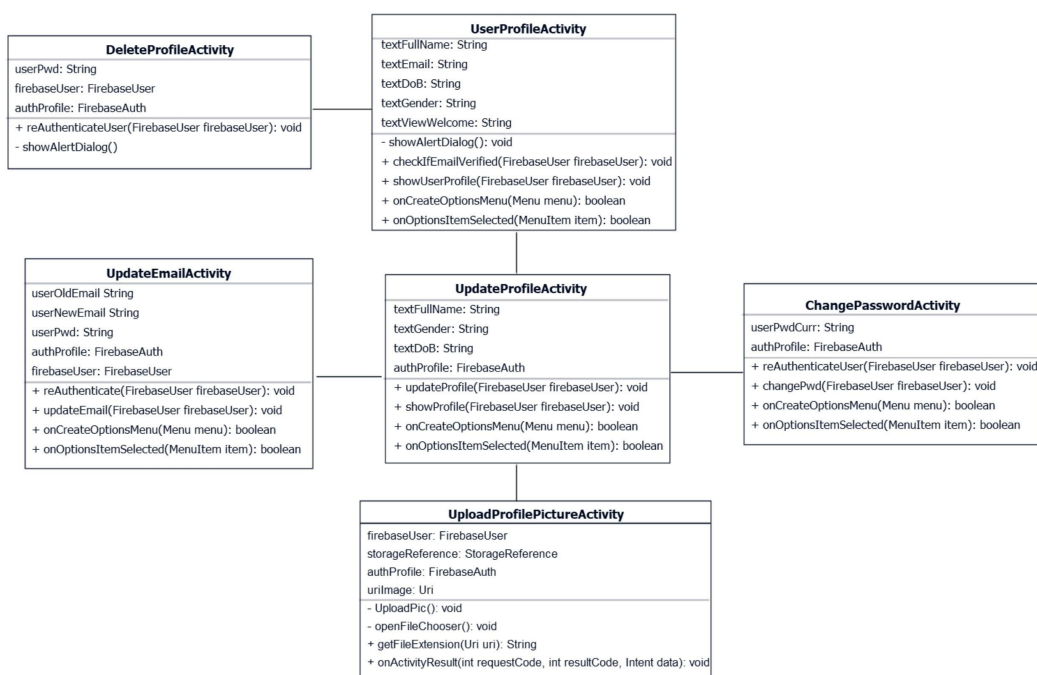
A classe *UploadProfileActivity* é responsável por atualizar as informações do perfil do usuário, através dela é possível atualizar o nome, data de nascimento e gênero do usuário. Através dela, é possível acessar a classe *UpdateEmailActivity* que tem como função atualizar o *email* do usuário com o método *updateEmail*. Para alterar a senha existe a classe *ChangePasswordActivity*, no qual o método *changePwd* se

encarrega de executar a tarefa.

A classe *DeleteProfileActivity* é responsável por deletar o perfil do usuário e todos os dados relacionados a ele. Nesta classe, encontra-se o método *showAlertDialog* encarregado de abrir uma caixa de diálogo informando o usuário que o perfil será deletado, sendo este processo irreversível.

Na Figura 7 abaixo, encontra-se o fragmento das classes relacionadas ao usuário.

Figura 7 – Fragmento das classes associadas ao usuário



Fonte: Autor, 2022

Para mais informações associadas à funcionalidade dos métodos de cada classe, consultar o Apêndice A.2

3.2.3 Metodologia referente ao Teste de Usabilidade

Este é um método de teste que visa estabelecer o grau de usabilidade, capacidade de aprendizado, compreensibilidade e atratividade para os usuários do produto desenvolvido no contexto de determinadas condições. O teste de software é necessário para verificar se o produto de software implementado atende às expectativas.

A tarefa do teste de usabilidade é identificar falhas estruturais e visuais na interface, além de verificar a facilidade de uso e o atendimento aos requisitos funcionais da aplicação. O teste de usabilidade é um teste importante para todos os serviços que interagem com o usuário para garantir que esteja em conformidade com os padrões aceitos e seus requisitos, verificando, por exemplo, como são exibidos os elementos da interface (texto, botões, menus, listas e outros elementos) e verificando a experiência do usuário.

Para o teste, foram preparadas tarefas de usabilidade que alguns voluntários precisavam concluir. Após a conclusão da usabilidade do sistema, foi proposto avaliar através de um formulário a conveniência de concluir tarefas em uma escala de cinco pontos, onde 5 - conveniente e intuitivo, 1 - não consegui fazer. Caso houvesse dúvidas, os voluntários foram aconselhados a entrar em contato para solucionar a questão.

Tarefas a cumprir:

1. Fazer cadastro no sistema.
2. Fazer login no sistema.
3. Selecionar foto da galeria para análise da *API*.
4. Capturar foto para análise da *API*.
5. Alterar informações da conta.
6. Sair da conta.
7. Deletar conta

O formulário de questões foi projetado para validar com notas os seguintes pontos:

- Cinco perguntas com atribuição de nota para verificar a aceitação referente aos processos relacionados ao usuário (cadastro, autenticação, edição do perfil, sair e deletar conta).
- Cinco perguntas com atribuição de nota relacionadas ao funcionamento geral do aplicativo (captura e escolha de imagens, tradução das palavras, e experiência geral).

As notas vão de 1 a 5, sendo 1 a nota mínima e 5 a nota máxima. O formulário também lista possíveis erros que poderiam acontecer durante o uso do aplicativo e, caso o usuário presencie algum problema, ele pode selecionar qual o problema enfrentado e detalhar mais caso seja necessário. O formulário também coleta dados do usuário como idade e o modelo do celular utilizado no teste, sendo este último dado importante para avaliar erros específicos que podem ocorrer. A última seção do formulário está destinada a sugestões dos usuários. A plataforma escolhida para realizar os procedimentos declarados nesta seção foi o Google *Forms*.

Se o teste de usabilidade revelar alguma dificuldade (por exemplo, dificuldades em entender instruções, executar ações ou interpretar respostas do sistema), mudanças devem ser aplicadas ao programa para solucionar os erros.

3.2.4 Metodologia referente à Validação

Para que o aplicativo seja considerado minimamente adequado aos usuários, deve ter um mínimo de aprovação por parte dos mesmos. Para tal, é feita a média das notas das respostas obtidas para cada pergunta no procedimento anterior. Este resultado trata-se da média por pergunta. Após calcular a média por pergunta, esses mesmos valores são somados e divididos por 10, que corresponde ao total de questões com atribuição de nota. O aplicativo pode ser considerado apto caso a média de aprovação geral seja superior a 3,5.

4 EXPERIMENTOS COMPUTACIONAIS

Nesta seção são descritos os experimentos computacionais realizados. Os experimentos são divididos em dois momentos:

- *Avaliação de serviços de processamento de imagens na nuvem*: contém as informações relacionadas os experimentos conduzidos nas API'S;
- Estudo de caso: contém os experimentos relacionados ao desenvolvimento do aplicativo proposto.

4.1 EXPERIMENTO #1: AVALIAÇÃO DE SERVIÇOS DE PROCESSAMENTO DE IMAGENS NA NUVEM

Nesta Seção são apresentados os experimentos que foram conduzidos em relação às *API's* do Google *Vision* e Amazon *Rekognition*. Estes experimentos tem como objetivo verificar a *API* com o melhor desempenho e tentar identificar qual poderá melhor se adequar ao problema proposto, considerando apenas as funções básicas para o funcionamento do aplicativo. A ideia é que a escolha da *API* de processamento de imagens na nuvem, para uso no aplicativo, seja realizada com base nos resultados obtidos no experimento.

Para o processamento das imagens, foram criados *scripts* na linguagem Java, sendo desenvolvido para o Android, com finalidade de fazer *upload* das imagens, consumindo as *API's* de ambos Amazon *Rekognition* e Google *Vision*. Todas as 600 imagens da coleção de testes foram submetidas às *API's* diretamente pelo aparelho celular, onde o retorno se dá em formato *JSON* contendo o rótulo identificado e a respectiva pontuação de confiança relacionada a cada rótulo identificado no range 0..100%. Para análise, considerou-se a pontuação de confiança a partir de 30%.

4.1.1 Avaliação do modelo

Para cada categoria, somou-se a quantidade de erros e acertos obtidos nos rótulos de cada imagem e então foi calculado os respectivos percentuais. A Tabela 1 apresenta o resultado dos testes por cada categoria de imagens referente ao Google *Vision*, enquanto a Tabela 2 mostram os resultados relacionados ao Amazon *Rekognition*.

Tabela 1 – Google Vision: Resultado por categorias

Categoria	total Rotulos	qtd Acertos	qtd Erros	% Acertos	% Erros
Animal	391	228	163	58,31%	41,69%
Comida	599	405	194	67,61%	32,39%
Construções	833	660	173	79,23%	20,77%
Cozinha	484	336	148	69,42%	30,58%
Escola	467	310	157	66,38%	33,62%
Instrumento Musical	342	267	75	78,07%	21,93%
Quarto	389	251	138	64,52%	35,48%
Rua	815	636	179	78,04%	21,96%
Sala	466	288	178	61,80%	38,20%
Veículo	558	333	225	59,68%	40,32%
Total	5344	3714	1630	69,50%	30,50%

Fonte: Autor, 2022

Tabela 2 – Amazon *Rekognition*: resultado por categorias

Categoria	total Rotulos	qtd Acertos	qtd Erros	% Acertos	% Erros
Animal	555	360	195	64,86%	35,14%
Comida	606	401	205	66,17%	33,83%
Construções	850	634	216	74,59%	25,41%
Cozinha	621	405	216	65,22%	34,78%
Escola	573	348	225	60,73%	39,27%
Instrumento Musical	447	273	174	61,07%	38,93%
Quarto	538	354	184	65,80%	34,20%
Rua	939	735	204	78,27%	21,73%
Sala	600	407	193	67,83%	32,17%
Veículo	567	338	229	59,61%	40,39%
Total	6296	4255	2041	67,58%	32,42%

Fonte: Autor, 2022

Examinando os resultados de saída produzidos pelos serviços e mostrado nas tabelas acima, é possível ter uma ideia geral dos conceitos que os serviços reconhecem com maior ou menor precisão. Os maiores índices de acerto em ambas tecnologias encontram-se em categorias relacionadas ao ar livre, como “ruas” e “construção”, enquanto as outras possuem índices mais baixos.

Mais especificamente em relação ao serviço do Amazon *Rekognition* foi possível observar *tags* de detecção de casos específicos, como por exemplo, rotular a raça de animais, nome de flores e marca de carros, enquanto o Google *Vision* detectava apenas rótulos gerais na imagem. Entretanto não foi observado um índice de acerto relevante quanto à detecção de rótulos específicos do Amazon *Rekognition*, incrementando a quantidade de erros. Pode-se observar também uma maior detecção de rótulos por parte do serviço da Amazon se comparado com o serviço oferecido pelo Google, e, apesar deste resultado, a taxa de acertos da Amazon foi inferior ao da Google.

Observou-se nos experimentos que os rótulos retornados pela *API* da Amazon tinham maior relação com sinônimos e *tags* relacionadas, enquanto os rótulos

da *API* da Google forneciam maior diversidade de dados. Por exemplo, ao enviar a imagem da Figura 8, que continha cães, o serviço da Amazon fez a detecção de alguns elementos conforme observa-se na Tabela 3, ou seja, o serviço detectou que haviam cães e, com isso, trouxe diversas *tags* relacionadas a cães. Os primeiros resultados retornados pela *API* (“mamífero”, “animal doméstico”, “canino”, “animal” e “animal de estimação”), embora corretos, tratam-se apenas de *tags* relacionadas, enquanto que o serviço da Google, conforme pode ser observado na Tabela 4 detectou “rede”, “metal” e “gaiola”, rótulos não relacionados diretamente a animais porém que contém na imagem. Este padrão de comportamento se repetiu nas outras imagens, demonstrando que o índice de acertos do *Amazon Rekognition*, embora similar ao serviço do concorrente, não oferece maior precisão na detecção de objetos contidos na imagem.

Figura 8 – Imagem enviada para a análise das *API'S*



Fonte: Freepick, 2021

Tabela 3 – Amazon Rekognition: resultado referente à imagem acima

nomeImagem	nomeRotulo	porcentConfianca
9	Husky	94,55%
9	Cão	94,75%
9	Animal de estimação	94,39%
9	Canino	94,01%
9	Mamífero	93,55%
9	Animal	92,35%
9	Leão	91,20%
9	Vida Selvagem	91,20%
9	Ao ar livre	55,09%

Fonte: Autor, 2021

Tabela 4 – Google Vision: resultado referente à análise da imagem acima

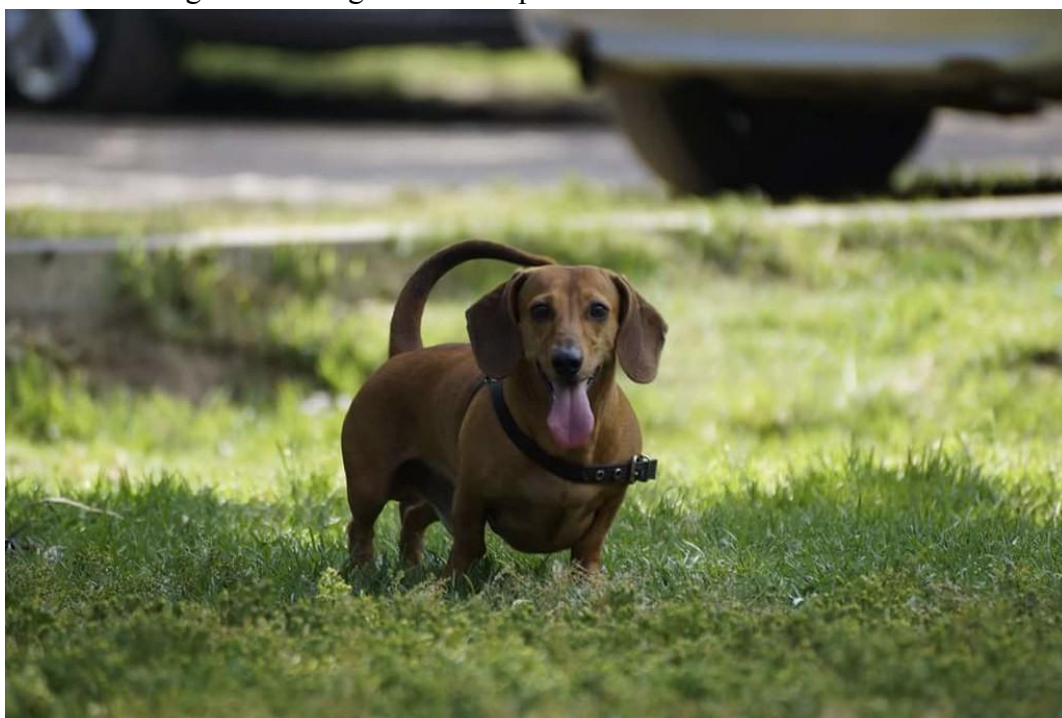
nomeImagem	nomeRotulo	porcentConfianca
9	Cachorro	92,93%
9	Pet	75,92%
9	Gato	66,16%
9	Pássaro	58,85%
9	Gaiola	53,97%
9	Metal	51,80%
9	Rede	48,45%
9	Pelo	46,35%
9	Cavalo	32,71%

Fonte: Autor, 2021

Na imagem acima também pode ser observado um padrão de comportamento relacionado à distribuição das pontuações de certeza por cada *API*. O padrão está

relacionado que os resultados com maior pontuação de confiança trazidos pelo Amazon Rekognition oferecem mais precisão do que os resultados com maior pontuação de confiança do Google *Vision*. Sendo os primeiros resultados da Amazon geralmente corretos em relação ao restante dos rótulos, enquanto no serviço da Google diversos rótulos são classificados com baixa pontuação de confiança, embora estejam tão corretos quanto aqueles classificados com maior pontuação de confiança. Abaixo podemos ver um exemplo desta ocorrência, a imagem apresentada na Figura 9 foi enviada para a análise das *API's*, e nas Tabelas 5 e 6 encontram-se os resultados.

Figura 9 – Imagem enviada para a análise de ambas as *API's*



Fonte: Ibaceta, 2016

Tabela 5 – Google vision: Resultado referente à análise da imagem acima

nomeImagem	nomeRotulo	porcentConfianca
2	Cão	99,32%
2	Animal de estimação	97,39%
2	Roda	63,56%
2	Pára-choques	56,03%
2	Pneu	50,98%
2	Cadeira	48,11%
2	Metal	47,20%
2	Veículo	45,19%
2	Pássaro	37,58%
2	Planta	37,31%
2	Bicicleta	31,19%
2	Lazer	30,96%

Fonte: Autor, 2021

Tabela 6 – Amazon Rekognition: Resultado referente à análise da imagem acima

nomeImagem	nomeRotulo	porcentConfianca
2	Grama	99,69%
2	Planta	99,69%
2	Cão	99,03%
2	Canino	99,03%
2	Animal de estimação	99,03%
2	Animal	99,03%
2	Mamífero	99,03%
2	Correia	90,41%
2	Pneu	75,29%
2	Ao ar livre	73,87%
2	Gramado	69,78%
2	Filhote	68,85%
2	Pétala	65,97%
2	Flor	65,97%
2	Beagle	59,17%
2	Trela	56,21%

Fonte: Autor, 2021

4.1.2 Conclusão do experimento

O resultado dos experimentos em relação aos serviços de identificação de imagens na nuvem não são abertamente positivos, pois a média de acerto das *API's*, para as imagens selecionadas neste experimento, foi de aproximadamente 68%. Em alguns casos ocorria de ambas as *API's* identificarem erroneamente as imagens por vezes seguidas. Considerando que a proposta deste trabalho trata-se do desenvolvimento de um aplicativo com base nestes resultados, os resultados sugerem que a experiência do usuário poderá ser um pouco afetada pela qualidade da detecção obtida pelo serviço de identificação de imagens. Entretanto como os resultados estão acima de 60%, pode-se considerar viável o desenvolvimento do aplicativo, visto que melhorias em trabalhos futuros podem ser aplicadas para ampliar o percentual de acerto no reconhecimento das imagens.

Estabelecendo um comparativo entre os resultados apresentados, os testes indicaram que Google *Vision* apresentou um percentual de acertos relativamente maior em relação ao serviço Amazon *Rekognition*. De acordo com os dados obtidos pelo experimento, pode-se concluir as seguintes concepções:

Pontos positivos do serviço Google *Vision*:

- Capaz de detectar com precisão relativamente maior, conforme pode-se observar na Figura 10;
- Capaz de detectar maior variedade de elementos em uma imagem do que o serviço da Amazon.

Como ponto negativo do serviço Google *Vision*, verifica-se que oferece resultados condizentes com a imagem distribuídos de forma menos homogênea nos percentuais de confiança. Muitos resultados relevantes para a imagem são classificados com baixa porcentagem de confiança, sendo que em alguns casos tags relevantes não chegam a 50% de confiança.

Pontos positivos do serviço Amazon *Rekognition*:

- Maior facilidade para filtrar resultados positivos relevantes. A porcentagem de confiança maior traz resultados muito mais positivos e precisos do que o serviço da Google.
- Faz detecção de *tags* específicas;

Pontos negativos do serviço Amazon *Rekognition*:

- Menor precisão de acertos comparados com o serviço da Google, conforme observado também na Figura 10;
- Erros frequentes relacionados às tags específicas;
- Retorna quantidade elevada de *tags* com sinônimos em torno do objeto principal reconhecido.

Figura 10 – Comparação do percentual total de acertos e erros referente a cada *API*



Fonte: Autor, 2022

4.2 EXPERIMENTO #2: ESTUDO DE CASO

Por fim, como parte dos experimentos, propõe-se um estudo de caso com o objetivo de avaliar a proposta metodológica na prática. A ideia foi a de desenvolver um aplicativo funcional utilizando a plataforma Android e, então, realizar o estudo de caso voltado para o ensino-aprendizado do idioma russo. Assim, alguns aspectos de implementação são necessários verificar.

4.2.1 Aspectos de implementação na plataforma Android

A plataforma Android foi escolhida para o desenvolvimento do aplicativo deste caso de uso devido ao fato de ser o sistema operacional móvel com maior quantidade de usuários na atualidade e também pelo código-fonte ser aberto e distribuído gratuitamente, proporcionando atualizações mais rápidas e acesso menos restrito do que um sistema fechado como sistema da Apple, conhecido como IOs.

Um componente importante no desenvolvimento de uma aplicação é a escolha de um IDE (ambiente de desenvolvimento integrado) adequado, que devido o *Android Studio* ser o IDE oficial para o sistema operacional Android foi a IDE escolhida.

O aplicativo necessitará de um banco de dados, sendo utilizado neste trabalho o Firebase, que é um banco de dados *NOSQL* baseado em nuvem que permite aos usuários guardar e recuperar informações armazenadas (FIREBASE, 2021), proporcionando ferramentas e métodos convenientes para interagir com ele. O Firebase armazena dados de texto no formato *JSON* e fornece métodos simples para ler, atualizar e recuperar dados. Além disso, pode ajudar com o registro e autorização de usuários, armazenamento de sessões (usuários autorizados) e arquivos de mídia que são facilmente acessíveis devido ao *Cloud Storage*.

Demais detalhes de implementação podem ser consultados no Apêndice A.1.

4.2.2 Fluxo das interfaces e escolhas de design

Esta seção tem como propósito elucidar sobre o fluxo das interfaces, a função dos componentes implementados e as escolhas de *design*.

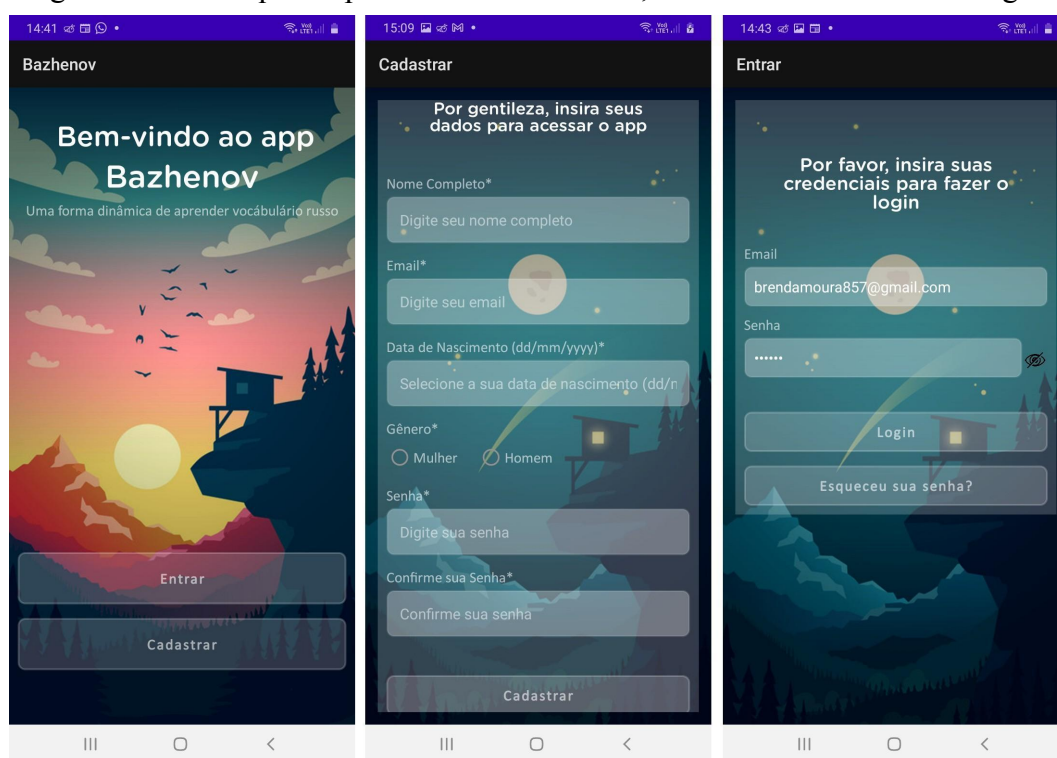
O design deste aplicativo visa uma experiência intuitiva para o usuário, para isso é respeitada a integridade estética, que refere-se a quão bem a aparência de um aplicativo combina com seus recursos. Por exemplo, um processo do aplicativo que envolve uma tarefa crítica pode manter o usuário mais atento ao utilizar elementos e textos com cores vibrantes Silva, Gama e Gonçalves (2017). Foi pensado também na consistência do aplicativo, que relaciona-se com a implementação de padrões familiares e paradigmas usando elementos de interface fornecidos pelo sistema, ícones conhecidos, estilos de texto padrão e terminologia comum. Referente a alguns elementos interativos, como botões, eles são destacados brevemente quando clicados, e barras de progresso comunicam o status das operações de longa duração (AXELSSON, 2012).

Assim que o usuário abre o aplicativo, a tela inicial é chamada. Nela, conforme pode se observar na primeira figura à esquerda na Figura 11, são exibidos o nome do aplicativo acompanhado de uma breve descrição logo abaixo esclarecendo a função na qual a aplicação se propõe, a fonte utilizada na descrição possui tamanho menor que o título para destacar o nome do aplicativo. Logo abaixo encontram-se os botões de login e cadastro, juntamente com um plano de fundo que envolve a tela inteira. Observa-se que algumas telas se trata da implementação de um design minimalista, evitando excesso de informações e poluição visual, sendo esses dois pontos considerados problemas que podem afetar a experiência do usuário (JÚNIOR, 2016). Importante mencionar que o aplicativo foi programado para exibir na barra superior o nome das respectivas telas em que o usuário se encontra, permitindo melhor esclarecimento sobre o propósito da tela em questão.

Caso seja um novo usuário, ele pode optar por realizar o cadastro, que o levará para a tela de cadastro. Nesta tela, conforme pode ser observado na figura central da Figura 11 o usuário deve preencher os dados solicitados, sendo esses dados o

nome completo, *email*, senha, data de nascimento e gênero. Um usuário que já se cadastrou no aplicativo pode ir para a tela de login ao clicar no botão “Entrar” da tela principal usando a tela de *login*. O usuário também pode escolher a opção “Esqueci minha senha”. A interface foi construída com o plano de fundo em tons escuros visando proporcionar menor sensação de cansaço, visto que exigem menos dos olhos. Todas as imagens usadas neste aplicativo foram encontradas na Internet, em distribuição gratuita.

Figura 11 – Da esquerda para direita: tela inicial, tela de cadastro e tela de login



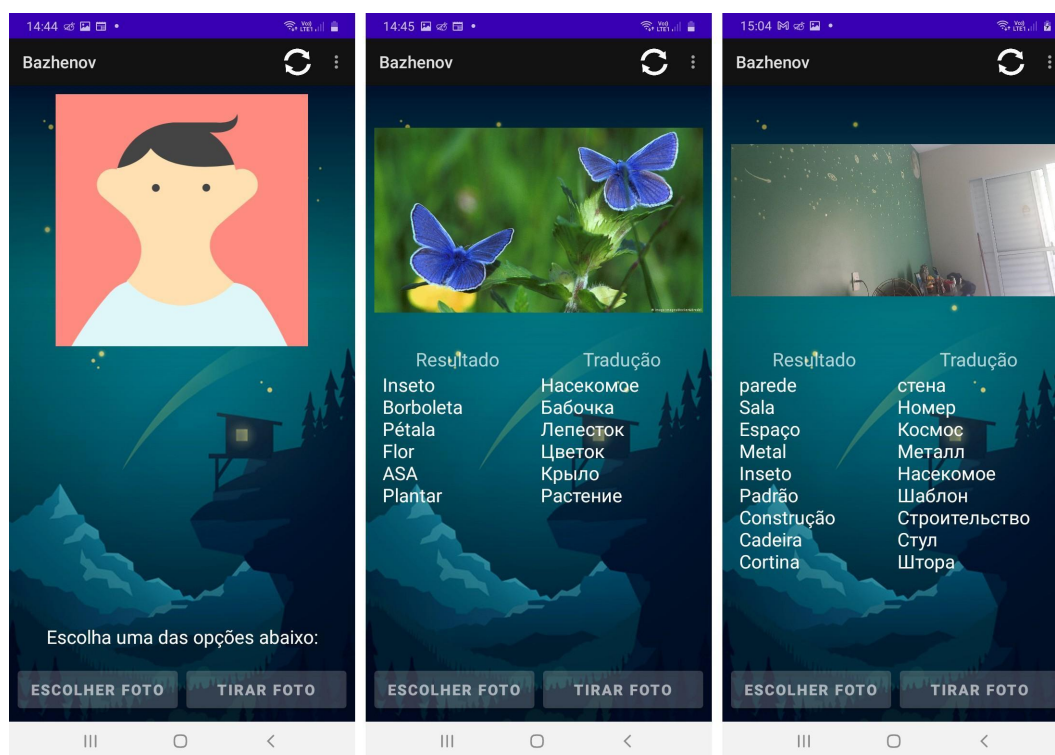
Fonte: Autor, 2022

Após a autorização bem sucedida, é chamada automaticamente a tela para escolha e captura de foto, pois corresponde à função principal do aplicativo. Nesta tela, conforme pode ser visto na Figura 12, existem dois botões dispostos horizontalmente, o primeiro com função de abrir a galeria de imagens do celular do usuário para que o mesmo escolha uma foto contida na galeria, enquanto o outro botão se encarrega de abrir a câmera traseira do usuário, permitindo que ele capture uma

foto.

Após a seleção da imagem ou captura da foto, é feito o envio da mesma para a *API* para reconhecimento, os resultados retornados (em inglês) são capturados pela *API* de tradução de idiomas que faz o procedimento de tradução para o português e para o russo. Estes resultados são armazenados e, posteriormente, dispostos em duas colunas na tela. A primeira coluna corresponde aos resultados em português e a segunda coluna exibe os resultados em russo. Na parte superior da tela mostrada na Figura 12 encontra-se a foto enviada para a *API*. A imagem mostrada na parte superior foi colocada para que o usuário confirmasse com maior facilidade quais rótulos retornados correspondem à foto enviada.

Figura 12 – Da esquerda para direita: tela principal, tela com resultados de foto da galeria e tela com resultados de imagem tirada com celular



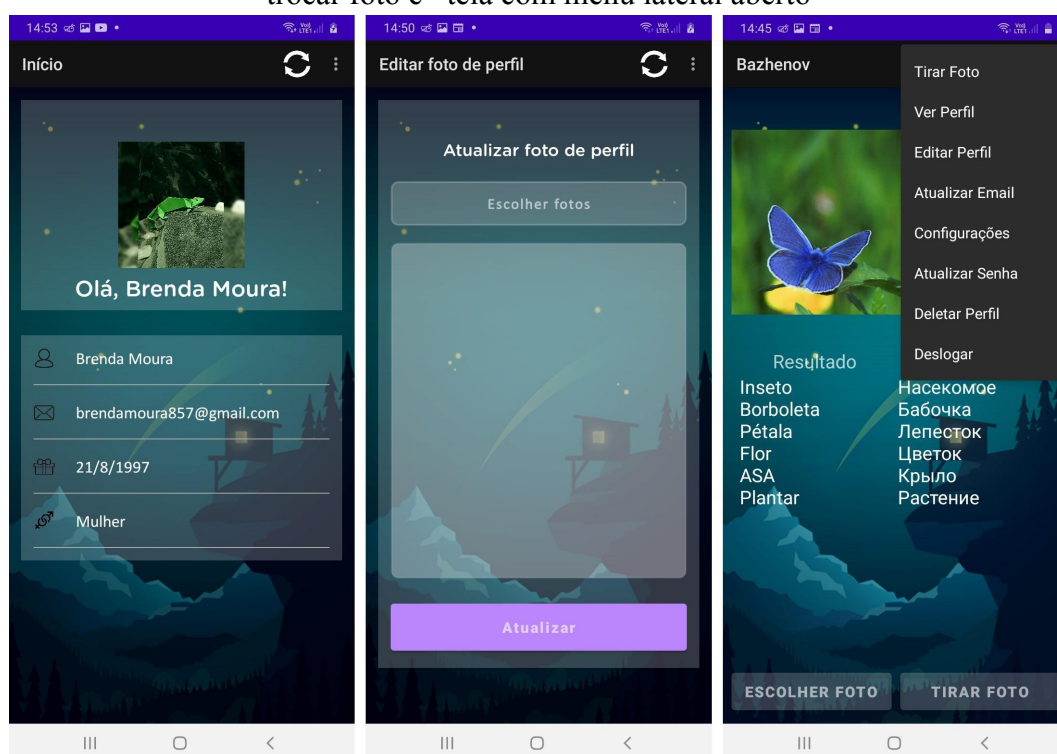
Fonte: Autor, 2022

As telas do perfil do usuário podem ser acessadas pressionando os três pontos localizados na parte superior direita do aplicativo, conforme mostrado na imagem à

direita na Figura 13. Na tela de visualizar o perfil, são exibidos na tela a foto do usuário, seguido logo abaixo por uma saudação acompanhada do nome do usuário. Nesta tela também podem ser encontrados o nome, *email*, data de aniversário e gênero cadastrados.

Ainda na tela de visualizar perfil, o usuário pode alterar a foto de perfil pressionando o espaço em que se localiza a foto de perfil. Feito isso, uma tela que permitirá a escolha e atualização da foto é aberta, ao pressionar o botão “escolher foto” a galeria de fotos do celular se abrirá, permitindo a seleção da foto desejada. Esta foto escolhida é então exibida no espaço abaixo do botão “escolher foto”. Em seguida, o usuário pode pressionar o botão atualizar. Ao realizar esse procedimento, ele é retornado automaticamente para a tela de visualização de perfil, bastando pressionar o botão de atualizar que se localiza ao lado esquerdo do botão de menu.

Figura 13 – Da esquerda para direita: tela de visualizar perfil, tela para trocar foto e tela com menu lateral aberto



Fonte: Autor, 2022

O usuário pode escolher no menu a opção de atualizar *email*, basta o

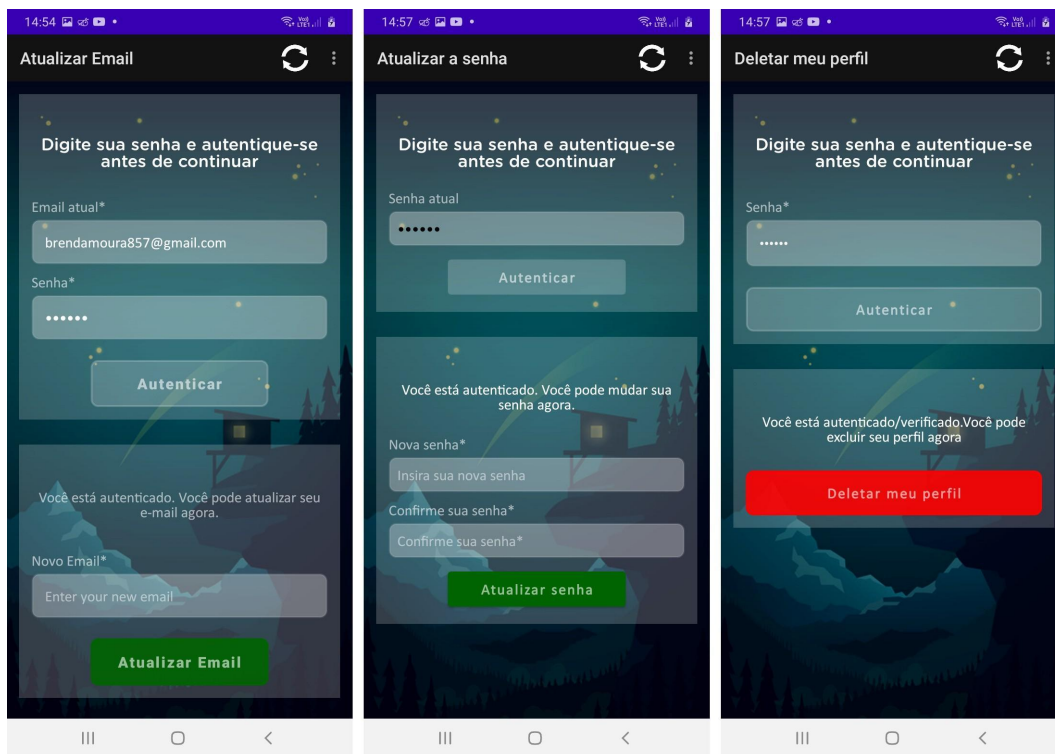
mesmo confirmar as credenciais para que a opção de atualização se habilite. O usuário saberá quando o botão de atualizar está habilitado no momento que o botão tornar-se da cor verde. A escolha da cor verde se deve pois, em *User Experience*, esta cor está associada a operações não críticas, realizadas com sucesso e segurança (SILVA; GAMA; GONÇALVES, 2017).

Na tela de atualizar senha, a mesma lógica se aplica quanto à autenticação e à cor do botão, indicando se o botão está habilitado ou não.

No menu, ainda podemos encontrar a opção de *deslogar* que, ao ser pressionada, encerra a instância do usuário levando-o automaticamente para a tela de início.

Por fim, existe a opção de deletar o perfil, para deletar o perfil é necessário autenticar-se. Após a autenticação, o botão deletar perfil torna-se da cor vermelho sólido. A cor vermelha em *User Experience* é usada em ações que envolvem criticidade, urgência e erros (SILVA; GAMA; GONÇALVES, 2017), por esta razão foi usada nesta ocasião. Quando o usuário pressiona o botão deletar perfil, uma janela é aberta informando que a operação é irreversível, dando a opção para o usuário continuar com a operação ou cancelar. Caso o usuário opte por continuar, as informações associadas a ele serão deletadas e o usuário será redirecionado para a tela inicial. Abaixo encontra-se a Figura 13 ilustrando as telas mencionadas.

Figura 14 – Da esquerda para direita: tela de atualização do *email*, tela de atualização da senha e tela para deletar o perfil

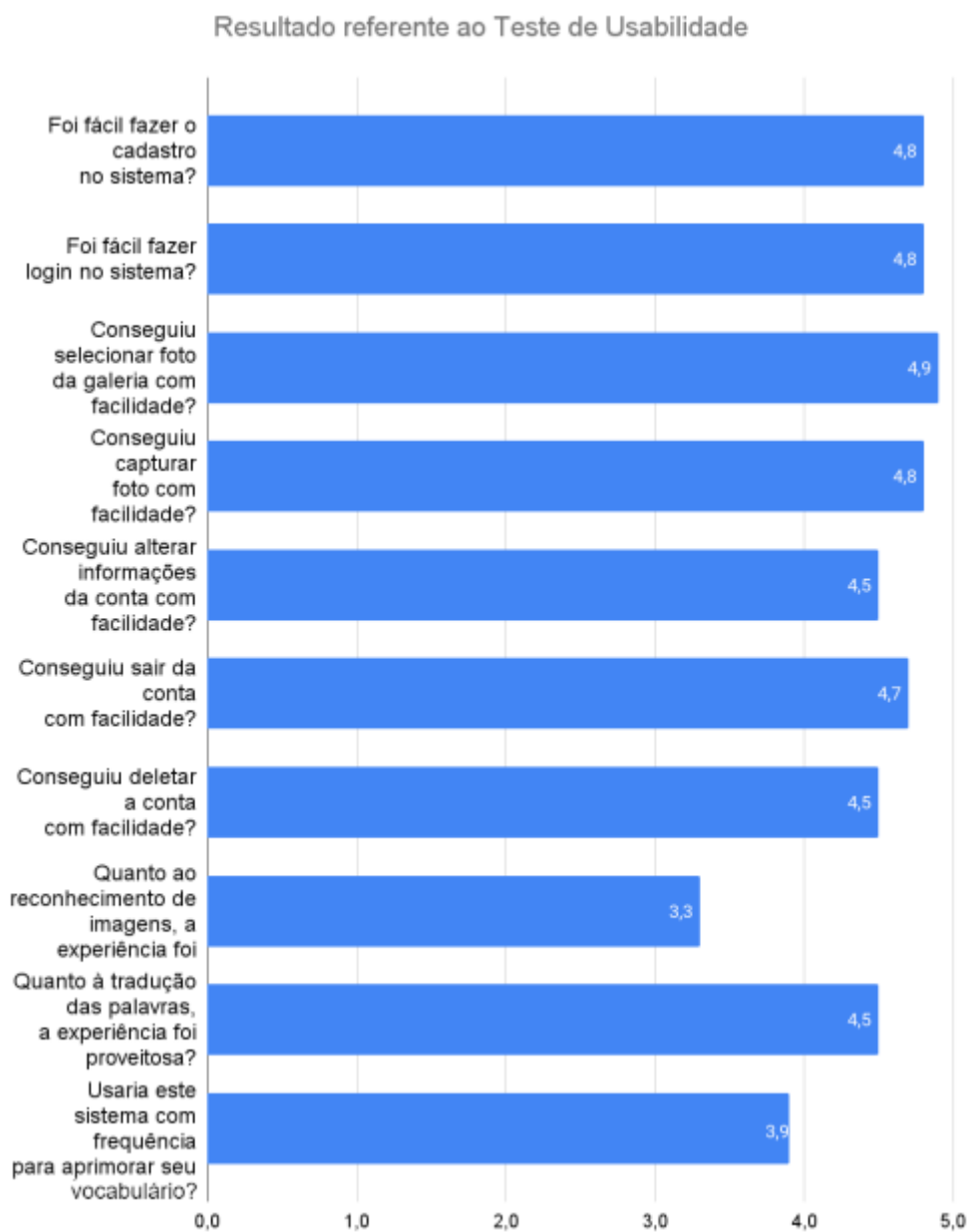


Fonte: Autor, 2022

4.2.3 Teste de Usabilidade

No total, 11 voluntários participaram do procedimento, este grupo possuía faixa etária de 22 à 45 anos. Os resultados do teste são apresentados no gráfico da Figura 15.

Figura 15 – Resultado do Teste de Usabilidade



Fonte: Autor, 2022

De acordo com os resultados do teste, a classificação média geral de satisfação é de aproximadamente 4,5, tratando-se de um resultado agradável considerando que a nota máxima é 5, indicando que não leva muito tempo para entender a lógica de

trabalho no aplicativo. De acordo com os resultados do formulário, a implementação das principais funções do aplicativo é intuitiva e não causa dificuldades para a maioria dos usuários. Entretanto, percebe-se que a categoria com menor avaliação refere-se ao reconhecimento de imagens. Este fato ocorre pois, conforme demonstrado nos testes apresentados na seção de experimentos, a média de acertos da *API* utilizada é suficientemente alta para construir o aplicativo e obter satisfação mínima do usuário, porém para resultados melhores, outros recursos podem ser aplicados, conforme explicado na seção de Trabalhos Futuros.

5 CONCLUSÃO E TRABALHOS FUTUROS

Observa-se que diversos aplicativos voltados ao ensino e educação estão disponíveis atualmente, incluindo para o ensino de idiomas. Contudo, verifica-se que os aplicativos voltados especificamente para o ensino de vocabulário carecem de interatividade, sendo este problema o principal motivo para o usuário sentir-se desmotivado a continuar utilizando o aplicativo de aprendizado. Desta forma emerge a ideia desta proposta, que visa a criação de um aplicativo que utiliza tecnologias em nuvem para proporcionar maior interatividade durante o aprendizado de vocabulário de um idioma estrangeiro.

Para utilizar o serviço em nuvem, considerou-se os serviços de IA e aprendizado de máquina em grande escala oferecidos pela AWS e Google. Neste sentido, foram conduzidos experimentos com o objetivo de verificar a melhor escolha de plataforma considerando o desenvolvimento do aplicativo em questão. Como resultado dos experimentos, foi possível concluir que a melhor opção de API a ser utilizada neste trabalho foi o Google Vision, embora ambas API's demonstrem eficácia de apenas aproximadamente 68%.

Apesar dos resultados das API's, o objetivo deste trabalho foi alcançado, ou seja, foi possível ser feito um aplicativo para aprendizado de vocabulário estrangeiro utilizando reconhecimento de imagem baseado na tecnologia em nuvem, cumprindo as propostas iniciais deste trabalho final de qualificação.

Com estas considerações, identifica-se que este trabalho contribui no campo social ao se desenvolver um aplicativo para suprir necessidades no ensino de idiomas estrangeiros, deixando também contribuição no campo científico ao analisar tecnologias atuais de reconhecimento de imagens em nuvem, além da fundamentação teórica das tecnologias estudadas e trabalhos relacionados.

Como direcionamentos de trabalhos futuros pode-se indicar novos experimentos com outras APIs de reconhecimento de imagens em nuvem disponíveis, podendo ser conduzidos experimentos futuros utilizando outras API's, como por exemplo IBM Watson e Microsoft Azure Computer Vision. Acrescenta-se também a

possibilidade de incrementar a proposta com redes neurais convolucionais ajustadas especificamente para o propósito deste trabalho, utilizando, por exemplo, *frameworks* como Tensorflow Lite e rodando diretamente no aparelho. Desta forma, a capacidade de reconhecimento de imagens poderia ser aumentada, proporcionando melhor experiência ao usuário. Por fim, o aplicativo desenvolvido também pode ter outras funcionalidades que proporcionem maior interatividade com o usuário e complementar o projeto, como por exemplo, funções que permitam ao usuário salvar as palavras que lhe chamarem a atenção, recurso de áudio para ouvir a pronúncia da palavra retornada no idioma estrangeiro e outras melhorias no *layout*.

REFERÊNCIAS

ABDELFATTAH, Abdellatif. Entre a carência e a profusão **Image Classification using Deep Neural Networks — A beginner friendly approach using TensorFlow**. 2017.

ALDA LUCIA E LEFFA, Vilson. Entre a carência e a profusão: Aprendizagem de línguas mediada por telefone celular. **Conexão - Comunicação e Cultura**, v. 13, n. 26, jul 2014.

ALTARAWNEH, Mokhled; AL-QAISI, Aws. Evaluation of cloud computing platform for image processing algorithms. **Journal of Engineering Science and Technology**, 08 2019.

AMAZON. **Amazon Rekognition Documentation**. 2021. Disponível em: <<https://docs.aws.amazon.com/rekognition/index.html>>.

ARKORFUL, Valentina. The role of e-learning, advantages and disadvantages of its adoption in higher education. p. 396, 12 2014.

AXELSSON, Carl-Anton Werner. **Consistency in Web Design from a User Perspective**. Tese (Doutorado), 09 2012.

BERMAN, Pamela. **E-Learning Concepts and Techniques**. Bloomsburg University of Pennsylvania, USA: Institute for Interactive Technologies, 2006. Bloomsburg University of Pennsylvania, USA, 2006. [Online].

BOWLING, Michael et al. Machine learning and games. **Machine Learning**, v. 63, p. 211–215, 06 2006.

CHING, Melvina Chung Hui et al. Testing the usefulness of a mobile edutainment software in learning malay language grammar in primary school. **MALAYSIA: Sci.Int.(Lahore)**, 2018.

CRIOLLO-C, Santiago et al. Towards a new learning experience through a mobile application with augmented reality in engineering education. **Applied Sciences**, maio 2021.

CUMMAUDO, Alex et al. Interpreting cloud computer vision pain-points: a mining study of stack overflow. **International Conference on Software Engineering**, p. 23–29, janeiro 2020.

DU, K.-L.; SWAMY, M.N.s. Perceptrons. Montreal, 2014. p. 67–81.

FIREBASE por plataforma. 2021. Disponível em: <<https://firebase.google.com/docs>>.

FLUMINHAN, Carmem et al. Educação móvel em um mundo híbrido: Mlearning para a aprendizagem de língua estrangeira. São Carlos, 2020.

FOUKI, Mohammed; AKNIN, Noura; KADIRI, kamal eddine el. Multidimensional approach based on deep learning to improve the prediction performance of dnn models. **International Journal of Emerging Technologies in Learning (iJET)**, v. 14, p. 30, 01 2019.

FRANCO, Jean Carlos. **Aplicativo para o reconhecimento de objetos em imagens utilizando a API Cloud Vision destinado a pessoas portadoras de deficiência visual**. Tese (Doutorado) — Universidade Regional de Blumenau, 2018.

FREEPICK. Cachorros. Disponível em:
<https://br.freepik.com/fotos-premium/muitos-filhotes-de-cachorro-em-uma-gaiola_3712338.htm#&position=14&from_view=detail#&position=14&from_view=detail>

GEORGIEV, Tsvetozar; GEORGIEVA, Evgenia; TRAJKOVSKI, Goran. Transitioning from e-learning to m-learning: Present issues and future challenges, 2006. p. 349–353.

GOOGLE. **Referência da API**. 2021. Disponível em: <<https://cloud.google.com/vision/docs/apis>>.

GOOGLE Lens. 2021. Disponível em: <https://play.google.com/store/apps/details?id=com.google.ar.lens&hl=pt_BR&gl=US>.

HARTNETT, Maggie. The importance of motivation in online learning, 2016.

HASHEMI, Masoud et al. What is mobile learning? challenges and capabilities. **Procedia - Social and Behavioral Sciences**, v. 30, p. 2477–2481, 12 2011.

HSU, Chiehwen; LOPEZ, Oscar; RAJ.S, Yeshwant. Considering parental perceptions on mobile learning systems with in-app advertising-taiwan case study. v. 29, p. 2437–2471, 04 2020.

IBACETA, Carlos. Cachorro. Disponível em:
<<https://www.flickr.com/photos/144209554@N07/30389601544>>.

IFTAKHAR, S. Google classroom: What works and how? **Journal of Education and Social Sciences**, v. 3, p. 12–13, feb 2016.

JÚNIOR, Francisco Monteiro de Sales. **Usability in a Virtual Learning Environment: A Study at IFRN**. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, 2016.

KAMBOURAKIS, Georgios; KONTONI, Denise-Penelope; SAPOUNAS, Ioakim. Introducing attribute certificates to secure distributed e-learning or m-learning services. 01 2004.

MASELTOV. **Project**. 2016. Disponível em:
<<http://www.maseltov.eu/Project%20%20C2%AB%20MASELTOV.html>>.

MATHEW, Amitha; ARUL, Amudha; SIVAKUMARI, S. Deep learning techniques: An overview. Coimbatore, 2021. p. 599–608.

MCCULLOCH, Warren; PITTS, Walter. A logical calculus of ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, 1943.

MENON, S. Designing online materials for blended learning: Optimizing on bookwidgets. **International Journal of Linguistics, Literature and Translation (IJLLT)**, v. 2, p. 12–13167–168, may 2019.

MITCHELL, Tom M. **Machine Learning**. McGraw-Hill, 1997.

MUREs, AN, Horea; OLTEAN, Mihai. Fruit recognition from images using deep learning. **Acta Universitatis Sapientiae, Informatica**, v. 10, p. 26–42, 06 2018.

PASINI, Carlos; CARVALHO, Élvio; ALMEIDA, Lucy. A educação híbrida em tempos de pandemia: algumas considerações. **FAPERGS**, p. 396, 06 2020.

PEDRO, Luis; BARBOSA, Cláudia; SANTOS, Carlos. A critical review of mobile learning integration in formal educational contexts. **International Journal of Educational Technology in Higher Education**, v. 15, 12 2018.

PEGRUM, M. **Mobile Learning: Languages, Literacies and Cultures**. Palgrave MacMillan, 2014.

GOOGLE, Primeiros passos no Android com o TalkBack. 2021. Disponível em:
<<https://support.google.com/accessibility/android/answer/6283677?hl=pt-BR>>.

REDDY, Y; PULABAIGARI, Viswanath; B, Eswara. Semi-supervised learning: a brief review. **International Journal of Engineering Technology**, v. 7, p. 81, 02 2018.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65 6, p. 386–408, 1958.

S., K; ABRAHAM, Sajimon. Accuracy evaluation of prediction using supervised learning techniques. In: **ICAICR '19**. 2019.

SAFANELLI, Arcângelo dos Santos; CATAPAN, Araci Hack; ROCHA, Paula Regina Zarelli. E-learning e capital intelectual: uma análise bibliométrica. **XII Coloquio Internacional de Gestión Universitaria**, 2012.

SATHYA, R.; ABRAHAM, Annamma. Comparison of supervised and unsupervised learning algorithms for pattern classification. **International Journal of Advanced Research in Artificial Intelligence**, v. 2, 02 2013.

SHISHAKIN, Vyacheslav. **EngCards - palavras em inglês**. 2021.

SILVA, Inês Santos; GAMA, Sandra; GONÇALVES, Daniel. **CogniHue: Studying the cognitive effect of color in HCI**. Tese (Doutorado), 11 2017.

SOARES, Pablo; SILVA, José da. Aplicação de redes neurais artificiais em conjunto com o método vetorial da propagação de feixes na análise de um acoplador direcional baseado em fibra Ótica. **Revista Brasileira de Computação Aplicada**, v. 3, 12 2011.

SRIVASTAVA, Rahul. **Batch image processing with Amazon Rekognition Custom Labels**. 2021. Disponível em:

<<https://aws.amazon.com/pt/blogs/machine-learning/batch-image-processing-with-amazon-rekognition-custom-labels/>>.

STAROVOYTOV, Maksim. **User interface design and user experience in a corporate intranet website redesign**. Tese (Graduação) — Turku University Of Applied Sciences, Turku, 2018.

STROMER-GALLEY, Jennifer. Interactivity as product and interactivity as process. **Inf. Soc.**, v. 20, p. 391–394, 11 2004.

SZU, Harold; ROGERS, G. Generalized mccullouch-pitts neuron model with threshold dynamics. 1992. p. 535 – 540 vol.3.

TAPTAPSEE. 2021. Disponível em:

<https://play.google.com/store/apps/details?id=com.msearcher.taptapsee.android&hl=pt_BR&gl=US>.

TAPTAPSEE: Assistive Technology for the Blind and Visually Impaired. 2021. Disponível em: <<https://taptapseeapp.com/>>.

TRAXLER, John. Defining mobile learning. **IADIS International Conference on Mobile Learning**, 01 2005.

YANG, Fan; SHEN, Fuyuan. Effects of web interactivity: A meta-analysis. **Communication Research**, v. 45, 03 2017.

ZAYEGH, Amer; BASSAM, Nizar. Neural network principles and applications, 2018.

APÊNDICE A – DETALHES DE IMPLEMENTAÇÃO DO APLICATIVO

A.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

Para realizar o cadastro de usuários foram utilizados três serviços da API da Firebase do Google:

- *Google Authentication*: faz parte do banco de dados em nuvem. Os dados são armazenados como *JSON* e sincronizados em tempo real com cada cliente conectado. Quando são criadas aplicações multiplataforma com os *SDKs* para iOS, Android e JavaScript, todos os usuários compartilham a mesma instância do *Realtime Database* e recebem atualizações automaticamente com os dados mais recentes. Este serviço foi utilizado neste trabalho para a validação do usuário via *email* e senha;
- *Realtime Database*: este serviço também pertence ao banco de dados em nuvem. Os dados são armazenados como *JSON* e sincronizados em tempo real com cada usuário conectado. Neste projeto, este serviço foi utilizado para registrar a foto do usuário;
- *Storage*: é uma solução de armazenamento em nuvem projetada para desenvolvedores de aplicativos que precisam armazenar e fornecer qualquer tipo de conteúdo enviado por usuários, como imagens ou vídeos. Incluindo no plano gratuito até 10 GB de espaço livre na nuvem O programador pode usar o SDK para ler e salvar arquivos de áudio, imagens, arquivos de vídeo ou qualquer outro formato de dados que possa ser necessário para exibir conteúdo. Os arquivos enviados podem ser visualizados no *FirebaseConsole*. Neste aplicativo, foi utilizado para guardar outros dados do usuário, como data de nascimento e gênero (FIREBASE, 2021).

Para consumir os serviços da API, foi optado pelo plano gratuito do serviço de banco de dados Firebase, que por ser gratuito possui algumas restrições, pos-

suindo limitações quanto à capacidade total de armazenamento, fixada em 1 GBA, e possui limitações quanto ao número de conexões simultâneas, disponibilizado a 100 usuários (GOOGLE, 2021).

Para a identificação de objetos em imagens foi utilizada a API da Google intitulada *Google Vision*, pois, conforme experimentos feitos, esta plataforma tem a possibilidade de se adequar melhor ao desenvolvimento do aplicativo. Foi optado pelo plano gratuito para consumir os serviços desta API, este plano permite uma determinada quantidade de detecção de rótulos por mês, caso ultrapasse quantidade máxima há uma cobrança a cada mil novos rótulos detectados (GOOGLE, 2021).

Para a tradução dos rótulos detectados para o idioma russo e também para o português, foi utilizada a API *Google Translate*. O plano gratuito também foi escolhido, e este possui limitações quanto à quantidade de palavras a serem traduzidas.

A.2 DETALHAMENTO DA IMPLEMENTAÇÃO DAS CLASSES

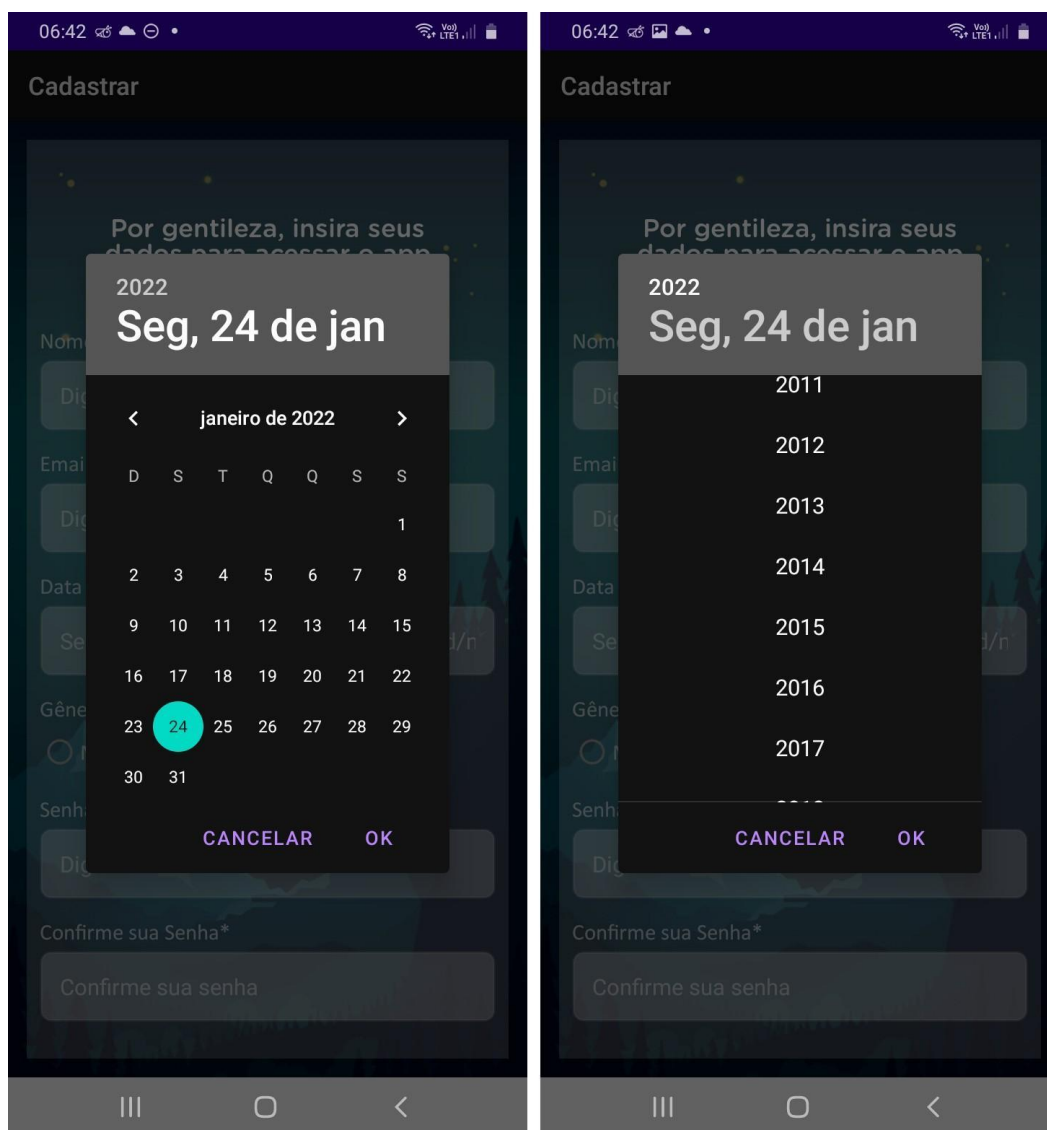
Na classe *RegisterActivity* o método responsável pelo cadastro trata-se do *registerUser*, que passa como parâmetro as informações declaradas pelo usuário nos campos de dados, estes dados tratam-se do “nome completo”, “*email*”, “senha”, “data de nascimento” e “gênero”.

Para preencher campos que exigem entrada de data, foi usado o elemento do modelo Android *DatePicker*, tornando a entrada conveniente e correta. Tais elementos são utilizados em duas *Acitivitys*: *RegisterUser* e *UpdateProfileActivity*. Um exemplo de exibição de *DatePicker* é mostrado na Figura 16.

Cabe destacar que a forma escolhida de implementação para verificar se o usuário está logado ou não está diretamente relacionada ao ciclo de vida das *Activitys* e às funcionalidades do banco de dados. Para verificar se o usuário está logado, o método *onStart()* é utilizado na classe *LoginActivity*, neste método encontra-se uma validação automática para saber se o usuário está logado. A verificação é feita pelo método *getCurrentUser()* disponibilizado pelo Firebase, este método é responsável

por capturar a instância atual do usuário, se o método retornar nulo significa que o usuário não está logado, portanto foi configurado para exibir uma mensagem na tela do usuário informando que ele pode efetuar o *login*. Porém se o método não retornar nulo, significa que o usuário está logado, então a classe *PhotoActivity* é chamada, redirecionando o mesmo para a *Activity* principal do aplicativo.

Figura 16 – Formato DatePicker



Fonte: Autor, 2021

Na classe *LoginActivity*, o método *loginUser* é responsável por identificar a instância atual e fazer as validações de email e senha necessárias. As validações são feitas utilizando métodos e classes providas pelo próprio Firebase. Estas validações

são:

- *FirebaseAuthInvalidUserException*: classe que estende a classe *FirebaseAuthException*, responsável por tratar as exceções que podem ocorrer. Esta exceção é chamada, por exemplo, ao tentar executar uma operação em uma instância *FirebaseUser* que não é mais válida. Geralmente é chamada quando não é possível encontrar o usuário no banco de dados.
- *FirebaseAuthInvalidCredentialsException*: Lançado quando uma ou mais credenciais passadas para um método falham em identificar o usuário que está realizando a operação. Esta exceção é chamada, por exemplo, quando o usuário digita o *email* ou senha de forma errônea.
- *isEmailVerified*: constata se o *email* do usuário é verificado para identificar se o endereço de *email* informado pelo mesmo é verdadeiro. Caso o *email* não tenha sido verificado ainda, foi configurado para encaminhar novamente um *email* para aquele cadastrado pelo usuário para que o mesmo possa confirmar. Também foi configurado para exibir uma mensagem ao usuário pedindo para acessar o *email* cadastrado e, assim, realizar a confirmação.

Na classe *ForgotPasswordActivity*, o procedimento para redefinição de senha é realizado pelo método *resetPassword*, no qual dentro dele é selecionada a instância do perfil do usuário e, a partir desta instância, chama o método *sendPasswordResetEmail*, que passa como parâmetro o *email* do usuário. Quando o usuário clica no *link* enviado por *email*, a senha é alterada.

Na classe *ImageClassificationActivity*, pós o usuário escolher ou capturar a foto para encaminhar para a *API*, a foto é transformada em *Bitmap* e, só então, enviada para análise da *API* que retornará os resultados em inglês. Esses resultados são capturados pela *API* do *Google Translate*, responsável por traduzir estes resultados para o idioma português e russo. Após a tradução, todos os rótulos retornados são passados para uma lista declarada como *ArrayList<Algo>*, estas a

serem passadas ao adaptador, responsável por exibir os dados na tela do usuário caso tenha conseguido identificar algum objeto.

Na classe *UserProfileActivity*, o método *showUserProfile* é responsável por exibir os dados do usuário na tela, dentro dele, o método *getInstance* é responsável por capturar a instância do usuário e, juntamente com a classe *DataSnapshot*, responsável por copiar os dados encontrados no banco de dados, conseguem capturar os dados do usuário contidos no banco de dados e, após isso, foi configurado para exibi-los na tela pelo método *setText()*. Para capturar a foto do usuário, foi utilizado o método *getPhotoUrl* disponível pelo Firebase, e para exibir a foto foi utilizada a biblioteca Picasso.

Na classe *UploadProfilePictureActivity*, para escolher a foto, o método *openFileChooser* é chamado, este método aciona uma nova Intent que, por intermédio do parâmetro *ACTION_GET_CONTENT*, responsável por abrir seletores para o usuário escolher um dado, permite o usuário escolher um certo tipo de dado que contém no dispositivo móvel, no caso, foi configurado para selecionar imagens. Após a escolha da imagem, a foto é atualizada pelo método *UploadPic()*, responsável por mandar uma requisição ao Firebase com a *Uri* da foto.

A classe *UploadProfileActivity* atualiza as informações do perfil do usuário como nome, data de nascimento e gênero do usuário. Esta atualização é feita pelo método *updateProfile* que recebe o usuário como parâmetro e utiliza o elemento de interface *setText* para atualizar as informações.

A classe *UpdateEmailActivity* atualiza o *email* do usuário com o método *updateEmail*. Este método utiliza o elemento *updateEmail* disponibilizado pelo Firebase para realizar a tarefa, passando o novo *email* com parâmetro. Nesta classe existe também o método *reAuthenticate*, encarregado de autenticar a senha do usuário.

Para alterar a senha existe a classe *ChangePasswordActivity*, no qual o método *changePwd* se encarrega de executar a tarefa. Esta tarefa é executada pelo elemento *updatePassword* provindo pelo Firebase, que passa a nova senha como parâmetro para ser atualizado no banco de dados. Nesta classe, existe o método

reAuthenticateUser, responsável por autenticar novamente o usuário antes do mesmo realizar a ação de trocar a senha.

Na classe *DeleteProfileActivity*, o método *deleteUser* é responsável pela remoção do usuário, no qual é passada a instância atual do usuário, passando o elemento *delete()* disponibilizado pelo Firebase. Nesta classe existe também o método *reAuthenticateUser* para validar novamente as credenciais do usuário.

Para fazer o *logout* da conta, a instância do usuário é capturada e o método *signOut* disponibilizado pelo Firebase é chamado para executar o procedimento.

A.2.1 Detalhamento das validações

Esta seção visa fornecer informações a respeito das validações implementadas a fim de prevenir erros e exceções.

Na tela de cadastro são feitas as seguintes validações:

- São validados se todos os campos foram preenchidos, caso haja algum campo não preenchido, o usuário é solicitado para que preencha o campo em questão.
- No campo “*email*”, se o usuário fornecer um email já existente no banco de dados, uma mensagem é exibida alertando que já existe um usuário com o endereço de *email* informado. Se o usuário informar um email inválido, como por exemplo, sem o “@”, uma mensagem é exibida informando sobre.
- O campo “senha” deve possuir quantidade maior do que 4 caracteres, caso tenha um número igual ou menor, o usuário é informado que a senha é fraca, sendo solicitado a realizar a correção.
- No campo “confirmar senha”, a senha deve ser idêntica ao que foi digitado no campo “senha”, caso esta correspondência de dados não seja detectada, o usuário é informado do ocorrido.

Importante mencionar que o usuário é impedido de prosseguir com o

cadastro até que todos os critérios mencionados anteriormente sejam atendidos.

Na tela de *login* são validadas as seguintes condições:

- No campo “*email*”, se o usuário digitar um *email* inválido, como por exemplo, sem o “@”, é informado que o usuário é inválido.
- Caso seja digitado um *email* válido que não contém no banco de dados, o usuário é instruído a verificar novamente o *email* ou efetuar o cadastro com o *email* mencionado.
- Caso o usuário digite um *email* válido que contém no banco de dados, porém que não corresponde com a senha cadastrada, é direcionado um aviso ao usuário informando que as credenciais informadas não correspondem, solicitando para que o mesmo verifique novamente os dados informados.

A permissão de *login* é concedida quando os critérios acima são devidamente cumpridos.

Algumas telas contém operações sensíveis para o usuário, como por exemplo, a operação de atualizar *email*, senha e deletar perfil. Para estas três operações, foi implementado um procedimento de autenticação como forma de garantir segurança extra. O usuário poderá dar início à operação somente quando autenticar-se conforme instruído na tela, caso a autenticação não seja cumprida com sucesso, o usuário é informado.

Na autenticação referente à tela de atualização do *email* é solicitado que o usuário digite o *email* atual e senha para prosseguir com a operação. Na tela de atualizar senha, é exigido que o usuário digite a senha atual para prosseguir. Por fim, na tela referente à remoção do perfil, a senha é solicitada antes de dar início à operação.

Para efetuar a remoção do perfil, além da exigência de autenticação do usuário, uma mensagem é exibida ao pressionar o botão de deletar, a mensagem informa que o perfil e dados relacionados ao usuário serão deletados, a remoção do perfil só é iniciada ao pressionar a opção “continuar”.

Na primeira tela após o *login*, se o usuário optar por capturar uma foto para enviar para a API e a foto não for capturada, uma mensagem é exibida informando que a foto não foi tirada. Caso o usuário opte por selecionar uma foto da galeria do celular, porém não escolher uma imagem, é informado que a imagem não foi selecionada.

Relacionado ao retorno de dados da API, existem casos em que não é possível identificar o que contém na imagem, retornando, desta forma, uma interface vazia. Para este caso, foi adicionada uma validação no qual é exibida uma mensagem esclarecendo ao usuário que a imagem enviada não pôde ser classificada.

Por fim, pode ocorrer o caso em que a API de tradução de idiomas receba muitas requisições e fique sobrecarregada, dessa forma, impedindo que os dados sejam processados e mostrados na tela. Para este cenário, uma mensagem é mostrada ao usuário informando que a API está sobrecarregada, portanto deve tentar o procedimento novamente mais tarde.