UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕES - CAMPUS DE ERECHIM
DEPARTAMENTO DE ENGENHARIAS E CIÊNCIA DA COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

VINICIUS EMANOEL ANDRADE

EMOGNIZER: APLICAÇÃO BASEADA EM INTELIGÊNCIA ARTIFICIAL PARA ANÁLISE EMOCIONAL DE REDES SOCIAIS

ERECHIM - RS
2020

VINICIUS EMANOEL ANDRADE

EMOGNIZER: APLICAÇÃO BASEADA EM INTELIGÊNCIA ARTIFICIAL PARA ANÁLISE EMOCIONAL DE REDES SOCIAIS

Projeto de Conclusão de Curso elaborado e apresentado na disciplina de Projeto de Conclusão de Curso, Curso de Ciência da Computação, Departamento de Engenharias e Ciência da Computação da Universidade Regional Integrada do Alto Uruguai e das Missões - Campus de Erechim.

Orientador: Prof. Dr. Guilherme Afonso Madalozzo

ERECHIM - RS
2020

AGRADECIMENTOS

Agradeço à Talissa, minha namorada, que sempre esteve ao meu lado durante toda minha jornada acadêmica, sempre me auxiliando, apoiando e dando forças para que eu pudesse seguir em frente.

Agradeço aos meus pais, Emanoel e Adriana, os quais nunca mediram esforços para que eu e minha irmã, Julia, tivéssemos uma excelente educação. Sou eternamente grato à Deus pela família maravilhosa que eu tenho.

Agradeço ao meu orientador, Prof. Guilherme Madalozzo, que me auxiliou para que este trabalho pudesse ser realizado. Estendo à todos os professores do curso de Ciência da Computação da URI, em especial ao Prof. Gerson Groth, responsável por sugestões de grande relevância.

Por fim, agradeço à todos os meus colegas da turma 2016, pessoas incríveis que tornaram esta jornada muito mais divertida. Em especial, agradeço ao João Vitor e ao Kelwin, os quais foram meus parceiros nos trabalhos da faculdade e, sem dúvida, serão amigos que carregarei para toda vida.

RESUMO

Tendo em vista que grande parte da população brasileira encontra-se ativa nas redes sociais e as emoções propagadas pelas pessoas são reflexo de sua saúde mental, foi desenvolvido um sistema que detecta emoções em textos produzidos por um indivíduo em seus perfis do Twitter e do Reddit. Tais emoções são abordadas através de representações que proporcionam encontrar indícios de transtornos mentais. As representações estão presentes em um aplicativo destinado aos dispositivos móveis, construído utilizando um *template* que prega pela imersão e boa experiência do usuário. Para tanto, realizou-se um levantamento de transtornos mentais que possuíssem características passíveis de serem representadas computacionalmente, o qual resultou em quatro transtornos: Transtorno Disruptivo da Desregulação do Humor (TDDH), Transtorno de Ansiedade Generalizada (TAG), Fobia Específica e Transtorno Depressivo Maior (TDM). Além de que, usou-se cinco *datasets* compostos por registros rotulados por especialistas, os quais foram pré-processados através de técnicas de Processamento de Linguagem Natural (PLN) e, posteriormente utilizados para realizar o treinamento e avaliação de um modelo de Rede Neural *LSTM* (*Long Short-Term Memory*) que, durante o processo de avaliação, mostrou ser capaz de reconhecer emoções em textos retirados de ambas as redes sociais citadas anteriormente.

Palavras-chave: Processamento de Linguagem Natural. Redes Neurais. Emoções. Transtornos Mentais.

ABSTRACT

Bearing in mind that a large part of the Brazilian population is active on social networks and the emotions propagated by people are a reflection of their mental health, a system was developed that detects emotions in texts produced by an individual in their Twitter and Reddit profiles. Then, it presents them through representations that provide finding signs of mental disorders. The representations are present in an application for mobile devices, built using a template that preaches for immersion and good user experience. Therefore, a survey of mental disorders that had characteristics that could be represented computationally was carried out, which resulted in four disorders: Disruptive Mood Dysregulation Disorder (DMDD), Generalized Anxiety Disorder (GAD), Specific Phobia, and Major Depressive Disorder (MDD). Also, five datasets composed of records labeled by specialists were used, which were pre-processed using Natural Language Processing (NLP) techniques and later used to carry out the training and evaluation of an LSTM (Long Short-Term Memory) Neural Network model who, during the evaluation process, showed to be able to recognize emotions in texts taken from both social networks mentioned above.

Keywords: Natural Language Processing. Neural Networks. Emotions. Mental Disorders.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
FIFO	*First-in First-out*
HTTP	*Hypertext Transfer Protocol*
HTTPS	*Hyper Text Transfer Protocol Secure*
IA	Inteligência Artificial
JSON	*JavaScript Object Notation*
LSTM	*Long Short-Term Memory*
NLTK	*Natural Language Toolkit*
PLN	Processamento de Linguagem Natural
RNCs	Redes Neurais Convolucionais
RNNs	Redes Neurais Recorrentes
TAG	Transtorno de Ansiedade Generalizada
TDDH	Transtorno Disruptivo da Desregulação do Humor
TDM	Transtorno Depressivo Maior
UML	*Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO

A escolha de áreas promissoras, sem dúvida foi um fator que moveu o desenvolvimento deste trabalho, porém ele não foi o único. Soma-se a ele outro importante fator: a perspectiva de criação de um produto promissor que pudesse impactar positivamente na vida das pessoas.

Segundo o estudo feito por Social e Hootsuite (2018), 130 milhões de brasileiros são usuários ativos nas redes sociais, isso representa 62% da população do país. O total global é de 3.196 bilhões, representando 42% da população mundial.

Como assegurado por MindMiners e FAAP (2019), em uma pesquisa realizada com 1000 pessoas, as quais fazem uso das redes sociais, 22% delas já sentiram tristeza e 17% já sentiram raiva enquanto utilizavam-nas. Além disso, os Autores constataram que, "31% dos respondentes já foi diagnosticado com ansiedade e 18% com depressão.".

Tanto a ansiedade, quanto a depressão fazem parte dos transtornos mentais. Contudo, vale ressaltar que, os transtornos de ansiedade e os transtornos depressivos são dois grupos compostos por múltiplos transtornos. O grupo dos transtornos de ansiedade é composto por 12 diferentes transtornos. Esse grupo caracteriza-se pela ansiedade e pelo medo, os quais são propagadas excessivamente. O grupo dos transtornos depressivos é composto por 8 diferentes transtornos. Esse grupo caracteriza-se pela tristeza, irritabilidade e pelas mudanças comportamentais de um indivíduo, as quais prejudicam-no pessoal e profissionalmente (AMERICAN PSYCHIATRIC ASSOCIATION, 2014).

Como acima descrito e também assegurado em HOSPITAL SANTA MÔNICA (2018), alguns transtornos mentais correlacionam-se com algumas emoções. Em AMERICAN PSYCHIATRIC ASSOCIATION (2014), é possível visualizar que, para diagnosticar alguns transtornos é necessária a observação do período de tempo em que uma determinada emoção foi propagada. Contudo, vale ressaltar que os diagnósticos são mais complexos do que isso, sempre compostos por uma série de características, as quais diversificam-se entre os muitos transtornos existentes.

Com base nas estatísticas apresentadas por MindMiners e FAAP (2019), evidencia-se que existem indivíduos com transtornos mentais que fazem uso das redes sociais. Inclusive, observa-se um percentual alto de diagnósticos dentre os 1000 participantes da pesquisa. A partir disso, colocando esses percentuais em uma escala maior como, por exemplo, no número de usuários ativos nas redes sociais apresentado em Social e Hootsuite (2018), imagina-se a grande quantidade de possíveis portadores de transtornos mentais existentes nas redes sociais. Entretanto, é improvável que todas essas pessoas obtiveram um diagnóstico.

É preciso, porém, ir mais além, pois a pandemia de *covid-19* trouxe impactos na saúde mental das pessoas. Por meio da pesquisa realizada pela YoungMinds (2020), com um total de 2.036 jovens - que já enfrentam problemas relacionados a saúde mental -, 81% deles relataram que a pandemia piorou sua situação.

Diante destas informações, mostrou-se oportuno o desenvolvimento de um sistema que possibilite encontrar indícios de transtornos mentais através do que é escrito por um indivíduo em suas redes sociais.

Considerando as informações obtidas, as quais evidenciam que emoções correlacionam-se com transtornos mentais, restou saber, então: é possível prever - de forma automatizada - quais são as emoções presentes nos textos criados por uma pessoa nas redes sociais? Além disso, é concebível transformar essas previsões em representações que proporcionem encontrar indícios de transtornos mentais?

Portanto, o objetivo geral deste trabalho foi desenvolver um sistema que proporciona localizar uma pessoa em duas redes sociais: Twitter e Reddit. Após localizada, busca-se uma quantidade específica de textos escritos por ela. Em seguida, prevê-se quais são as emoções subjacentes à esses textos. Por fim, transforma-se essas previsões em representações intuitivas e que possibilitam que indícios de transtornos mentais sejam encontrados.

Para atingir o objetivo geral, foi preciso atender alguns objetivos específicos: (1) Foi desenvolvido um modelo de rede neural que prevê emoções presentes em textos. (2) Foi desenvolvida uma aplicação que comunica-se com as redes sociais Twitter e Reddit para localizar os perfis de uma pessoa, obter os textos produzidos por ela, pré-processá-los e utilizá-los como entrada na rede neural. (3) Foi desenvolvido um aplicativo que consome serviços, obtém os resultados e prepara-os para o usuário final.

Por fim, resultou-se em um sistema denominado *Emognizer*. Cujo, proporciona que seus usuários realizem uma análise emocional das redes sociais de uma pessoa (familiar, amigo, conhecido, paciente, etc...) e, com base em seus conhecimentos, tirem suas próprias conclusões sobre a saúde mental dela. Ou seja, o *Emognizer* não realiza diagnósticos e nem sinaliza que indícios de transtornos mentais foram encontrados, somente fornece os subsídios necessários para que o próprio usuário encontre-os.

Em um primeiro momento, pode parecer que o *Emognizer* é exclusivo para profissionais da área da saúde mental como, por exemplo, os psicólogos e os psiquiatras. Contudo, ele foi planejado para que pessoas sem conhecimento científico sobre transtornos mentais, também possam tirar proveito. Uma melhor explicação sobre isso é encontrada no decorrer deste trabalho.

O trabalho foi dividido em sete Capítulos e, na sequência deste parágrafo está uma breve explicação do conteúdo presente em cada um deles. O Capítulo 2 apresenta toda a base teórica utilizada para o desenvolvimento deste trabalho. Nele, explica-se o que são as emoções, quais são as emoções, como elas correlacionam-se com transtornos mentais e como é possível encontrá-las em textos de forma manual. Em seguida, expõe-se quais são as técnicas necessárias para encontrar emoções em textos de forma automatizada, entre elas estão: Processamento de Linguagem Natural (PLN), Redes Neurais e Aprendizagem Supervisionada. O Capítulo 3 contém a metodologia utilizada para resolver os problemas que estão no cerne deste trabalho. Ou seja, como prever emoções em textos e como transformar as previsões em representações

que possibilitam encontrar indícios de transtornos mentais. O Capítulo 4 destaca todos os detalhes do sistema. Primeiro, apresenta-se sua proposta de forma detalhada. Em seguida, através de diagramas, todas as funcionalidades são expostas. O Capítulo 5 explica como as partes do sistema foram desenvolvidas. Ele possui Subseções que distribuem-se da seguinte forma: rede neural, *backend*, prototipação do aplicativo, programação do aplicativo e bancos de dados. O Capítulo 6 exibe os resultados do sistema. Nele, mostra-se o desempenho do sistema prevendo as emoções em textos curtos e em textos que derivaram das redes sociais. O Capítulo 7 contém as considerações finais. Nele, está um detalhamento do que foi feito ao longo do trabalho e sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Antecedendo qualquer etapa prática do trabalho, é necessário o domínio dos principais tópicos que serão abordados no mesmo. No decorrer desta Seção, apresenta-se o conhecimento adquirido durante as pesquisas. A primeira Subseção apresenta o que são e quais são as emoções. Em seguida, é feita uma correlação entre emoções e transtornos mentais. Na segunda Subseção, são exibidos os métodos adotados por outros trabalhos para reconhecer manualmente emoções em textos. A terceira Subseção é sobre redes sociais, dando ênfase nas duas que foram utilizadas. A quarta Subseção aborda as técnicas utilizadas no trabalho, iniciando pelo PLN, seguido pelo Aprendizado Profundo e finalizando com a Aprendizagem Supervisionada.

2.1 Emoções

Afinal, o que são as emoções? Distante do que muitos pensam, emoções são um conjunto de ações executadas pelo corpo humano. Algumas dessas ações são visíveis, outras completamente imperceptíveis. As ações são desencadeadas de forma automática, ou seja, os seres humanos não possuem nenhum controle sobre os efeitos gerados por uma emoção (DAMASIO, 2011).

O ciclo emocional pode ser desencadeado por alguns fatores, são eles: algum acontecimento no exato momento, acontecimentos relembrados do passado, imagens de objetos vistos ou até mesmo imagens criadas mentalmente. São essas imagens que, após processadas, desencadeiam uma série de complexos eventos. Vale ressaltar que o ciclo emocional sempre inicia em uma região específica do cérebro, em seguida alastra-se para as demais regiões - envolvidas no processo emocional e sentimental - e para o resto do corpo. O final do ciclo também envolve o cérebro, mas não necessariamente a mesma região que deu inicio à ele (DAMASIO, 2011).

Segundo Bear, Connors e Paradiso (2017, p. 622), a primeira região cerebral atrelada às emoções, foi o lobo límbico de Broca. Ela foi descoberta pelo neurologista Paul Broca e é descrita pelos autores da seguinte forma: “um grupo de áreas corticais que são bastante distintas do córtex circundante”. Broca não mencionou em seu artigo - escrito em 1878 - sobre a relação dessa região com as emoções, inclusive, inicialmente pensava-se que ela estaria ligada ao olfato.

Ainda segundo Bear, Connors e Paradiso (2017, p. 622), “o neurologista norte-americano James Papez propôs que houvesse, na parede medial do encéfalo, um ‘sistema da emoção’, que ligaria o córtex ao hipotálamo”. Através de estudos veio a comprovação de que, de fato, os componentes do sistema proposto por Papez estariam interconectados. Apesar disso, nada comprovava que eles estavam ligados às emoções de uma pessoa. Tudo o que ele possuía eram evidências, em sua grande maioria relacionadas ao comportamento emocional de pessoas que tinham lesões naquela região. Era perceptível que as lesões mudavam a experiência

emocional do indivíduo, porém, sua inteligência e percepção não eram afetadas. Inclusive, vale mencionar que, segundo o próprio Papez, algumas dessas lesões poderiam causar perturbações emocionais, como por exemplo, medo, irritabilidade e depressão.

Como mencionado anteriormente, o lobo límbico não havia sido relacionado com as emoções, porém, existiam muitas semelhanças entre os dois sistemas, devido à isso, o sistema de Papez - também conhecido como Circuito de Papez - é frequentemente chamado de sistema límbico, a popularização desse termo ocorreu através de Paul MacLean, em 1952. Ao longo da história, novas evidências foram surgindo e alguns dos componentes desses sistemas acabaram se desaliando com as emoções. Apesar de todas as hipóteses apresentadas, é cada vez mais evidente que são inúmeros sistemas que relacionam-se diretamente com as emoções de uma pessoa, não existe apenas um, como proposto no passado (BEAR; CONNORS; PARADISO, 2017).

Através das ações do corpo durante o ciclo emocional, percebe-se que a emoção funciona como um mecanismo de defesa, possibilitando que o indivíduo sobressaia-se em uma situação de perigo. Para uma melhor compreensão de como a emoção propaga-se no corpo, vamos analisar o medo. Como bem explicado por Calabrez (2016), dentre as ações perceptíveis está a clássica máscara do medo, causada pela redução de sangue do rosto e pelos movimentos dos músculos faciais. Não é só do rosto que o sangue é removido, todos os músculos julgados não importantes naquele momento, acabam tendo essa mesma remoção. E para onde esse sangue é enviado? Para as pernas e os braços, por exemplo, afinal, se uma evasão for necessária, esses serão os grandes protagonistas.

Algumas das ações realizadas internamente no corpo são: a liberação de cortisol na corrente sanguínea, o aumento dos batimentos cardíacos e até mesmo a redução do processamento de dores, pois em caso de fuga, a probabilidade de uma parada por conta de uma dor deve ser mínima (DAMASIO, 2011).

Existem seis emoções consideradas universais, ou seja, todas as pessoas são capazes de vivência-las, independente de quais foram suas experiências culturais. Elas são denominadas: alegria, tristeza, nojo, medo, surpresa e raiva. O responsável por essa descoberta foi Paul Ekman (CHERRY, 2020). Ele dedicou sua vida aos estudos das expressões faciais e das emoções, por consequência tornou-se uma das maiores referências no assunto. (PAULEKMANGROUP, 2020) Abaixo, uma breve explicação sobre todas as seis emoções através da perspectiva de Ekman (2011).

- **Raiva**: É caracterizada por manifestar-se quando uma pessoa é hostilizada, seja física ou psicologicamente. Também, ocorre em caso de interferências indesejadas de terceiros durante um determinado ato, impedindo que o mesmo ocorra. A raiva só é capaz de gerar mais raiva, podendo chegar ao ponto de ferir o próximo, por conta disso ela é a mais perigosa das emoções.
- **Surpresa**: É uma emoção de transição, porque dura poucos segundos e geralmente antecede outra. Ela não é uma emoção neutra, visto que pode ter polaridade positiva ou

negativa. Sua polaridade não está ligada somente à emoção sucessora, existem indivíduos que detestam surpresas, inclusive positivas.

- **Medo**: Não existem limites para o medo, é possível senti-lo com qualquer coisa. Entretanto, algumas formas ou ações mostram-se mais propícias em disparar gatilhos que fazem com que essa emoção manifeste-se. Existem algumas situações que permitem uma tomada de atitude, geralmente ela está atrelada a vivência do indivíduo, ou seja, o que ele fez em situações anteriores. A reação padrão do corpo humano para situações de medo é a fuga - inclusive, ela torna-se propícia pelos motivos citados nos parágrafos anteriores.
- **Alegria**: É indiscutivelmente uma emoção positiva, podendo ter algumas derivações de acordo com sua intensidade. Caracteriza-se por arrancar sorrisos das pessoas e, em alguns casos, lágrimas.
- **Tristeza**: Comumente atrelada à perda. Caracteriza-se por ser uma emoção de longa duração, funcionando em ciclos que intercalam entre o desamparo e a angústia (sensação de que o objeto ou o indivíduo pode ser recuperado). Sua ligação com a perda faz com que algumas pessoas considerem-na um sentimento sem propósito. Entretanto, ela possui um papel muito importante durante o processo de aceitação. As alterações que ela causa no corpo - por exemplo, no rosto e na fala - atraem atenção das pessoas que, por sua vez, tendem a consolar o indivíduo em questão.
- **Nojo**: Também conhecida como aversão, ao contrário do que muitos pensam, sua relação com os sentidos vai muito além do paladar, do olfato e do tato. É possível sentir nojo somente ao olhar algo - como por exemplo, um ferimento exposto ou um inseto. Além disso, o nojo pode manifestar-se em um diálogo, derivando das ideias propagadas por um terceiro.

2.1.1 Transtornos Mentais

O termo “depressão” é frequentemente utilizado de uma forma equivocada, onde um indivíduo utiliza-o com o intuito de representar a tristeza - ou alguma derivação dela. Entretanto, a depressão não é um termo que representa a tristeza, mesmo que ela seja abundante (EKMAN, 2011).

A tristeza abundante pode ser contornada após a realização de uma atividade prazerosa, pois a atividade em questão servirá de gatilho para que um novo ciclo emocional manifeste-se. Um transtorno mental, como por exemplo algum dos transtornos depressivos, faz com que uma única emoção propague-se de forma contínua - nesse caso, a tristeza -, impedindo que novas emoções manifestem-se (EKMAN, 2011). Um indivíduo com um transtorno depressivo dificilmente teria motivação para realizar uma nova atividade, visto que uma das principais características dos transtornos mentais é interferir de forma prejudicial na rotina das pessoas, sejam em tarefas profissionais ou pessoais (AMERICAN PSYCHIATRIC ASSOCIATION, 2014). É importante ressaltar que essa interferência pode ser duradoura, visto que os transtornos mentais não possuem um tempo específico de duração, logo podem acompanhar um indivíduo

por um longo período de vida (FIRST, 2017).

Além da tristeza, outro exemplo de emoção presente em transtornos mentais é o medo, comumente propagado por indivíduos que possuem alguma fobia. Ainda sobre as fobias, elas também caracterizam-se pelo excesso de nojo - ou aversão -, principalmente quando estão atreladas a sangue ou animais. Todas as emoções negativas tendem a aparecer em algum dos transtornos, o importante é a compreensão de que um transtorno mental é a propagação de forma desproporcional de uma emoção, ao ponto de causar desconforto e afetar o cotidiano (EKMAN, 2011).

2.2 Emoções em textos

Wu, Chuang e Lin (2006) evidenciam a relevância dos textos, pois sua produção é abundante e eles são muito versáteis. A versatilidade vem do conteúdo subjetivo à eles, pois além de opiniões e ideias, textos contém emoções.

Antecedendo o reconhecimento automatizado de emoções em textos, é necessário compreender quais são as formas existentes de encontrá-las manualmente. Para isso, foram investigados métodos presentes na literatura. Eles serão descritos no decorrer desta Seção.

Hancock, Landrigan e Silver (2007) propuseram uma pesquisa para averiguar a capacidade das pessoas em externar suas emoções e reconhecer as emoções propagadas por outro indivíduo, durante uma conversa. Além disso, eles conduziram os participantes para que manifestassem quais eram suas formas de demonstrar emoções positivas e negativas.

Aman e Szpakowicz (2007) utilizaram o conhecimento de especialistas para rotular as emoções presentes em frases. O conteúdo textual do estudo derivou de blogs. Os possíveis rótulos eram as seis emoções básicas de Ekman, emoções mistas e sem emoção.

Outro exemplo que utiliza anotações de especialistas, pode ser encontrado em Strapparava e Mihalcea (2007). Esse trabalho utilizou seis especialistas para identificar as emoções presentes em machetes de notícias. Segundo os autores, manchetes são feitas para desencadear - de forma proposital - emoções nos leitores.

Scherer e Wallbott (1994) apresentam uma abordagem semelhante, mas com métodos distintos. Os autores utilizaram questionários para capturar auto-relatos de experiências emocionais dos participantes. Segundo eles, a utilização de questionários anônimos tende a extrair relatos sinceros - e íntimos.

2.3 Redes Sociais

Segundo Labadessa (2012), as redes sociais tornaram-se ferramentas indispensáveis na sociedade em que vivemos. O principal motivo é o poder de comunicação que elas tem, proporcionando que as pessoas interajam, aprendam e relacionem-se.

Elas são grandes montantes de informações pessoais. Porém, muitos usuários não fazem a menor ideia para onde seus dados estão indo e como serão utilizados (SANTANA

et al., 2009). São as corporações que gerenciam as redes sociais que decidem como utilizar os dados, claro que sempre respeitando a legalidade. Por mais que sejam burocráticos e extensos, são os termos de uso que transparecem ao usuário tudo que ele precisa saber, inclusive, quais foram seus dados coletados e como serão empregados (TSUTSUMI, 2014).

A utilização das informações presentes nas redes sociais não serve somente para fins comerciais, também podem ser usadas para questões de saúde pública. Já existem inúmeros estudos e pesquisas que coletaram esses dados em prol do bem-estar coletivo (ARAUJO et al., 2012). Inclusive, esse foi o objetivo deste trabalho, explorar todos os recursos proporcionados por um grupo seleto de redes sociais e transformá-los em informações úteis na mão das pessoas. Para isso, é necessária uma investigação sobre as redes sociais selecionadas.

2.3.1 Twitter

O Twitter é uma das redes sociais mais utilizadas no mundo, com aproximadamente 326 milhões de usuários ativos. Uma boa parcela desses usuários está no Brasil, onde ocorreu sua popularização em 2009. Ele possui algumas características marcantes, como, por exemplo, as sinceras opiniões dos seus usuários, as inúmeras discussões sobre os temas relevantes da atualidade e as notícias em primeira mão, sejam elas derivando de veículos da mídia, empresas ou governantes (IMME, 2020).

Ele possui um programa de desenvolvedores, conhecido como *Twitter Developer*. Esse programa proporciona acesso à uma série de recursos, como por exemplo buscar todos os *tweets* de um determinado usuário. Essa busca tem um limite de 100.000 requisições por dia e retorna no máximo 3.200 *tweets* (TWITTER, 2020).

2.3.2 Reddit

Como nos mostra a própria página do Reddit (2020a), ele é caracterizado pela grande quantidade de comunidades existentes, notícias atualizadas, conhecimento sobre os mais variados assuntos e longos debates entre os seus usuários. Apesar de ser uma plataforma popular nos dias atuais, ele foi fundado no começa da década passada, mais precisamente em 2005. Segundo dados da Imme (2020), ele possui aproximadamente 330 milhões de usuários ativos.

O Reddit proporciona aos desenvolvedores uma série de funcionalidades. No contexto deste trabalho, a principal delas é a possibilidade de visualizar todos os comentários feitos por um usuário (REDDIT, 2020b).

2.4 Inteligência Artificial

Uma interpretação possível é que a IA (Inteligência Artificial) é o campo responsável por proporcionar que máquinas realizem tarefas que exijam intelecto. Por consequência, retirando algumas incumbências dos seres humanos (CHOLLET, 2017).

O termo IA nasceu de um encontro em 1956, idealizado por John McCarthy na Universidade de *Dartmouth*. Onde, importantes pesquisadores - interessados na evolução da área - foram convidados. É possível pensar que, o principal feito do encontro foi consolidar a IA como uma disciplina independente. No cerne dessa questão está: a divergência entre os objetivos da IA e de suas disciplinas subjacentes (NEAPOLITAN; JIANG, 2018).

Segundo Russel e Norvig (2013, p. 18), o encontro durou dois meses e, em oposição às expectativas dos seus dez participantes, nenhuma ideia inovadora foi consolidada. Outro interessante fator citado pelos autores é: "O seminário de *Dartmouth* [...] apresentou uns aos outros todos os personagens importantes da história. Nos 20 anos seguintes, o campo seria dominado por essas pessoas".

Com o passar dos anos a IA foi evoluindo e cada uma de suas "eras" foi marcada pelas abordagens utilizadas. A primeira era foi consolidada pela utilização de sistemas especialistas. A segunda era apresentou uma enorme evolução, pois os sistemas começaram a aprender com a exposição aos dados. A terceira era destacou-se também pelo aprendizado, no entanto com sistemas muito mais robustos e capazes de aceitar gigantescas quantidades de dados (DENG; LIU, 2018).

2.4.1 Processamento de Linguagem Natural

As máquinas não interpretam naturalmente as linguagens utilizadas pelos seres humanos. O PLN é um campo que nasceu com o objetivo de fazer com que as máquinas compreendam essas linguagens e, através do seu conteúdo, executem ações. (DENG; LIU, 2018)

O PLN também pode ser caracterizado como um conjunto de algoritmos que proporcionam às máquinas interpretar linguagens humanas. Esses algoritmos podem atuar na entrada e na saída de dados. Seu papel é transformar linguagens naturais em linguagens de máquina, ou transformar os dados processados pela máquina em linguagens compreensíveis por humanos. (GOLDBERG, 2017)

Como bem assegura Russel e Norvig (2013), o PLN teve sua primeira aparição na década de 1950. Ele era uma das quatro capacidades necessárias para que uma máquina pudesse passar no teste criado por Alan Turing, conhecido como teste de Turing. Segundo Deng e Liu (2018, p. 2, tradução nossa), "Este teste é baseado em conversas em linguagem natural entre um humano e um computador projetado para gerar respostas semelhantes a humanos.".

É preciso, porém, ir mais além para que seja possível compreender o surgimento desse campo. Segundo Russel e Norvig (2013), em meados da década de 1950, também surgia um outro campo subjacente ao PLN, a linguística moderna. Essa época foi marcada pela obra *Syntactic Structures*, de Noam Chomsky. Com isso, ainda segundo Russel e Norvig (2013, p. 16) , "a linguística moderna e a IA 'nasceram' aproximadamente na mesma época e cresceram juntas, cruzando-se em um campo híbrido chamado linguística computacional ou processamento de linguagem natural.".

Tão importante quanto a compreensão dos objetivos do PLN, é compreender de fato o que são as linguagens naturais. Com isso, uma linguagem natural é o artifício utilizado pelas pessoas para comunicarem-se umas com as outras, seja no seu dia a dia ou na internet - em sites e redes sociais. (LANE; HAPKE; HOWARD, 2019).

Diante desse esclarecimento, ainda segundo Lane, Hapke e Howard (2019), podemos diferenciar uma linguagem natural de uma linguagem de programação. Ao contrário de uma linguagem natural, uma linguagem de programação tem a capacidade de dizer à máquina quais tarefas ela deve fazer. A melhor maneira de compreender essa diferença é considerar que, desde os primórdios da computação as máquinas possuem interpretadores e compiladores para linguagens de programação.

Com base nesta comparação, torna-se possível encontrar o momento da disrupção, ou seja, quando uma máquina deixa de processar apenas dados para começar a processar linguagem natural. Um excelente exemplo é o programa *wc*, mencionado na obra de Jurafsky e Martin (2008, p. 2, tradução nossa), "Quando usado para contar *bytes* e linhas, o *wc* é um aplicativo comum de processamento de dados. [...] quando é usado para contar as palavras em um arquivo, requer conhecimento sobre o que significa ser uma palavra".

2.4.2 Aprendizado Profundo

O Aprendizado Profundo é uma vertente do campo de Aprendizado de Máquina. Ele enfatiza a utilização de múltiplas camadas de representações. Sucessivamente, cada uma das camadas obtém representações diferentes que, por sua vez, tornam-se cada vez mais relevantes para o modelo (CHOLLET, 2017). Além disso, ele faz uso das Redes Neurais. Logo, pode-se dizer que a forma com que ele realiza o processamentos dos dados, é uma representação do cérebro de um ser humano (DATA SCIENCE ACADEMY, 2019).

Para que possamos dar sequência no entendimento do Aprendizado Profundo, vamos compreender o que são Redes Neurais e qual foi o seu papel ao longo da história. Segundo Neapolitan e Jiang (2018, p. 7, tradução nossa), "Uma rede neural artificial consiste em uma grande coleção de unidades neurais (neurônios artificiais), cujo comportamento é basicamente baseado na maneira como os neurônios reais se comunicam no cérebro.".

É notável a popularidade que as Redes Neurais possuem nos dias de hoje, não só na comunidade científica, mas também em veículos de mídia popular. Essa condição veio por consequência dos grandes feitos que ela obteve ao longo dos últimos anos. Esses fatores resultaram em novos entusiastas e pesquisas que ajudaram a enriquecê-la (DATA SCIENCE ACADEMY, 2019).

Apesar do entusiasmo nos dias atuais, nem sempre foi assim que os especialistas olharam para as Redes Neurais. Elas tiveram sua primeira aparição na época em que a IA estava dando seus primeiros passos. Inclusive, isso é muito bem detalhado na obra de Russel e Norvig (2013, p. 16):

> O primeiro trabalho agora reconhecido como IA foi realizado por Warren McCulloch e Walter Pitts (1943). Eles se basearam em três fontes: o conhecimento da fisiologia básica e da função dos neurônios no cérebro; uma análise formal da lógica proposicional criada por Russell e Whitehead; e a teoria da computação de Turing. Esses dois pesquisadores propuseram um modelo de neurônios artificiais

A década adjacente à sua criação, foi marcada pelo inicio da IA simbólica. Essa abordagem consolidou os famosos sistemas especialistas, que tiveram seu ápice na década de 1980. Os sistemas especialistas, eram ótimos em resolver tarefas com domínios específicos, porém tinham extrema dificuldade em lidar com o desconhecido. Principalmente pelo fato de que seu conhecimento derivava de uma série de regras implementadas manualmente (CHOLLET, 2017).

Um fator importante que contribuiu para o abandono das Redes Neurais na era da IA simbólica, foi a complexidade de treinamento dos modelos. Porém, em 1984, Hinton conseguiu o feito de treinar de forma eficaz um modelo de Rede Neural profundo (HEATON, 2015). Tão importante quanto o surgimento de algoritmos capazes de treinar modelos de Redes Neurais, foi o aumento significativo da capacidade computacional ao longo dos anos. Ambos, foram alguns dos principais fatores para o ressurgimento das Redes Neurais (NEAPOLITAN; JIANG, 2018).

Diante desse esclarecimento, podemos prosseguir e começar a compreender como o Aprendizado Profundo funciona. Primeiramente, vamos entender o que significa o termo ‘profundo’ da aprendizagem. Segundo Chollet (2017), ao contrário do aprendizado superficial, - onde o modelo possui uma camada de entrada, uma camada de saída e apenas uma ou duas camadas ocultas - o Aprendizado Profundo pode ter dezenas ou centenas de camadas ocultas no modelo.

Em relação às camadas ocultas, cada uma delas possuirá representações distintas da entrada. Isso funciona como uma espécie de filtragem, consequentemente aproximando-o cada vez mais do resultado esperado. Cada uma das camadas funciona de forma diferente da outra, isso ocorre devido à influencia dos pesos que lhe são atribuídos. Vale ressaltar que os pesos funcionam como parâmetros e, modificar um peso significa mexer na estrutura inteira do modelo, pois a saída de uma camada será a entrada de sua sucessora (CHOLLET, 2017).

Afinal, o que é aprender? Ainda segundo Chollet (2017), é mapear corretamente uma determinada entrada com a saída correta. Um ótimo exemplo é utilizado pelo próprio Autor, onde as entradas seriam imagens e as saídas rótulos, como por exemplo, ‘gato’. Heaton (2015) menciona que, geralmente, em Redes Neurais, utiliza-se algoritmos de aprendizado supervisionado - como é o exemplo acima mencionado - e também não-supervisionado - não utilizando dados rotulados -, inclusive, é possível realizar uma composição entre os dois, executando-os paralelamente, claro que isso dependerá da escalabilidade do modelo.

A forma como um modelo será treinado é muito exclusivo, afinal, cada um dependerá de uma base de dados diferente e algumas acabam sendo mais limitadas que outras. Um método comumente utilizado é inicializando o modelo com dados não-rotulados e complementando-o

com dados rotulados. Em resumo, os dados não-rotulados servem para inicializar os pesos das camadas e os dados rotulados acabam otimizando-os (HEATON, 2015).

É cada vez mais comum ver aplicações modernas tentando resolver problemas de classificação *multi-label*. Esse tipo de classificação diferencia-se da classificação binária pelas suas possibilidades de rótulos. A classificação binária possui apenas dois rótulos e a classificação *multi-label* possui três ou mais possibilidades (TSOUMAKAS; KATAKIS, 2009). Para resolver esse tipo de problema, as redes neurais são uma grande alternativa. Pois, elas já são nativamente preparadas para a classificação *multi-label*. Bibliotecas modernas, como por exemplo o *Keras* (Seção 5.1.1), fornecem alternativas que simplificam a declaração e avaliação desse tipo de classificação (BROWNLEE, 2020a).

Além disso, as redes neurais são ótimas para lidar com problemas que envolvem linguagem natural. Um de seus componentes - conhecido como *Embedding Layer* -, proporciona transformar palavras em vetores, cujos podem ser internamente manipulados e calculados (GOLDBERG, 2017). É importante ressaltar a influência das redes neurais na evolução do PLN (Seção 2.4.1), pois além dos resultados obtidos atualmente, existe a perspectiva de um futuro muito promissor com a utilização desses dois campos (DENG; LIU, 2018).

Conforme anteriormente descrito, existiram alguns fatores que foram primordiais para a popularização das redes neurais. A partir disso, segundo Ceccon (2020), o uso constante delas evidenciou que o modelo mais simples - popularmente conhecimento como rede neural profunda - não seria capaz de resolver todos os problemas existentes. Por isso, novas abordagens foram criadas, cada uma delas visando resolver um novo problema ou melhorar alguma rede já existente. Hoje, existem vários tipos de Redes Neurais, contudo algumas são bem mais populares que as outras. Em seguida, são apresentados os detalhes de algumas dessas redes:

- **Redes Neurais Convolucionais** (RNCs): Esse tipo de rede impactou de forma significativa a área de visão computacional, principalmente pela sua capacidade de processamento, onde é possível avaliar escala, rotação e ruído. É possível utilizá-la fora do campo de visão computacional, entretanto é necessário adaptar as entradas, pois esse tipo de rede foge do padrão tradicional, onde os vetores não necessariamente precisam estar ordenados. Para melhor representar isso, deve-se levar em consideração os *pixels* de uma imagem, cuja ordenação é crucial para que ela possua sentido (HEATON, 2015).
- **Redes Neurais Recorrentes** (RNNs): A principal característica desse tipo de rede neural é sua capacidade de memorizar estados. Para descrever seu comportamento, pode ser feita uma analogia com uma pessoa realizando uma leitura. Sempre que ela lê uma nova palavra em uma frase, um novo processamento é realizado, sempre levando em consideração o que já foi lido anteriormente (CHOLLET, 2017). É claro que esse tipo de rede não é adequado somente para lidar com textos, ela é indicada para resolver qualquer tipo de problema temporal, onde a saída é prevista baseando-se em eventos do passado, como por exemplo, indicar qual será a previsão do tempo através de um histórico

climático (CECCON, 2020).

- ***LSTM*** (*Long Short-Term Memory*): Teoricamente, as RNNs são excelentes alternativas para prever saídas baseadas em séries temporais, contudo, na prática, elas provaram-se insuficientes. Os dois problemas mais conhecidos são: o *vanish gradient* e o *exploding gradient* (PASCANU; MIKOLOV; BENGIO, 2013). A *LSTM* surge para solucionar os problemas das RNNs, proporcionando uma memória de longo prazo. Seu maior feito foi evitar que informações antigas sejam perdidas no decorrer do processamento (CHOLLET, 2017).

2.4.3 Aprendizagem Supervisionada

Essa técnica caracteriza-se por utilizar pares de dados, onde x é um exemplo de entrada e y é a saída esperada. Quando um modelo recebe esse conjunto de dados, ele começa criar diversas hipóteses para chegar ao valor de y, visto que a função que gerou y é desconhecida. Essa abordagem exige que além de um conjunto de dados para treinamento, exista um conjunto de dados para a posterior avaliação da acurácia do modelo. Vale ressaltar que, não necessariamente os valores de x e y precisam ser numéricos. (RUSSEL; NORVIG, 2013).

Para complementar essa explicação, Heaton (2015) fornece uma ótima analogia. Identificar um veículo não é uma tarefa difícil para uma criança, porém é improvável que ela saiba qual é o tipo dele, afinal ainda possui um conhecimento muito estreito acerca do mundo. Essa tarefa torna-se muito mais fácil quando um adulto auxilia na identificação, verbalmente rotulando os tipos de veículos. Após muitos exemplos ela acaba adquirindo a capacidade de identificá-los por conta própria.

3 METODOLOGIA

Nesta Seção, encontram-se as definições para resolver os problemas que permeiam este trabalho, são eles: “é possível prever - de forma automatizada - quais são as emoções presentes nos textos criados por uma pessoa nas redes sociais?” e “é concebível transformar essas previsões em representações que proporcionem encontrar indícios de transtornos mentais?”. Com base neles, duas Seções foram criadas, respectivamente cada uma delas apresentará como foi encontrada a solução para o seu problema.

3.1 Reconhecer emoções em textos

Para a resolução do primeiro problema, quatro etapas foram necessárias. A primeira etapa foi destinada à pesquisa de um modelo eficiente para o reconhecimento de emoções em textos. A segunda etapa concentrou-se em localizar um conjunto de dados, cujo deveria respeitar alguns critérios pré-estabelecidos que iam de encontro com os objetivos do projeto. Diante do conjunto encontrado, a terceira etapa teve enfoque na seleção das melhores técnicas para o pré-processamento dos dados. A quarta etapa foi utilizada para definir a configuração empregada no treinamento e avaliação do modelo.

3.1.1 Rede Neural

Através das pesquisas realizadas sobre o Aprendizado Profundo (Seção 2.4.2), considerou-se que as Redes Neurais eram um conjunto de técnicas muito promissor. Essa consideração foi estimulada pelos seguintes fatores: elas proporcionam ferramentas para lidar com linguagem natural (GOLDBERG, 2017), estão nativamente prontas para classificações *multi-label* (BROWNLEE, 2020a) e quando trata-se de PLN, seu desempenho é muito superior que o das outras abordagens da IA (DENG; LIU, 2018).

Dentre todas as possibilidades de arquitetura das Redes Neurais, a escolhida foi a *LSTM*. Segundo Li et al. (2017a), ela é uma variante das RNNs, cujas provaram ser a melhor alternativa para lidar com problemas que envolvem séries temporais, como por exemplo áudio e texto.

Na prática, as RNNs clássicas apresentaram dificuldades relacionadas à memória (Seção 2.4.2). A *LSTM* surgiu para corrigir esse problema, proporcionando uma memória de longo prazo (CHOLLET, 2017). Para isso, foram adicionadas células, também conhecidas como estados, acopladas com as camadas ocultas da rede, onde esses estados são atualizados constantemente durante o processo de aprendizagem (LANE; HAPKE; HOWARD, 2019). Ainda segundo Lane, Hapke e Howard (2019, p. 276, tradução nossa):

> Em LSTMs, as regras que governam as informações armazenadas no estado (memória) são as próprias redes neurais treinadas [...] Eles podem ser treinados

> para aprender o que lembrar, enquanto ao mesmo tempo o resto da rede recorrente aprende a prever o rótulo alvo! Com a introdução de uma memória e estado, pode-se começar a aprender dependências que se estendem não apenas um ou dois *tokens*, mas por toda amostra de dados. Com essas dependências de longo prazo preparadas, pode-se começar a ver além das próprias palavras e em algo mais profundo sobre a linguagem.

3.1.2 *Datasets*

Para treinar a Rede Neural, optou-se pela utilização da Aprendizagem Supervisionada (Seção 2.4.3). Diante disso, foi necessário escolher quais dados seriam utilizados no treinamento. O *dataset* selecionado surgiu do trabalho proposto por Bostan e Klinger (2018), onde foi realizada uma união de vários *datasets* distintos, todos propostos para a classificação de emoções.

Segundo Bostan e Klinger (2018), os *datasets* selecionados foram anotados de três maneiras diferentes. A primeira foi utilizando especialistas, cujos usam seu conhecimento para vincular um texto com uma emoção. A segunda forma é através do *crowdsourcing*, onde é feita uma coleta de dados vindos de múltiplos colaboradores. A terceira forma é pela supervisão distante. Ela faz uso de frases que possuem algo que caracterize uma suposta emoção, como por exemplo as *hashtags*.

Para compor o *dataset* utilizado neste trabalho, apenas alguns dos *datasets* selecionados por Bostan e Klinger (2018) foram escolhidos, são eles: Strapparava e Mihalcea (2007), Scherer e Wallbott (1994), Li et al. (2017b), Alm (2008) e Ghazi, Inkpen e Szpakowicz (2015). Todos possuem duas características semelhantes que, inclusive, foram os principais critérios de escolha: eles utilizam as seis emoções básicas de Ekman (parágrafo final da Seção 2.1) e suas anotações foram feitas por especialistas.

	texto	emocao
0	During the period of falling in love, each tim...	alegria
1	When I was involved in a traffic accident.	medo
2	When I was driving home after several days of...	raiva
3	When I lost the person who meant the most to me.	tristeza
4	The time I knocked a deer down - the sight of ...	nojo

Figura 1 – Primeiras linhas do *Dataset*.

A Figura 1 mostra as primeiras cinco linhas do *dataset*. Através dela, é possível visualizar algumas de suas características, como por exemplo o número de colunas que, apesar da imagem mostrar três, apenas duas são relevantes: **texto** e ***emocao***. Além das emoções que aparecem na figura - alegria, medo, raiva, tristeza e nojo -, também existem textos rotulados com a surpresa. A coluna texto é composta por frases em inglês.

Inicialmente, o *dataset* possuía um total de 31.774 registros. A Figura 2 mostra como ele estava distribuído. Com base nela, é possível notar a superioridade de registros que possuem o rótulo de alegria, contudo os outros rótulos possuem quantidades próximas, sem tanta discrepância.

Datasets desbalanceados costumam gerar modelos imprecisos e, um dos grandes problemas do desbalanceamento é mascarar essa imprecisão, porque a acurácia do modelo não necessariamente estará com um percentual baixo. Entretanto, na prática o modelo só será preciso prevendo os rótulos que aparecem em maior quantidade no *dataset*. Em contrapartida, as previsões dos rótulos que aparecem em menor quantidade serão imprecisas (TRIPATHI, 2019). Diante disso, 13.072 registros com o rótulo de alegria foram removidos. Essa remoção resultou em um *dataset* com 18.702 registros. Sua nova distribuição pode ser verificada na Figura 3.

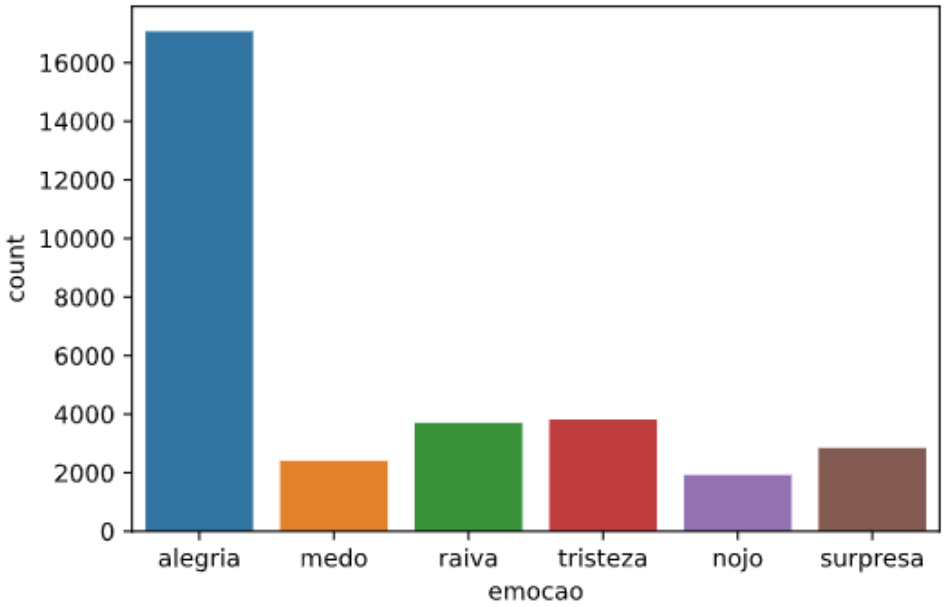


Figura 2 – *Dataset* desbalanceado.

3.1.3 Pré-processamento dos dados

O pré-processamento é uma etapa que realiza a remoção de tudo aquilo que é irrelevante e prepara os dados para o processo de aprendizagem. Para realizá-lo, o responsável deve estar ciente do que deve ser removido e de quais técnicas devem ser adotadas, logo ele pode ser considerado um processo semi-automático (BATISTA, 2003).

A primeira etapa do pré-processamento foi destinada à remoção de alguns caracteres especiais. Segundo Petit (2019), é importante manter os textos limpos, removendo tudo que for insignificante para o objetivo final. Para isso, foram utilizadas expressões regulares que, segundo Nagy (2018), são excelentes ferramentas para realizar a filtragem de dados. Elas proporcionam inúmeros recursos, como por exemplo validar entradas, fragmentar *strings* e a mais importante

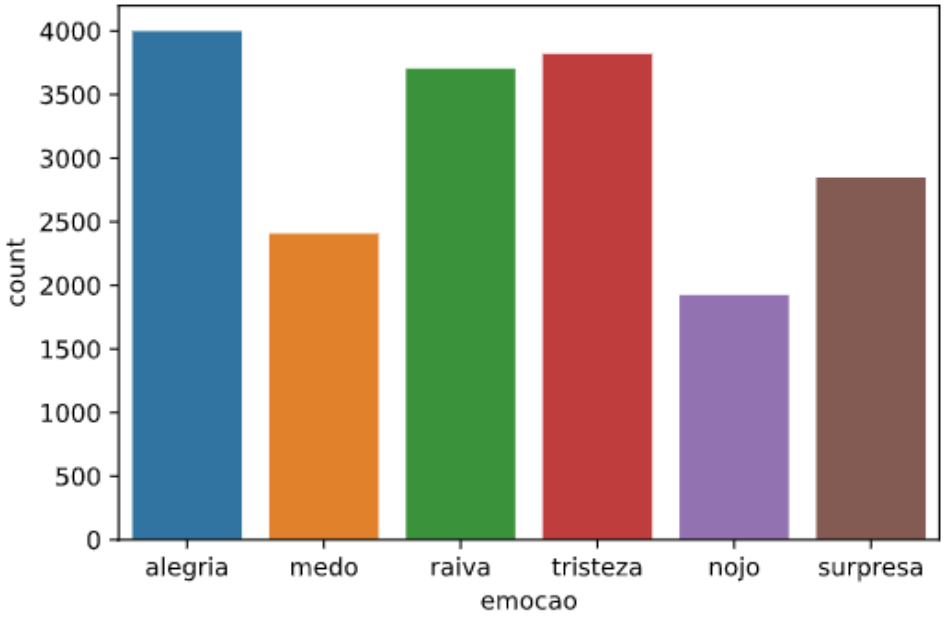


Figura 3 – *Dataset* balanceado.

nesse contexto de filtragem de dados, utilizar um padrão de pesquisa para remover caracteres específicos.

Datasets que possuem conteúdo textual costumam carregar consigo palavras que aparecem com muita frequência, elas são conhecidas como *stopwords* (MANNING; RAGHAVAN; SCHUTZE, 2008). A segunda etapa do pré-processamento foi utilizada para realizar a remoção desse tipo de palavra pois, segundo Kulkarni e Shivananda (2019), essa recorrência torna-as desinteressantes para o processo de aprendizagem e, é importante manter no *dataset* somente as palavras-chave. Além disso, Lane, Hapke e Howard (2019) mencionam que, tratando-se de desempenho, remover as *stopwords* reduz a quantidade de recursos computacionais utilizados.

A forma mais tradicional de encontrar *stopwords* é vasculhando manualmente o *dataset* e contando o número de vezes que as palavras aparecem (MANNING; RAGHAVAN; SCHUTZE, 2008). Entretanto, essa abordagem não foi optada para esse trabalho. A forma escolhida de removê-las foi encontrada em Lane, Hapke e Howard (2019), cujos fazem menção ao *NLTK* (*Natural Language Toolkit*), apresentando na sequência.

Segundo Bird, Klein e Loper (2009, p. 60, tradução nossa), "O *NLTK* inclui alguns corpora que nada mais são do que listas de palavras. [...] existe um corpus de palavras irrelevantes, ou seja, palavras de alta frequência". Com base nisso, optou-se pela utilização dessa lista de palavras para remover as *stopwords* dos textos do *dataset*.

Quanto maior for o *dataset*, consequentemente maior será o número de palavras distintas presentes nele. Por mais que todas sejam palavras-chave, ainda assim é interessante manter essa distinção enxuta. Pois, segundo Bengfort, Bilbro e Ojeda (2018), essa variedade de palavras pode ser prejudicial ao processo de aprendizagem. Diante disso, a terceira etapa

empregou-se em amenizar essa situação. Ainda segundo Bengfort, Bilbro e Ojeda (2018), o *stemming* é uma técnica interessante para resolver esse tipo de problema, visto que ele remove prefixos e sufixos presentes em *strings*, mantendo apenas uma *substring* representativa. Em outros termos, mantém-se apenas a raiz da palavra (KULKARNI; SHIVANANDA, 2019) ou mantém-se apenas seu radical (LANE; HAPKE; HOWARD, 2019).

Em Lane, Hapke e Howard (2019), é possível observar um exemplo que elucida o funcionamento do *stemming*, segundo os autores as palavras *house*, *houses* e *housing*, tornam-se *hous*. Kulkarni e Shivananda (2019) também apresentam um ótimo exemplo, segundo eles as palavras *fish*, *fishes* e *fishing*, tornam-se *fish*.

Após toda preparação dos textos, ainda falta uma importante etapa para que eles sejam representativos, afinal, segundo Lane, Hapke e Howard (2019), modelos de Rede Neural não podem fazer cálculos utilizando palavras. Para fazer a transição dos textos para a matemática, utilizam-se as representações vetoriais, onde todos os documentos (frases) são semelhantes, ou seja tem uma mesma origem.

Diante disso, a última etapa do pré-processamento foi destinada à escolha de uma técnica eficaz para o contexto atual. A escolhida foi o *Tf-idf* que, como bem apresentado por Manning, Raghavan e Schutze (2008), ela leva em consideração a quantidade de vezes que um termo aparece em um documento, a quantidade de vezes que um termo aparece em múltiplos documentos e também se o tamanho do documento é coerente com a recorrência de um termo.

3.1.4 Treinamento e avaliação

A etapa de treinamento e avaliação é onde unem-se todas as definições anteriores. Além disso, também determina-se quais serão os parâmetros utilizados e como será feita a divisão do *dataset* (LANE; HAPKE; HOWARD, 2019).

Referente à divisão do *dataset*, os percentuais foram encontrados em Lane, Hapke e Howard (2019), Deitel e Deitel (2019) e Kulkarni e Shivananda (2019). Logo, 80% destinou-se ao treinamento e 20% destinou-se à avaliação.

Os parâmetros necessários são: o *batch size* - quantidade de amostras necessárias para atualização dos parâmetros da rede (DATA SCIENCE ACADEMY, 2019) - e o número de épocas - quantidade de vezes que o *dataset* será percorrido durante o treinamento (DATA SCIENCE ACADEMY, 2019). Ambos foram encontrados através de experimentações com a Rede Neural, baseando-se na perda e na acurácia que estavam sendo apresentadas durante os treinamentos. Por fim, baseados nesses experimentos, definiu-se 32 para o *batch size* e 7 para as épocas.

3.2 Encontrar indícios de transtornos mentais

Tão importante quanto reconhecer as emoções nos textos, é utilizá-las para detectar possíveis indícios de transtornos mentais presentes no indivíduo que as propagou. Na Seção

2.1.1 já foi feita uma correlação de algumas emoções com alguns transtornos mentais, mostrando que elas estão diretamente ligadas com a saúde mental de uma pessoa.

Antes de prosseguir, é necessário esclarecer o que é considerado um indício. Segundo DICIO, Dicionário Online de Português (2020), “Sinal; aquilo que indica o que, provavelmente, ocorreu ou existiu: o paciente expressa indícios de febre;”. Dentro do contexto do trabalho, o seguinte exemplo ajudará esclarecer quando um indício é encontrado. Como pode ser verificado em Barnhill (2018), o diagnóstico dos Transtornos Fóbicos Específicos é repleto de características, como por exemplo: o indivíduo sente medo de um objeto específico, de forma exagerada; o medo propagado é incoerente com o real perigo apresentado pelo objeto; o medo foi propagado com intensidade durante, no mínimo, seis meses. Portanto, se o trabalho proporcionar que o usuário enxergue alguma dessas características - ou uma parte considerável delas -, considera-se que um indício foi encontrado.

Com base nisso, no decorrer desta Seção, serão descritos os métodos adotados para tentar localiza-los. Vale ressaltar que **o trabalho proposto não sinalizará que um indício foi encontrado**, somente fornecerá as ferramentas para que o usuário os localize.

Inclusive, tratando-se dos usuários, o trabalho não visa ser **útil somente para profissionais da área da saúde mental**, como por exemplo os psicólogos e os psiquiatras. Sua preparação almejou atender, também, outros tipos de usuários (denominados normais). Acredita-se que essa é uma ferramenta **útil na mão de familiares ou pessoas próximas de um determinado indivíduo que tenha dificuldade em expressar suas emoções**.

Com base nisso, imagina-se que o seguinte questionamento possa surgir no leitor: como é possível que um usuário normal (não profissional) consiga encontrar um indício de transtorno mental, visto que isso exige conhecimento científico? As representações propostas no trabalho não exigem nenhum tipo de conhecimento para que sejam interpretadas, logo é possível que um usuário normal consiga extrair resultados delas. Por exemplo, é possível ter uma indicação de que algo está errado com um indivíduo quando ele costuma propagar muita tristeza. Sem conhecimento técnico, dificilmente o usuário presumirá um transtorno mental específico, mas ele alimentará uma suspeita que pode desencadear numa aproximação com o indivíduo em questão, para que, se necessário, posteriormente ele seja encaminhado para um profissional da área de saúde mental.

Para resolver esse problema, duas etapas foram necessárias. Primeira, localizar características dos transtornos mentais que possam ser computacionalmente representadas. Segunda, representá-las de forma adequada, para que, de fato, os indícios tornem-se perceptíveis.

Conforme verificado em AMERICAN PSYCHIATRIC ASSOCIATION (2014), existem vários tipos de transtornos mentais. É impossível abordá-los em sua totalidade, pois além de seres abundantes, cada qual possui suas peculiaridades. Diante disso, as representações foram pensadas baseando-se em quatro transtornos mentais, são eles:

- Transtorno Disruptivo da Desregulação do Humor (TDDH)

- Transtorno de Ansiedade Generalizada (TAG)
- Fobia Específica
- Transtorno Depressivo Maior (TDM)

Entretanto, o sistema não está restrito aos mesmos, ou seja, é possível que indícios de outros transtornos, cujas características sejam compatíveis com os métodos aqui adotados, também possam ser encontrados. Após a analise das características presentes nesses quatro transtornos, notou-se que elas poderiam ser encontradas através de duas representações: uma *timeline* e um gráfico.

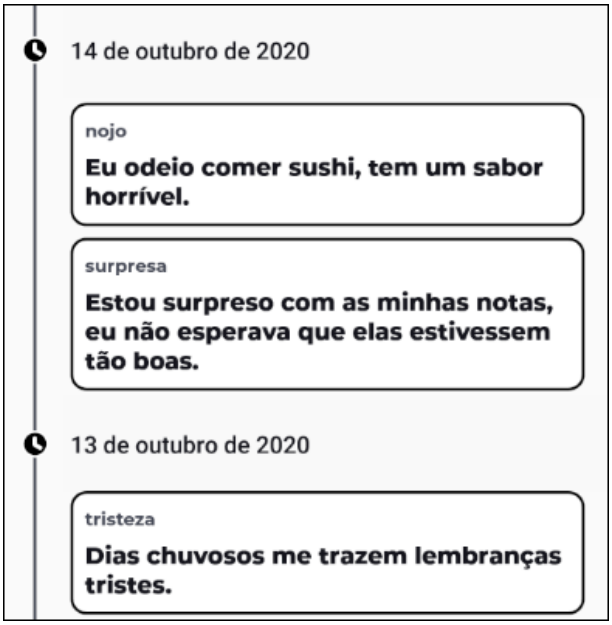


Figura 4 – *Timeline*.

A *timeline* pode ser visualizada na Figura 4. Ela agrupa o conteúdo textual pelo dia de sua criação. Além disso, todos os textos possuem um rótulo com a emoção predominante detectada. O gráfico pode ser visualizado na Figura 5, onde são agrupadas as emoções propagadas durante um mês. Cada coluna apresenta o número de vezes que aquela emoção apareceu.

Em seguida, serão apresentadas as características (consideradas previsíveis) dos transtornos mentais. Após cada apresentação, demonstra-se como é possível encontrá-las utilizando as representações criadas no trabalho.

O primeiro transtorno mental selecionado é o TDDH. Ele faz parte do grupo dos Transtornos Depressivos. As características encontradas em AMERICAN PSYCHIATRIC ASSOCIATION (2014, p. 156) são:

1. “A. Explosões de raiva recorrentes e graves manifestadas pela linguagem [...]”
2. “C. As explosões de raiva ocorrem, em média, três ou mais vezes por semana.”
3. “D. O humor entre as explosões de raiva é persistentemente irritável ou zangado na maior parte do dia, quase todos os dias [...]”

Figura 5 – Gráfico Mensal.

4. “E. Os Critérios A-D estão presentes por 12 meses ou mais. Durante esse tempo, o indivíduo não teve um período que durou três ou mais meses consecutivos sem todos os sintomas dos Critérios A-D.”

Como encontrá-las: a *timeline* não mostra a intensidade da emoção, somente qual foi predominante, mas através do texto é possível observar quão agressivas foram as palavras utilizadas (1). Ela também proporciona visualizar se as explosões de raiva ocorreram dentro da frequência semanal esperada (2) e se nos intervalos a raiva é a emoção mais propagada (3). O gráfico não informa quantas explosões de raiva ocorreram durante o mês, contudo ele provê uma visualização mais ampla, possibilitando enxergar se a raiva é a emoção que preponderou durante os meses (4).

O segundo transtorno mental selecionado é o TAG. Ele faz parte do grupo dos Transtornos de Ansiedade. As características encontradas em AMERICAN PSYCHIATRIC ASSOCIATION (2014, p. 222) são:

1. “Ansiedade e preocupação excessivas (expectativa apreensiva), ocorrendo na maioria dos dias por pelo menos seis meses, com diversos eventos ou atividades (tais como desempenho escolar ou profissional).”

2. Além de ansioso e preocupado, o indivíduo mostra-se diariamente irritado. O período é equivalente ao item anterior.

Antecedendo a explicação de como encontrá-las, algumas considerações importantes precisam ser feitas. A ansiedade e a preocupação não estão presentes nas seis emoções básicas de Ekman, ou seja, não é possível prevê-las. Contudo, como bem assegurado por Clark e Beck (2012), o medo é uma emoção que está subjacente ao ciclo de ansiedade. Em Miguel (2015), ansiedade e preocupação são pertencentes a categoria do medo, pois elas possuem características que se assemelham com ele.

Como encontrá-las: a *timeline* possibilita que seja feita uma análise diária das emoções, contribuindo para localizar os dois itens. O gráfico também contribui para ambos, pois ele possibilita ver mensalmente a predominância do medo e da raiva.

O terceiro transtorno mental selecionado é a Fobia Específica. Ele faz parte do grupo dos Transtornos de Ansiedade. As características encontradas em AMERICAN PSYCHIATRIC ASSOCIATION (2014, p. 197) são:

1. "Medo ou ansiedade acentuados acerca de um objeto ou situação (p. ex., voar, alturas, animais, tomar uma injeção, ver sangue)."
2. "O medo ou ansiedade é desproporcional em relação ao perigo real imposto pelo objeto ou situação específica e ao contexto sociocultural."
3. "O medo, ansiedade ou esquiva é persistente, geralmente com duração mínima de seis meses."

Como encontrá-las: a *timeline* é útil para encontrar os dois primeiros itens (1 e 2). Ela possibilita ver qual é o contexto da frase (por exemplo, o indivíduo está falando de aranhas) e se subjacente ao texto, encontra-se o medo - detectado pelo próprio sistema. Através do contexto observado, o próprio usuário pode concluir se o medo ou a ansiedade são desproporcionais. O contexto (fator causador do medo) não estará presente na representação gráfica, mas ela contribuirá para a análise da recorrência do medo, em uma escala mensal (3).

O quarto transtorno mental selecionado é o TDM. Ele faz parte do grupo dos Transtornos Depressivos. As características encontradas em AMERICAN PSYCHIATRIC ASSOCIATION (2014) são:

1. Durante o período de duas semanas, o indivíduo demonstrou mudança em alguns aspectos, como por exemplo: mostra-se mais deprimido e desinteressado. Além disso, propaga constantemente sentimentos de culpa e pensamentos relacionados com a morte, tanto o medo dela, quanto o desejo.
2. Apesar do período mínimo ser de duas semanas, os episódios podem ser mais duradouros.

Como encontrá-las: a *timeline* é capaz de mostrar - através do próprio texto - o recorrente desinteresse de um indivíduo e suas possíveis manifestações relacionadas com a morte. Ela também pode detectar o medo (vinculado à morte) e a predominância da tristeza. O gráfico tende ser útil para episódios mais duradouros, onde é necessário visualizar a preponderância da tristeza.

Como pode ser observado nos exemplos acima, olhar para as representações e encontrar indícios de um transtorno mental específico, exige conhecimento científico. Mas, mesmo sem esse conhecimento, ainda assim é possível extrair informações dessas representações, afinal elas não são complexas. Com base nisso, acredita-se que o gráfico será mais importante para usuários normais do que a *timeline*. Afinal, ele mostra de forma simples o número de vezes que uma emoção foi propagada durante um mês. Emoções propagadas de forma abundante, certamente soarão como um alerta para esses usuários que, por sua vez, podem tomar alguma atitude em prol do bem-estar de um indivíduo.

Por fim, vale ressaltar que, em nenhum momento o sistema fará algum diagnóstico ou alguma menção à indícios encontrados. Ele somente entregará subsídios para que o próprio usuário do sistema enxergue-os.

4 Aplicação

Como descrito na Seção 1, o objetivo deste trabalho é reconhecer emoções em textos e transformá-las em representações que proporcionem encontrar indícios de transtornos mentais. Para isso, decidiu-se que um aplicativo seria desenvolvido, alguns dos motivos são: sua flexibilidade, mobilidade e desburocratização pois, dispositivos móveis costumam estar boa parte do tempo com as pessoas.

Inclusive, o fato dos dispositivos móveis estarem presentes de forma constante no cotidiano das pessoas, proporcionou uma mudança gigantesca na forma em que elas comunicam-se. Hoje em dia, é muito comum ver as pessoas comunicando-se de forma assíncrona, pois elas sabem que quando encaminharem-na uma resposta, receberão uma notificação em seu dispositivo móvel (SEBRAE, 2020). Esse fator vai de encontro com os objetivos pretendidos pelo sistema aqui proposto, afinal reconhecer emoções em textos é um trabalho um tanto quanto custoso, logo será necessário criar uma tarefa e notificar o usuário somente quando ela for concluída. O fato dos usuários estarem com o sistema consigo - em seus respectivos dispositivos -, sem dúvidas proporcionará uma melhor experiência.

O aplicativo tem o objetivo de conduzir o usuário na busca pelos resultados. Inicialmente, é necessário encontrar os perfis de uma pessoa (familiar, amigo, colega, conhecido, paciente, etc...) nas redes sociais, cujos serão utilizados posteriormente. Para facilitar o encontro, o aplicativo mostra na tela o maior número de informações possíveis, todas fornecidas pelas próprias redes sociais. Por exemplo, ao iniciar uma busca pelo perfil do indivíduo no twitter, imediatamente o aplicativo mostrará seu *username*, seu nome completo, sua localização, seu número de postagens - conhecidos como *tweets* -, a quantidade de seguidores e a quantidade de amigos - pessoas que o indivíduo segue. No caso de uma busca no Reddit, será mostrado seu *username* e informações das interações que a pessoa costuma ter na rede social, denominadas de *Karma*.

Para encontrar um indivíduo é necessário saber qual é o seu *username* completo. O sistema não fará buscas pelo nome, endereço ou parte do *username*. Logo, precisa-se levar em consideração que não necessariamente um indivíduo será localizado, seja pelo fato do usuário não saber seu *username*, informá-lo incorretamente ao sistema ou o indivíduo não possui um perfil na rede social. A partir do momento em que o indivíduo foi localizado, em ao menos uma das redes sociais, as previsões podem ser iniciadas.

Antes de solicitar que o sistema inicie as previsões, o usuário tem a possibilidade de informar quantos textos ele gostaria que fossem analisados. Supondo que nenhum valor seja informado, o sistema - por padrão - pegará os últimos dez textos pois, considera-se que essa é uma quantidade aceitável para que as previsões sejam significativas. Após informar a quantidade desejada e pressionar o botão de iniciar as previsões, o sistema encarrega-se do restante do processo. Onde, pressupondo que o indivíduo foi localizado nas duas rede sociais,

serão buscados os *tweets* e os comentários do Reddit produzidos por ele.

Com o conteúdo textual armazenado temporariamente no aplicativo, iniciam-se as previsões. O tempo necessário para realizá-las dependerá da quantidade de textos que foi solicitada pelo usuário. Quando as previsões forem finalizadas, envia-se uma notificação ao usuário para que ele possa conferir seus resultados.

Os resultados são apresentados em forma de uma *timeline* e de gráficos que agrupam mensalmente a quantidade de vezes que uma emoção apareceu. Essas representações são melhor explicadas na Seção 3.2, além disso, na mesma Seção encontram-se exemplos de como é possível utilizá-las para encontrar os indícios de transtornos mentais.

O aplicativo possui um histórico, onde estão armazenadas todas as previsões que foram realizadas naquele dispositivo. Ao abri-las, é possível visualizar tanto a *timeline*, quanto o gráfico.

Apesar do aplicativo ser quem conduz o usuário durante toda a jornada, ele é responsável por apenas uma parte das tarefas realizadas pelo sistema. Existem duas partes ocultas que são essenciais para o todo, são elas: a *API* (*Application Programming Interface*) e a Rede Neural. Ambas serão melhor explicadas na Seção 5. No decorrer desta Seção, encontra-se toda parte conceitual do sistema, ou seja, quais são seus elementos e como eles comunicam-se.

4.1 Diagramas

Para realizar a construção da parte conceitual do sistema, fez-se uso dos conceitos da UML (*Unified Modeling Language*). Ela é composta por um conjunto de técnicas ideais para modelar *softwares*, principalmente aqueles que foram construídos utilizando a Programação Orientada a Objetos. Sua função principal é servir como um esboço do projeto, onde são apresentados todos os elementos que o compõem (FOWLER, 2005). Além disso, ela caracteriza-se por sua flexibilidade, possibilitando que qualquer tipo de domínio seja atendido (GUEDES, 2011).

4.1.1 Atores

Um sistema é composto por recursos - ou funcionalidades -, quem interage com um ou muitos recursos, denomina-se ator. O ator pode ser considerado uma representação de um usuário (FOWLER, 2005). Contudo, ele não necessariamente é uma pessoa, outros sistemas, componentes, dispositivos também podem ser considerados atores. Além da interação com o sistema em si, atores sempre são figuras externas, ou seja não compõem o sistema (WAZLAWICK, 2015).

A Figura 6 apresenta quais são os atores do sistema. Tanto o profissional, quanto o normal herdam de usuário, afinal eles usufruem dos mesmos recursos. Entretanto, o acréscimo de conhecimento técnico por parte do profissional proporcionará uma interpretação mais

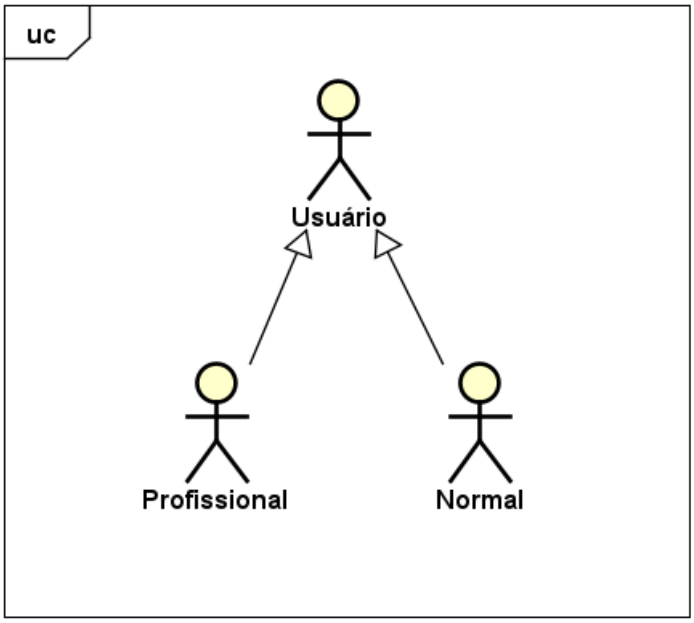


Figura 6 – Atores do sistema.

refinada dos recursos. Uma melhor explicação dessas diferenças pode ser encontrada na Seção 3.2.

4.1.2 Casos de uso

Os diagramas de casos de uso são representações que proporcionam expor quais são as funcionalidades de um sistema. Além disso, através deles é possível demonstrar como funcionam as interações entre as funcionalidades e os demais elementos, como por exemplo, os atores. Esses diagramas são muito importantes para o desenvolvimento de *software*, pois proporcionam uma visão clara do que um usuário precisa no sistema (BEZERRA, 2002). Inclusive, isso é complementado por Fowler (2005, p. 104), “"Os casos de uso são uma técnica para captar os requisitos funcionais de um sistema. Eles servem para descrever as interações típicas entre os usuários de um sistema e o próprio sistema, fornecendo uma narrativa sobre como o sistema é utilizado.”.

A Figura 7 apresenta o diagrama de casos de uso geral do sistema. Ou seja, ela mostra todas as funcionalidades presentes no mesmo. Além disso, é possível observar como o ator “Usuário” interage com elas.

As duas principais funcionalidades do sistema são o “Pesquisar Perfis nas Redes Sociais” e o “Prever Emoções”. O “Pesquisar Perfis nas Redes Sociais” precisa ser executado obrigatoriamente antes da execução do “Prever emoções”, visto que as emoções previstas derivam do conteúdo textual produzido por perfis de redes sociais. Logo, ao menos um dos dois perfis precisa ser previamente encontrado, possibilitando que o “Prever emoções” busque seu conteúdo textual e execute as demais operações. Para um maior esclarecimento, ambas

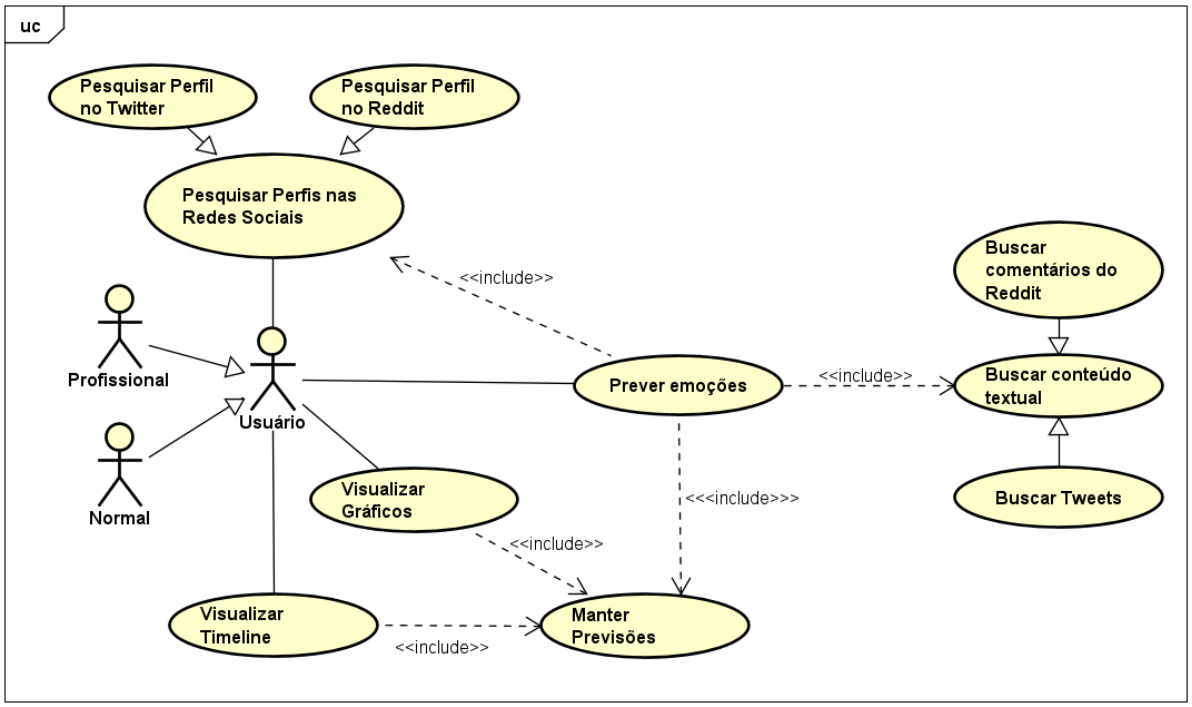


Figura 7 – Diagrama de casos de uso geral.

funcionalidades tiveram seu funcionamento detalhado nas Figuras 8 e 9, respectivamente.

4.1.3 Sequência

O diagrama de sequência é uma representação sequencial da comunicação entre as classes pertencentes a um caso de uso. Seu objetivo é mostrar os entremeios do evento, expondo como ele inicia e como ele chega ao fim. Além disso, ele mostra quais são os gatilhos disparados pelos atores que, inclusive, são os responsáveis por desencadear os eventos (GUEDES, 2011).

Em conformidade com a Seção 4.1.2, foram criados os diagramas de sequência para as duas principais funcionalidades do sistema: “Pesquisar Perfis nas Redes Sociais” e “Prever Emoções”. Eles podem ser visualizados nas Figuras 10 e 11.

4.1.4 Classe

Esse tipo de diagrama é responsável por mostrar quais são as classes subjacentes ao sistema. Além disso, demonstra como ocorrem os relacionamentos entre as mesmas. Sem dúvida ele é o mais popular dos diagramas, por conta disso acaba sofrendo diversas variações entre os projetos que o implementam (FOWLER, 2005). Sua popularidade vem de sua importância, afinal o diagrama de classes serve de apoio para diversos outros tipos de diagrama (GUEDES, 2011).

Para uma melhor representação de como o sistema foi proposto, dois diagramas de

Descrição de Casos de Uso

Caso de Uso:	Pesquisar perfil nas redes sociais
Ator(es):	Usuário
Pré-condições:	Sem pré-condições.
Pós-condições:	O perfil é localizado e seus dados são armazenados temporariamente.

	Ator		**Sistema**	
1	Informa o *username* desejado.			
		2	Comunica-se com a Api da rede social.	
		3	Obtém como resposta os dados do perfil.	E1
		4	Apresenta em tela os dados do perfil.	
5	Através dos dados em tela, verifica se é o perfil correto e seleciona-o.			A1
		6	Armazena temporariamente os dados do perfil selecionado.	
		7	Finaliza o caso de uso.	

E1	Nenhum perfil foi localizado com o *username* informado.
A1	O perfil encontrado não é o esperado, o usuário descarta-o e o caso de uso é reiniciado.

Figura 8 – Descrição do caso de uso pesquisar perfil.

classe foram criados. O primeiro é apresentado na Figura 12, ele mostra todas as classes que estão presentes na *API*. O segundo é apresentado na Figura 13, ele mostra todas as classes que estão presentes no aplicativo.

Descrição de Casos de Uso

Caso de Uso:	Prever emoções presentes em textos produzidos por perfis em redes sociais
Ator(es):	Usuário
Pré-condições:	Os perfis já foram localizados
Pós-condições:	As emoções são reconhecidas através dos textos

	Ator		**Sistema**	
1	Pressiona o botão de iniciar previsões.			
		2	Através dos perfis previamente localizados, busca o conteúdo textual produzido por eles.	E1
		3	Unifica e prepara os textos.	
		4	Envia os textos para que a rede neural extraia as emoções.	
		5	Armazena as previsões.	
		6	Cria uma mensagem para notificar o usuário que as previsões foram concluídas com sucesso.	
		7	Finaliza o caso de uso.	

E1	Nenhum conteúdo textual é localizado.

Figura 9 – Descrição do caso de uso prever emoções.

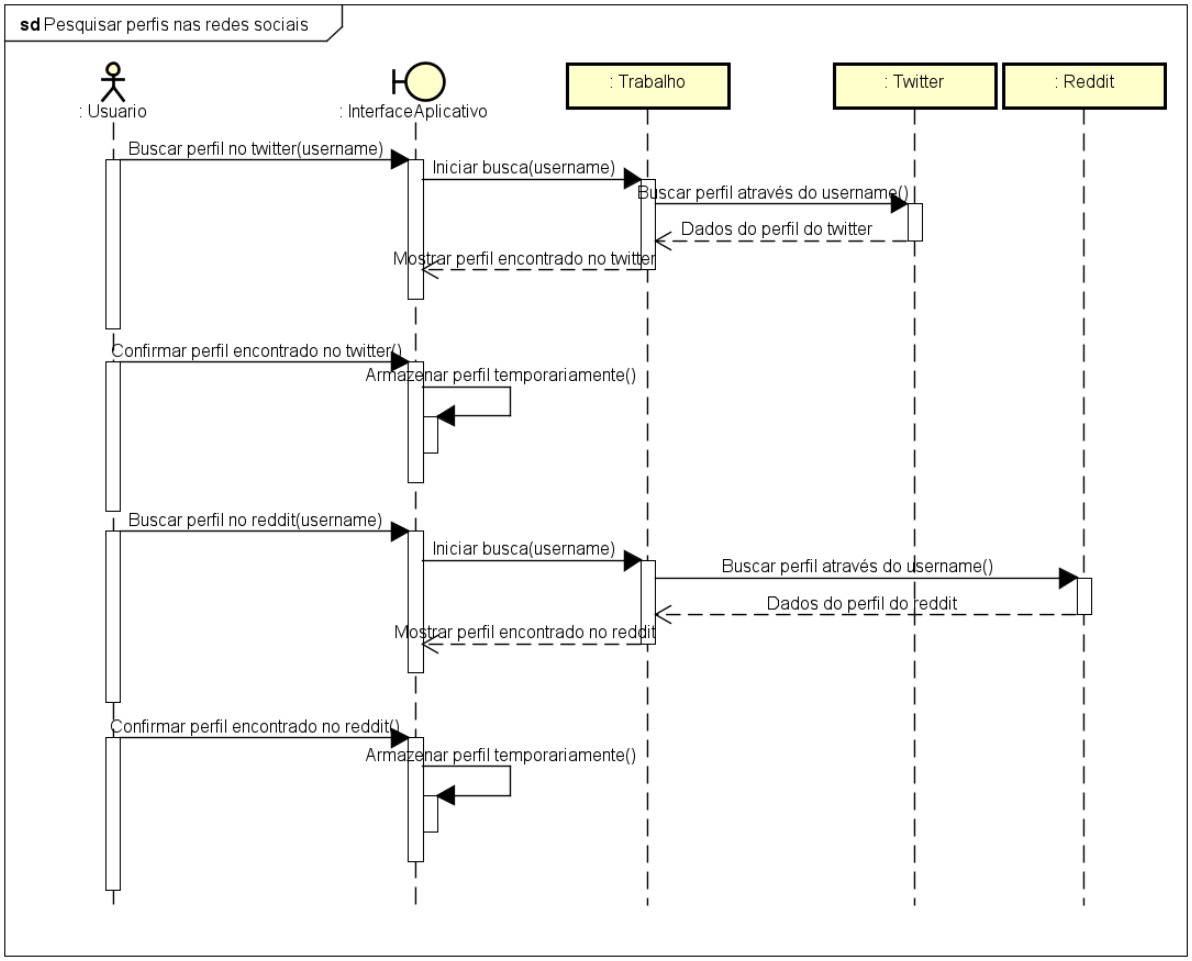


Figura 10 – Diagrama de sequência pesquisar perfil.

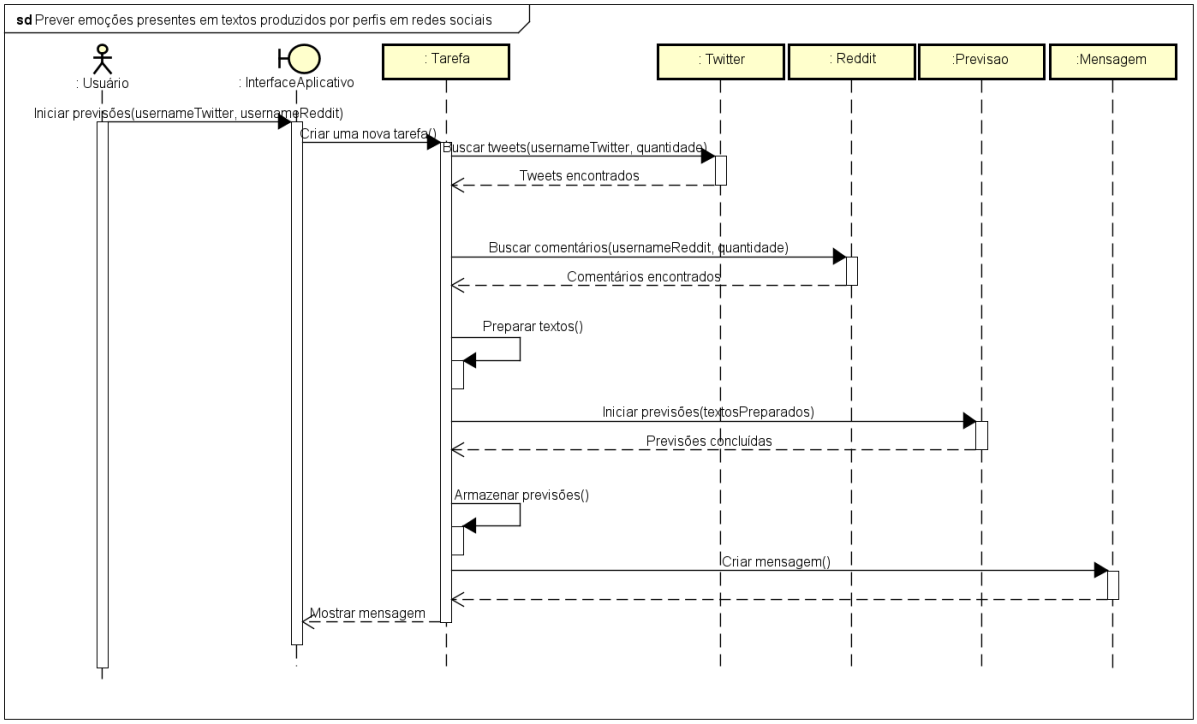


Figura 11 – Diagrama de sequência prever emoções.

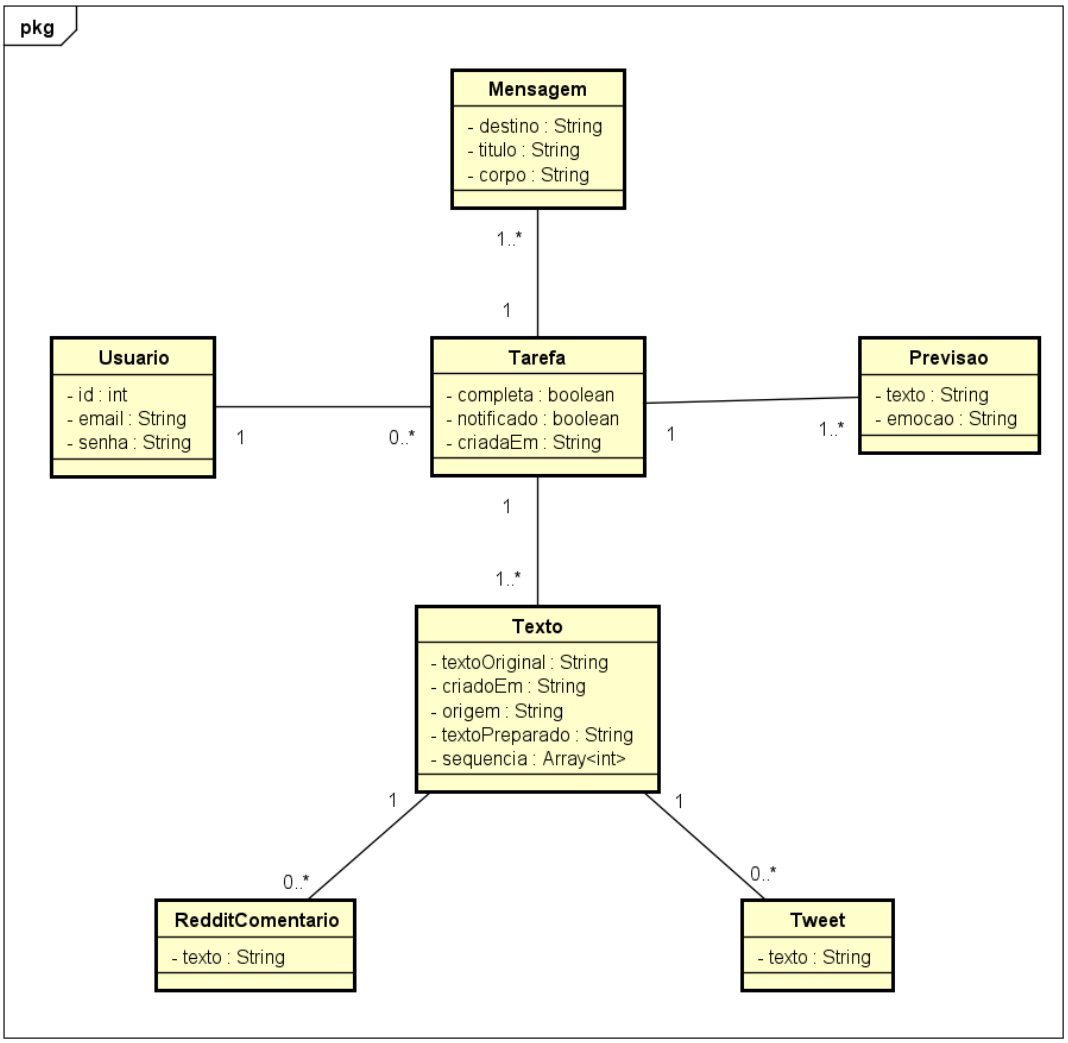


Figura 12 – Diagrama de classes da *API*

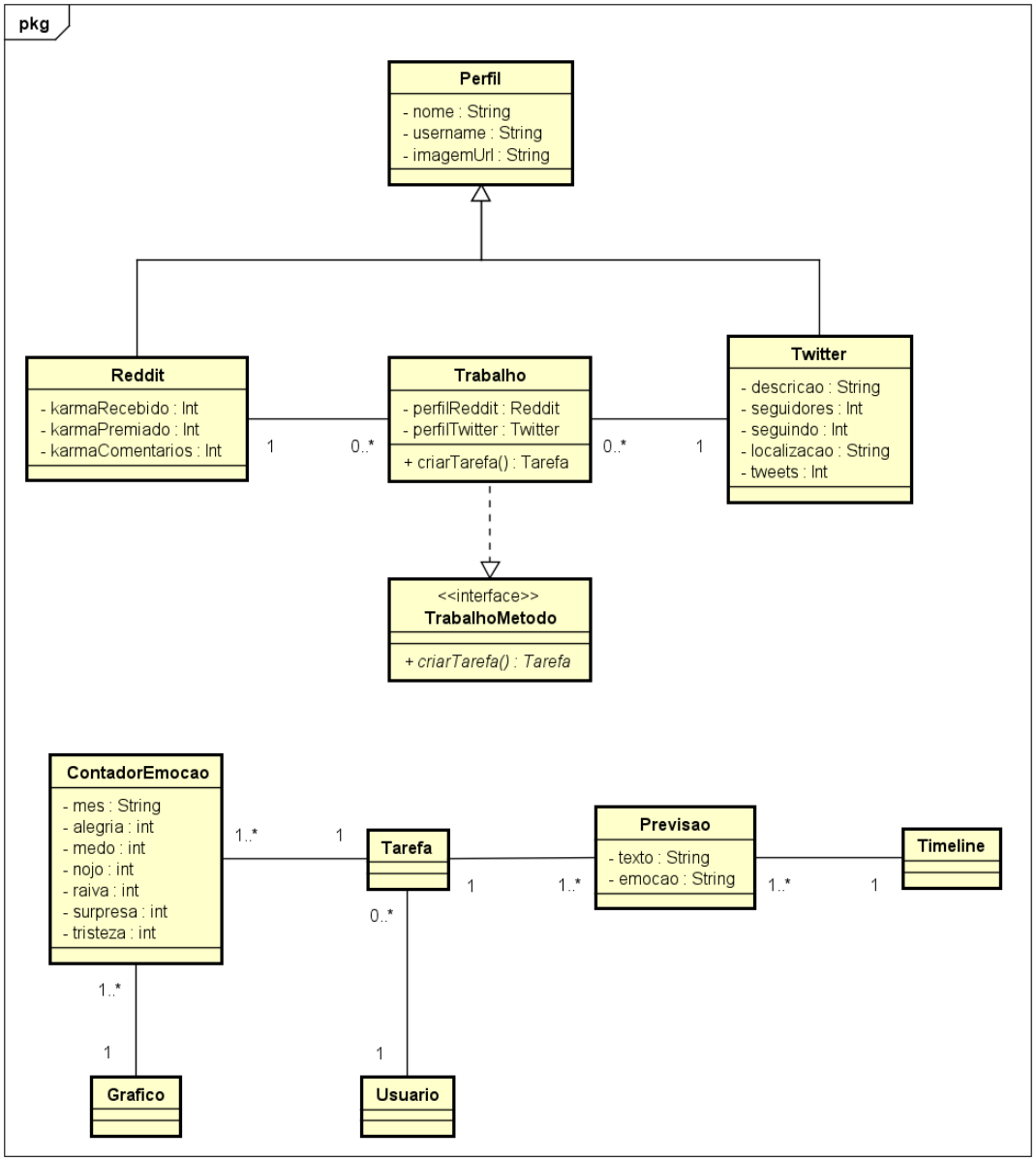


Figura 13 – Diagrama de classes do aplicativo.

5 Desenvolvimento

Neste Capítulo, apresenta-se como as teorias anteriormente descritas foram postas em prática. Contudo, antes de expor o desenvolvimento, todas tecnologias utilizadas foram apresentadas. Na sequência, estão os detalhes da Rede Neural, do *backend*, da prototipação do aplicativo e da sua programação, respectivamente.

5.1 Tecnologias

Esta Seção dedica-se em mostrar quais foram as tecnologias utilizadas no desenvolvimento do sistema. Foram destacadas as características que influenciaram na sua escolha, juntamente com seus respectivos contextos de uso.

5.1.1 *Python*

É uma linguagem de programação, criada em 1991. Inicialmente, não era tão popular, entretanto, atualmente é uma das linguagens de programação mais populares no mercado e na área acadêmica. Ela é extremamente abrangente, suprindo diversas áreas do desenvolvimento de *software* (MCKINNEY, 2018).

O *Python* pode ser considerado o conjunto de ferramentas ideal para desenvolvedores que lidam com aprendizagem de máquina, processamento de linguagem natural e análise de dados. Ele possui inúmeras bibliotecas que facilitam realizar esse tipo de tarefa. Um exemplo de ferramenta é o *Jupyter Notebook* [1], cujo proporciona um ambiente de desenvolvimento extremamente ágil e interativo. Além dele, existem inúmeras bibliotecas que proporcionam gráficos e representações, todas são de grande valia para averiguar os resultados da aplicação (MüLLER; GUIDO, 2016).

Outras características importantes são destacadas por Deitel e Deitel (2019). Segundo os autores, o *Python* é uma tecnologia de código aberto, logo, ela é gratuita, o que torna-a acessível para todos. Ele possui uma vasta comunidade de desenvolvedores que, inclusive, costumam compartilhar abertamente seus projetos. O mercado para essa tecnologia está muito aquecido, colaborando para que novas pessoas utilizem-na, por consequência ela tende a evoluir ainda mais. Existem dois *frameworks* muito populares para desenvolvimento de aplicações web que são feitos em *Python*, são eles: o *Flask* e o *Django*.

Ao contrário do que é comumente encontrado nos *frameworks* de desenvolvimento web, o *Flask* possui uma estrutura extremamente enxuta e sólida. Nela, encontram-se somente os componentes necessários para que a aplicação possa ser executada e depurada. Contudo, existe uma vasta quantidade de extensões que são facilmente acopladas com o *Flask*, por

[1] Ferramenta de desenvolvimento interativa, onde é possível misturar trechos de código com anotações (VIANA, 2017)

exemplo autenticação, formulários e banco de dados. Essa filosofia do *Flask* transforma-o em uma excelente alternativa para o desenvolvimento web, pois permite que o desenvolvedor decida o que é realmente necessário para sua aplicação (GRINBERG, 2018).

Dentre as bibliotecas mais populares do *Python*, encontra-se o *Keras*. Ele caracteriza-se pela sua eficácia e praticidade. Através dele é possível criar praticamente todos os tipos de combinações de Rede Neural, destaque para as RNCs e as RNNs. Outra característica interessante do *Keras*, é seu suporte às CPUs e GPUs, cujo proporciona que os modelos sejam treinados com rapidez. O protagonismo do *Keras* é evidenciado pelas suas conquistas, como por exemplo o fato de estar presente em projetos do Google, da Netflix e do Uber. Além disso, em competições de Aprendizagem Profunda realizadas recentemente, todos os vitoriosos utilizam-no (CHOLLET, 2017).

O *NLTK* é outra biblioteca de extrema relevância para a comunidade acadêmica. Inclusive, seu surgimento foi na Universidade da Pensilvânia, em 2001. Pelo fato de ser uma biblioteca de código aberto, expandiu-se rapidamente com as contribuições de inúmeros desenvolvedores. Ele caracteriza-se pela facilidade de uso, afinal sua estrutura é toda modularizada, possibilitando que os usuários utilizem somente o que é realmente necessário, despreocupando-se com o restante dos recursos. Além disso, os responsáveis procuram mantê-lo sempre atualizado, em conformidade com os avanços do PLN (BIRD; KLEIN; LOPER, 2009).

Em McKinney (2018), é possível encontrar duas importantes bibliotecas para lidar com dados, o *Pandas* que, segundo o Autor, tem como principal objetivo simplificar a manipulação de dados, fornecendo diversos métodos que agilizam o trabalho de um desenvolvedor e, o *sklearn*, cujo possui inúmeras ferramentas para lidar com a aprendizagem de máquina. Nele, encontram-se uma série de algoritmos e recursos que possibilitam avaliar com eficácia os modelos criados.

Além dessas bibliotecas abrangentes para o Aprendizado Profundo, o PLN, a ciência de dados e o desenvolvimento web, existem outras feitas em *Python* que possuem objetivos bem específicos. Dois exemplos são: o Tweepy (2020) e o PRAW (2020). O primeiro funciona como uma interface para todos os recursos proporcionados pela Api do Twitter. O segundo, semelhante ao primeiro, funciona como uma interface para os recursos proporcionados pela Api do Reddit. Os dois abstraem a forma como os recursos são utilizados, proporcionando muita praticidade aos desenvolvedores.

5.1.2 *Kotlin*

O *Kotlin* é uma linguagem de programação desenvolvida pela *Jetbrains*, empresa que destacou-se no desenvolvimentos de ferramentas para diversas outras linguagens de programação. A ideia do *Kotlin* surgiu em 2010, quando seus engenheiros notaram a necessidade de uma nova tecnologia no mercado. A ideia foi unir o melhor das tecnologias existentes, dando ênfase no design da linguagem e na facilidade de aprendizado da mesma

(JEMEROV; ISAKOVA, 2017).

Ele destaca-se pela interoperabilidade com o *Java*, inclusive, essa característica tem um motivo específico. Os desenvolvedores da *Jetbrains* consideraram toda base de código *Java* existente na empresa, ela era de grande valia e não podia ser desperdiçada ou deixada de lado. Logo, *Kotlin* nasceu não somente para interoperar com *Java*, mas para ser executado em qualquer ambiente no qual o *Java* pode ser executado (JEMEROV; ISAKOVA, 2017).

Apesar de ser uma linguagem relativamente recente, o *Kotlin* já superou o *Java* em alguns aspectos, o primeiro deles ocorreu em 2017, quando foi anunciado como a nova linguagem de programação oficial do Android, superando, inclusive, a *GoLang* [2]. Tratando-se da parte técnica, o *Kotlin* possibilita que um desenvolvedor escreva uma quantidade muito menor de código para executar tarefas. Também, o *Kotlin* conseguiu resolver um erro clássico do desenvolvimento *Java*, o *NullPointerException* que, basicamente, é quando a aplicação tenta acessar algum objeto inexistente em memória (RESENDE, 2018). A solução adotada pelos engenheiros foi implementar um sistema que permite declarar tipos que podem ser nulos e tipos que não podem ser nulos. Além disso, seu compilador proporciona segurança aos desenvolvedores, afinal ele procura erros previsíveis que não eram encontrados por outras tecnologias. (SUBRAMANIAM, 2019).

Aplicações voltadas para dispositivos móveis precisam lidar com tarefas assíncronas, geralmente efetuando requisições para algum serviço armazenado em nuvem. Para isso, a alternativa fornecida pelo *Kotlin* são as *coroutines*, uma biblioteca eficiente e segura para esse tipo de tarefa (SUBRAMANIAM, 2019).

Inicialmente, recomendava-se utilizá-lo no *backend* de aplicações web e no desenvolvimento Android. Contudo, hoje ele é uma linguagem ampla que, inclusive, consegue ser executada no *iOS* e também no navegador, quando compilada para *Javascript*. Sua segurança e simplicidade de escrita torna-a uma excelente alternativa para o desenvolvimento de aplicações que precisam ser disponibilizadas rapidamente (JEMEROV; ISAKOVA, 2017). Em caso de expansão, ainda assim é uma excelente opção, pois seu design elegante e conciso proporciona um código altamente legível e fácil de dar manutenção (SUBRAMANIAM, 2019).

5.1.3 *PostgreSQL*

Ele é uma continuação do banco de dados relacional *Ingres* que, originou-se na "Universidade da Califórnia, Berkeley" (UCB), em 1977. Entretanto, antes de ser nomeado como *PostgreSQL*, seu nome era *Postgres95*. Sua nomenclatura foi modificada a partir de sua popularização, em 1996, após seus desenvolvedores abrirem-no para a ajuda de outros desenvolvedores voluntários (STONES; MATTHEW, 2005). Segundo Shaw (2014), esse foi o momento de disrupção, onde ele deixou de ser um banco de dados acadêmico e começou a dar seus primeiros passos no mundo empresarial.

[2] Linguagem de programação de alto desempenho criada por engenheiros do Google (PEDRO, 2019)

O *Postgres* mantém-se com o código aberto, possibilitando que diversos desenvolvedores contribuam para sua constante evolução. Ele possui uma licença permissiva, proporcionando que os usuários utilizem-no de diversas formas e, até mesmo, customizem seu código (OBE; HSU, 2017). Além dos desenvolvedores independentes, existem várias empresas do mercado que contribuem de forma regular no desenvolvimento do *PostgreSQL*, alguns exemplos são: CISCO, NTT Data, NOAA, Research in Motion, Google e Fujitsu (SHAW, 2014).

Hoje, o PostgreSQL encontra-se entre os melhores banco de dados do mercado (OBE; HSU, 2017). Nele, destacam-se algumas características: possui recursos exclusivos, ou seja, não são encontradas em nenhum outro banco de dados; é compatível com múltiplos sistemas operacionais; em todos os sistemas, instalá-lo é uma tarefa simples; é um *software* estável (SHAW, 2014).

Outra importante característica, é a sua escalonabilidade (SHAW, 2014). Em equivalência, está sua popularidade entre os serviços de armazenamento em nuvem, como por exemplo, *Heroku*, *Google Cloud SQL for PostgreSQL*, *Amazon Aurora for PostgreSQL* e *Microsoft Azure for PostgreSQL* (OBE; HSU, 2017). Além disso, todas as principais linguagens de programação possuem recursos para que um desenvolvedor trabalhe com o *PostgreSQL*, alguns exemplos são: *C*, *C++*, *Python*, *Java* e *PHP* (STONES; MATTHEW, 2005).

Como assegurado por Shaw (2014, p. 10, tradução nossa), “É um DBMS que resistiu ao teste do tempo e provou que tem o apoio e o poder para assumir qualquer tarefa de gerenciamento de dados que você possa imaginar.”. Entretanto, segundo Obe e Hsu (2017), existem alguns contextos em que o *PostgreSQL* não é recomendado ou apenas desnecessário. Por exemplo, se a pretensão de um desenvolvedor é colocar regras de segurança na própria aplicação, ele torna-se dispensável, afinal sua capacidade de segurança faz parte de sua robustez, logo, se ela é desnecessária, aconselha-se utilizar um banco de dados com uma necessidade de permissionamento inferior. Também, ele não é ideal para dispositivos com baixa capacidade de armazenamento pois, sua instalação mínima (sem extensões) é de 100MB.

Ainda segundo Obe e Hsu (2017), é comum combiná-lo com outras alternativas de banco de dados presentes no mercado. Caso exista a necessidade de um dispositivo consultar informações sem conexão de rede, recomenda-se a utilização do *SQLite*. Para soluções que envolvem cacheamento de dados, recomenda-se a utilização do *Redis* ou do *Memcached*.

5.1.4 *Redis*

O *Redis* é um banco de dados *NoSQL* - denominado não relacional. Na prática, os dados armazenados nele não precisam relacionar-se obrigatoriamente, ao contrário do que acontece com as tabelas nos bancos de dados *SQL* - denominados relacionais (CARLSON, 2013). Essencialmente, sua forma de armazenamento é através da utilização de chaves, onde cada chave é um novo bloco de informações. Internamente, ele trabalha da seguinte maneira: os dados ficam temporariamente armazenados em memória e, após um período de tempo ou

um número X de chaves alteradas, as modificações são gravadas no disco. Ambos critérios de gravação, tanto tempo, quanto número de chaves podem ser modificados (SEGUIN, [210-]).

Um dos contextos em que o *Redis* pode ser utilizado, é nas filas de tarefas. Onde, são armazenadas informações sobre uma tarefa - geralmente custosa - que será executada em um momento posterior (CARLSON, 2013). Outros tipos de informações comumente encontradas no *Redis*, são apresentadas por Macedo e Oliveira (2011, p. 2, tradução nossa), "[...] detalhes transacionais, dados históricos e logs de servidor.".

5.1.5 *SQLite*

O *SQLite* é um sistema de gerenciamento de banco de dados. Ele caracteriza-se por utilizar uma quantidade mínima de recursos computacionais, devido ao fato dele armazenar todos os dados e os metadados em um único arquivo. Inclusive, isso torna-o ideal para dispositivos com pouca capacidade de armazenamento e memória. Apesar de exigir poucos recursos, o *SQLite* é muito confiável e robusto, suportando a grande maioria das funcionalidades da linguagem *SQL* (KREIBICH, 2010). Todas essas características transformaram-no em uma ótima alternativa para armazenamento de dados em dispositivos móveis e, isso pode ser verificado em Allen e Owens (2010), onde, segundo os Autores, o *SQLite* passou a compor o pacote de bibliotecas que possuem suporte nativo do Android.

5.2 Rede Neural

A primeira etapa do desenvolvimento, destinou-se à organização do *dataset*, encontrado em Bostan e Klinger (2018). Para isso, foi criado um *script Python*. Ele foi executado para percorrer o arquivo original (*jsonl*), copiar as informações relevantes de cada objeto (texto e emoção) e enviá-las para um novo arquivo (*xlsx*). A Figura 14 mostra o trecho de código principal, onde são capturados os textos e as emoções.

Em princípio, as etapas posteriores à primeira também estavam sendo realizadas utilizando *scripts* feitos em *Python*. Contudo, o desenvolvimento da Rede Neural dependeu de diversas tarefas repetitivas e custosas. Por consequência, elas estavam deixando o desenvolvimento exageradamente lento. A partir disso, buscou-se uma alternativa mais adequada para lidar com Redes Neurais e, a escolhida foi o *Jupyter Notebook*.

Com o ambiente de desenvolvimento pronto, os dados presentes no novo arquivo *xlsx* foram carregados. Então, deu-se início ao processo de balanceamento, descrito na Seção 3.1.2. Ele foi realizado com o trecho de código apresentado na Figura 15. Onde, uma variável contadora é inicializada com o valor 0 e um laço de repetição é utilizado para percorrer todos os registros do *dataset*. Sempre que um registro com o rótulo de alegria for percorrido, a variável contadora é acrescida com 1. No momento em que ela chegar em 4000, todos os registros posteriores com o rótulo de alegria são deletados, utilizando seu índice.

```
ekman_emotions = ['joy', 'anger', 'sadness', 'disgust', 'fear', 'surprise']
with open("../data/unify-emotion-datasets/unified-dataset.jsonl", "r") as opened_file:
    i = 2
    for row in opened_file:
        obj = json.loads(row)
        source = obj["source"]
        text = obj["text"]
        emotions = obj["emotions"]
        for emotion in emotions:
            if emotions[emotion]:
                break
        if emotion in ekman_emotions and source in selected:
            worksheet.set_column('A:A', 50)
            worksheet.write(f"A{i}", text)
            worksheet.write(f"B{i}", emotion)
            i = i + 1
        print(i)
    workbook.close()
```

Figura 14 – Código que extrai o conteúdo de um arquivo *jsonl* e adiciona em um arquivo *xlsx*.

```
joy_count = 0
i = 0
delete = []
for text, emotion in data.values:
    if emotion == "joy":
        joy_count = joy_count + 1
        if joy_count >= 4000:
            delete.append(i)
    i = i + 1

data = data.drop(delete)
```

Figura 15 – Código que realiza o balanceamento do *dataset*.

A próxima etapa foi destinada para preparar os textos presentes no *dataset*. Vale ressaltar que, todas as técnicas utilizadas estão presentes na Seção 3.1.3. Os algoritmos utilizados para realizar o pré-processamento dos dados foram separados em um *script Python*, denominado *commons*. Essa separação foi necessária pois, a *API* precisa utilizá-los obrigatoriamente em etapas posteriores. A Figura 16 demonstra como está distribuída a estrutura do *backend*, dando ênfase na localização do *commons*.

O primeiro método encontrado no *commons* é o *prepare_texts*. Nele, ocorrem as seguintes etapas de pré-processamento: remoção de caracteres especiais, *stemming* e remoção das *stopwords*. O segundo método encontrado é o *create_sequences*, onde ocorre a vetorização dos textos e a padronização desses vetores. Ambos podem ser visualizados na Figura 17.

Na sequência, com os dados pré-processados, os mesmos são separados em 4 variáveis: *X_train*, *X_test*, *Y_train*, *Y_test*. 80% deles foram armazenados na *X_train* e 20% foram

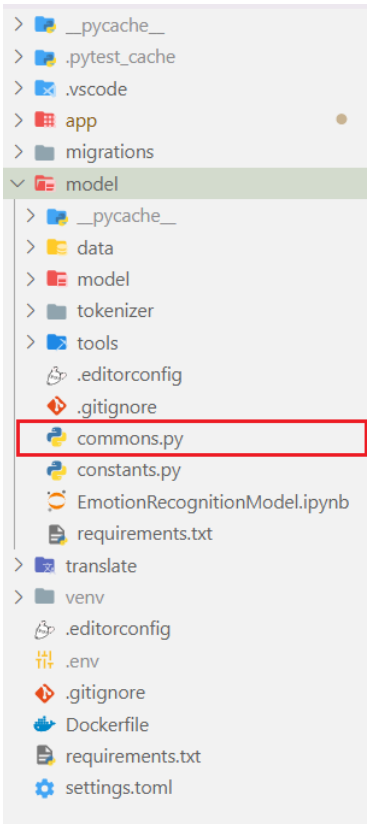


Figura 16 – Estrutura dos diretórios do *backend*.

```
def prepare_texts(texts):
    stop_words = set(stopwords.words('english'))
    ps = PorterStemmer()
    prepared_texts = []
    for text in texts:
        text = clear_text(text)
        word_list = text_to_word_sequence(text)
        no_stop_words = [ps.stem(w) for w in word_list if not w in stop_words]
        no_stop_words = " ".join(no_stop_words)
        prepared_texts.append(no_stop_words)
    return prepared_texts

def create_sequences(tokenizer, texts, maxlen):
    sequences = tokenizer.texts_to_sequences(texts)
    sequences = pad_sequences(sequences, maxlen=maxlen)
    return sequences
```

Figura 17 – Métodos presentes no arquivo *commons.py*

armazenados na *X_test*. Tanto *Y_train*, quanto *Y_test* armazenam os rótulos compatíveis com os dados armazenados nas outras variáveis, de acordo com seu prefixo. Essa divisão foi realizada de uma forma simples, utilizando um recurso fornecido pelo *sklearn*, onde só é necessário passar qual é o percentual desejado. Para um maior esclarecimento sobre o percentual informado, recomenda-se retornar à Seção 3.1.4.

```
def lstm():
    input_shape = (INPUT_LENGTH,)
    model_input = Input(
        shape=input_shape,
        name="input", dtype='int32')
    embedding = Embedding(
        NUM_WORDS,
        EMBED_DIM,
        input_length=INPUT_LENGTH,
        name="embedding")(model_input)
    lstm = LSTM(
        EMBED_DIM, dropout=0.2, recurrent_dropout=0.2,
        name="lstm")(embedding)
    model_output = Dense(
        len(unique),
        activation='softmax', name="softmax")(lstm)
    model = Model(inputs=model_input, outputs=model_output)
    model.compile(
        loss = 'categorical_crossentropy',
        optimizer='adam', metrics = ['accuracy'])
    return model
```

Figura 18 – Declaração da estrutura do modelo.

Após a separação dos dados, deu-se início à declaração da Rede Neural, baseando-se nos aspectos descritos na Seção 3.1.1. Apesar das características da Rede Neural já terem sido anteriormente expostas, ainda assim é necessário destacar alguns aspectos presentes em sua declaração que, inclusive, pode ser visualizada na Figura 18.

Primeiramente, é importante frisar que o modelo da Rede Neural foi construído utilizando o *Keras*. A primeira camada declarada é a *Input*, onde foi possível informar o tamanho das entradas do modelo (KERAS, 2020b). No trabalho, utilizou-se a constante *INPUT_LENGHT*, na qual está armazenado o valor máximo de palavras. Em seguida, é possível visualizar a camada *Embedding*, cuja é a primeira camada oculta da rede. Segundo Keras (2020a), ela espera alguns parâmetros necessários, como, por exemplo, o tamanho do vocabulário de palavras, o tamanho constante dos vetores de entrada e a quantidade de vetores densos que serão criados. Como pode ser observado no modelo, as saídas dos neurônios da camada *Embedding*, são as entradas da camada *LSTM*. Por fim, o modelo contém uma camada *Dense*, cuja é a camada de saída.

Na camada *Dense*, utilizou-se a função de ativação *softmax*. Ela funciona utilizando probabilidades, as quais estão relacionadas com cada provável rótulo passado para o modelo (BROWNLEE, 2020b). No contexto atual, o *softmax* aponta qual a probabilidade de cada uma das emoções atreladas a uma frase. Inclusive, vale ressaltar que, somente a maior probabilidade está sendo considerada neste trabalho.

Com o modelo devidamente declarado, através do próprio *Keras* foi possível treiná-lo. O método utilizado para realizar o treinamento pode ser visualizado na Figura 19. Todas as

```
model.fit(
    X_train,
    Y_train,
    batch_size=BATCH_SIZE,
    epochs=EPOCHS,
    validation_data=(X_test, Y_test))
```

Figura 19 – Código que realiza o treinamento do modelo.

variáveis com dados foram passadas como parâmetros, contudo somente *X_train* e *Y_train* são utilizadas no processo de aprendizado. *X_test* e *Y_test* são utilizadas para avaliar o modelo após cada época completada.

Por fim, após todas as épocas estarem concluídas, o modelo foi salvo em um arquivo *h5*. Dessa forma, tornou-se possível que a *API* carregasse-o e utilizasse-o para realizar previsões.

5.3 *API*

Ela foi desenvolvida utilizando o *framework Flask*. A Figura 20 mostra a saída do comando ***flask routes***. Nela, encontram-se todas as rotas disponíveis na *API*. Onde, a primeira coluna mostra o nome do método, a segunda coluna mostra o verbo *HTTP* (*Hypertext Transfer Protocol*) suportado por ele e a terceira coluna mostra o *endpoint*, juntamente com seus parâmetros obrigatórios, os quais são sinalizados pelo sinal de menor (<) e maior (>).

```
fetch_predictions           GET     /predictions/<task_id>
fetch_predictions_by_month  GET     /predictions/month/<task_id>
find_user_reddit            GET     /find_user/reddit/<username>
find_user_twitter           GET     /find_user/twitter/<username>
login                       GET     /login
new_user                    POST    /user/add
recognize_emotions          POST    /recognize_emotions
```

Figura 20 – Rotas disponíveis na *API*.

Esta Seção está dividida em cinco Subseções, cada uma delas dedica-se em explicar os recursos das principais rotas da *API*. Antes de iniciá-las, é necessário destacar a rota *login*, a rota *new_user* e outros aspectos importantes que compõem a estrutura da aplicação.

O *login* foi criado para funcionar como um se fosse um *ping*, onde o cliente faz uma requisição *GET* e, supondo que esteja autenticado, recebe uma resposta com o código 200. A *new_user* é utilizada para registrar novos usuários.

A *Basic Authentication* é uma das possíveis formas de autenticação que uma aplicação web pode implementar. Nela, tanto o identificador do usuário, quanto a senha são enviadas no cabeçalho da requisição, ambos codificados utilizando *Base64* [3] (MOZILLA DEVELOPERS

[3] Esquemas que permitem transformar dados brutos em representações textuais, as quais possibilitam que os

NETWORK, 2019a). Essa forma de autenticação foi escolhida para ser padrão em todas as rotas da *API*. A única rota desprotegida é o ***new_user***, afinal, nela não existem recursos que precisam ser resguardados.

Existe uma ressalva em relação ao *Basic Authentication*. Quando a aplicação é enviada para produção, é necessária a utilização do protocolo *HTTPS* (*Hyper Text Transfer Protocol Secure*) para que, de fato existam garantias de segurança (MOZILLA DEVELOPERS NETWORK, 2019a).

Na prática, o *Basic Authentication* foi implementado utilizando uma anotação sobre todas as rotas protegidas. Sempre que elas recebem uma nova requisição, aciona-se a função *verify_password*, a qual está exposta na Figura 21. Em resumo, essa função procura - através do e-mail - o usuário no banco de dados, se encontrar, confere se a senha informada é válida. Vale ressaltar que, a senha não é puramente armazenada no banco de dados, utiliza-se a codificação *sha256* [4].

```
@auth.verify_password
def verify_password(email, password):
    user = User.query.filter_by(email = email).first()
    if not user or not user.verify_password(password):
        return False
    g.user = user
    return True
```

Figura 21 – Função de autenticação.

5.3.1 Procurar usuário no Twitter

Denominado *find_user_twitter*, essa rota utiliza o *tweepy* para buscar o perfil de um usuário. Para tanto, é necessário passar como parâmetro um *username*. Se o perfil for encontrado, retorna-se o código 200. Senão, os códigos mais comuns são: (404) perfil não encontrado, (429) muitos pedidos foram realizados e (500) erro inesperado. Todas essas características estão presentes na Figura 22. Os códigos de insucesso (diferentes de 200) foram definidos através das possíveis exceções retornadas pelo *tweepy*.

5.3.2 Procurar usuário no Reddit

Denominado *find_user_reddit*, essa rota utiliza o *PRAW* para buscar o perfil de um usuário. Para tanto, em conformidade com a rota do Twitter, é necessário passar como parâmetro um *username*. Se o perfil for encontrado, retorna-se o código 200. Senão, os códigos mais comuns são: (404) perfil não encontrado e (500) erro inesperado. Todas essas características

dados sejam trafegados sem serem corrompidos (MOZILLA DEVELOPERS NETWORK, 2020).

[4] Algoritmo utilizado para criar codificações de 256 bits (SUNNATOVICH, 2019).

```python
@app.route("/find_user/twitter/<username>")
@auth.login_required
def find_user_twitter(username):
    """
    Localiza um usuário pelo seu username na api do twitter
    Respostas:
    - 200: Usuário encontrado.
    - 404: Usuário não encontrado.
    - 429: Muitos pedidos realizados.
    - 500: Um erro inesperado ocorreu no servidor.
    """
    try:
        twitter = Twitter()
        user = twitter.fetch_user(username)
        return TwitterUserSchema().dump(user)
    except TweepError as error:
        if isinstance(error.response, Response):
            status_code = error.response.status_code
        else:
            status_code = 500
        return response({
            "error": error.reason,
            "status_code": status_code
        }, status_code)
```

Figura 22 – Rota utilizada para localizar perfil de usuário no Twitter.

estão presentes na Figura 23. Os códigos de insucesso (diferentes de 200) foram definidos através das possíveis exceções retornadas pelo *PRAW*.

5.3.3 Reconhecer emoções

Denominada *recognize_emotions*, essa rota foi construída para esperar as informações apresentados na Figura 24, as quais estão formatadas utilizando o padrão *JSON* [5]. É possível verificar que, em um primeiro momento, existem dois atributos: *device_id* e *social_networks*. Ambos terão suas funções descritas no decorrer desta Seção.

O *social_networks* é um atributo que espera um objeto, cujo possui dois atributos: *twitter* e *reddit*. Ambos atributos também esperam objetos que, por sua vez, possuem de forma idêntica os atributos *username* e *record_count*. Tanto *username*, quanto *record_count* são utilizados para buscar os textos produzidos pelo usuário em questão, cada qual em sua respectiva rede social, respeitando a quantidade definida pelo *record_count*.

Após a busca ser realizada em ambas as redes sociais, o conteúdo textual é unificado e transformado em uma lista de dicionários. Cada dicionário possui as seguintes chaves: *original_text*, *created_at* e *source*. Onde, estão armazenados o texto encontrado, sua data de

[5] Forma de representação composta por chaves, as quais obrigatoriamente devem ser *strings*. Em suas chaves, é possível armazenar tipos primitivos, nulos, listas e objetos (MOZILLA DEVELOPERS NETWORK, 2019b)

```python
@app.route("/find_user/reddit/<username>")
@auth.login_required
def find_user_reddit(username):
    """
    Localiza um usuário pelo seu username na api do reddit
    Respostas:
    - 200: Usuário encontrado.
    - 404: Usuário não encontrado.
    - 500: Um erro inesperado ocorreu no servidor.
    """
    try:
        reddit = Reddit()
        user = reddit.fetch_user(username)
        return RedditUserSchema().dump(user)
    except MissingRequiredAttributeException as mrae_error:
        return response({
            "error": mrae_error.args,
            "status_code": 500
        }, 500)
    except NotFound as nf_error:
        return response({
            "error": nf_error.args,
            "status_code": nf_error.response.status_code
        }, nf_error.response.status_code)
```

Figura 23 – Rota utilizada para localizar perfil de usuário no Reddit.

```json
{
    "device_id":
    "f6ypikIMSquvU97K-SLN1k:APA91bHqJVHlqnmTn0kcpf6JjtCl-YFEO1T
    XV4Y3SZFC7yD0t4Qu73dyHSjahvYBDrBKD4nkbdFu9IM3lF9jrZPwkw3xGS
    pZ97oWecSQtXHscmxIa9kjc79dxwfxYEy_XlgwWeDNIYPE",
    "social_networks": {
        "twitter": {
            "username": "viniciusandd",
            "record_count": 5
        },
        "reddit": {
            "username": "viniciusanddd",
            "record_count": 5
        }
    }
}
```

Figura 24 – Dados esperados na rota que reconhece emoções.

criação e sua origem (Twitter ou Reddit), respectivamente. A função responsável pela unificação está apresentada na Figura 25.

Com o conteúdo textual unificado, tornou-se possível iniciar as previsões. Entretanto, prever as emoções mostrou ser um processo custoso, portanto definiu-se que ele seria alocado

```
def unify_textual_content(
        tweets: List[Status] = None,
        reddit_comments: List[Comment] = None
    ) -> List[dict]:
    texts = []
    if tweets:
        for tweet in tweets:
            texts.append({
                "original_text": tweet.text,
                "created_at": tweet.created_at.date(),
                "source": "twitter"})
    if reddit_comments:
        for reddit_comment in reddit_comments:
            texts.append({
                "original_text": reddit_comment.body,
                "created_at": reddit_comment.created_at.date(),
                "source": "reddit"})
    return texts
```

Figura 25 – Função que unifica o conteúdo textual.

em uma fila de tarefas *FIFO* (*First-in First-out*) [6]. Para um maior controle, dedicou-se uma tabela do banco de dados para armazenar todas a tarefas criadas. Ela pode ser visualizada na Figura 26.

```
class Task(db.Model):
    __tablename__ = "tasks"
    id = db.Column(db.String(36), primary_key=True)
    name = db.Column(db.String(128), index=True, nullable=False)
    description = db.Column(db.String(128))
    complete = db.Column(db.Boolean, default=False, nullable=False)
    enqueued_at = db.Column(db.DateTime, nullable=False)
    user_id = db.Column(db.Integer, db.ForeignKey('users.id'), nullable=False)
    notified = db.Column(db.Boolean, default=False, nullable=False)
```

Figura 26 – Tabela que armazena as tarefas.

Assim que uma tarefa é inserida no banco de dados e enviada para a fila, a *API* responderá o cliente de imediato, enviando-lhe informações sobre a mesma, incluindo seu identificador. Nesse momento, a requisição está finalizada. Contudo, ainda existirá uma tarefa na fila aguardando ser processada e, é nesse momento que o atributo *device_id* torna-se importante pois, após a tarefa alocada na fila ser concluída, é através desse atributo que o cliente será avisado.

[6] É uma fila que comporta-se da seguinte maneira: a remoção dos elementos ocorre de acordo com a ordem de chegada, logo o primeiro que chegar, é o primeiro a ser removido (FARIAS, 2009)

O *device_id* nada mais é que um *token* gerado pelo *Firebase*, no qual é utilizado pelo serviço *Cloud Messaging* (FIREBASE, 2020). Sua geração ocorre na primeira execução do aplicativo.

Adentrando nas tarefas, a primeira coisa a ser feita é carregar o modelo da Rede Neural, salvo anteriormente após o processo de treinamento, o qual está descrito na Seção 5.2. Para carregá-lo, utilizou-se um método disponibilizado pelo *Keras*, cujo pode ser visualizado na Figura 27.

```
# Load model
model = keras.models.load_model('model/model/model_lstm_20.h5')
model.summary()
```

Figura 27 – Código que realiza o carregamento do modelo.

Antes de submeter os textos para que o modelo realize as previsões, foi necessário traduzi-los. Afinal, o trabalho visa proporcionar que textos em português tenham suas emoções reconhecidas e, como pode ser observado na Seção 3.1.2, todos os textos presentes no *dataset* de treinamento estão em inglês. Logo, o modelo só é capaz de lidar com textos em inglês, porque assim foi treinado.

O *Amazon Translate* é um serviço na nuvem que proporciona que uma grande quantidade de dados seja traduzida (AMAZON WEB SERVICES, 2020). Por isso, decidiu-se utilizá-lo na tradução dos textos em português para o inglês, antes de submetê-los ao modelo.

```
predictions = []
for tc in textual_content:
    translated_text = translate.translate_with_amazon(tc['original_text'])
    prepared_text = prepare_texts([translated_text])
    sequence = create_sequences(tokenizer, prepared_text, INPUT_LENGTH)
    predict = model.predict(sequence)
    predominant_value = predict.argmax(axis=-1)[0]
    emotion = sorted(labels)[predominant_value]

    predictions.append(Prediction(
        portuguese_text=tc['original_text'],
        english_text=translated_text,
        emotion=pt_br_labels[emotion],
        task_id=job.get_id(),
        source=tc['source'],
        text_created_at=tc['created_at'],
        text_creation_month=tc['created_at'].month,
        text_creation_year=tc['created_at'].year
    ))
```

Figura 28 – Código que realiza as previsões através dos textos.

Através da Figura 28, é possível visualizar as seguintes etapas da tarefa: tradução dos textos utilizando o *Amazon Translate*; preparação dos textos utilizando os métodos presentes

no *commons*, expostos na Figura 17; previsão utilizando o modelo; captura da emoção predominante. Em seguida, as previsões são armazenadas em uma lista que será persistida no banco de dados.

Por fim, utilizando o *Cloud Messaging*, notifica-se o cliente através do seu *device_id*.

5.3.4 Buscar previsões

Denominada *fetch_predictions*, essa rota foi construída para fornecer os resultados das previsões realizadas em uma tarefa. Para usá-la, obrigatoriamente deve ser fornecido o parâmetro *task_id*, obtido como resposta na rota *recognize_emotions*. Através desse parâmetro, realiza-se uma consulta no banco de dados e todas as previsões encontradas são retornadas.

5.3.5 Buscar previsões por mês

Denominada *fetch_predictions_by_month*, essa rota foi construída para fornecer os resultados das previsões realizadas em uma tarefa. Em conformidade com a rota *fetch_predictions*, para usá-la, obrigatoriamente deve ser fornecido o parâmetro *task_id*, obtido como resposta na rota *recognize_emotions*.

Entretanto, essa rota vai além de uma consulta simples às previsões do banco de dados. Primeiro, realiza-se uma consulta que agrupa por mês e ano as previsões. Após, realiza-se uma segunda consulta que agrupa as previsões por emoção.

Por fim, retorna-se uma lista de objetos que contém dois atributos, são eles: *date* e *emotions*. O primeiro atributo armazena uma concatenação entre o mês e o ano daquelas previsões. O segundo atributo contem outros seis atributos, cada um possui deles armazena um inteiro que aponta a quantidade de vezes que cada emoção apareceu naquele período.

5.4 Prototipação

Devido à importância de se ter uma *UI* amigável e fácil de usar, decidiu-se que, antecedendo a criação do aplicativo, um protótipo seria construído. Para isso, buscou-se uma ferramenta adequada.

O *Figma* é uma plataforma que proporciona inúmeros recursos para desenhar as telas de uma aplicação, sendo possível executá-lo no próprio navegador (GUERRA, 2016). Foi necessário destacá-lo nesta Seção pois, ele possui uma comunidade que costuma criar projetos abertos e disponibilizá-los para terceiros utilizarem. Foi um desses projetos o escolhido para ser o *template* do aplicativo, ele pode ser localizado em: <https://www.figma.com/community/file/833515051385038928>.

O *template* escolhido possui uma vasta quantidade de componentes e exemplos de telas que, além de amigáveis, possuem uma ótima usabilidade. Contudo, o principal critério da escolha foi sua temática relacionada às pessoas, as quais são as propagadores das emoções. Essa decisão baseou-se no que é dito por Motta (2018), segundo o Autor, é importante levar em

consideração os aspectos que permeiam a aplicação, possibilitando que ela conecte-se com as pessoas e, além disso, um visual moderno sempre prenderá mais a atenção dos usuários.

Nesta Seção, apresenta-se algumas das telas que foram prototipadas. Na Figura 29, mostra-se os protótipos criados para a tela de login, a tela do menu principal e duas das telas de apresentação do sistema.

Figura 29 – Protótipos das telas do aplicativo.

5.5 Aplicativo

O aplicativo foi desenvolvido utilizando o *Kotlin*. Como anteriormente mencionado na Seção 5.1.2, essa tecnologia possibilita desenvolver aplicações para múltiplas plataformas. Em princípio, definiu-se que seria desenvolvida uma versão para Android e iOS, porém devido as diferenças no ambiente de desenvolvimento entre eles, optou-se por priorizar apenas um. Com isso, a versão apresentada neste trabalho foi desenvolvida dando ênfase no Android que, segundo (STATCOUNTER, 2020), hoje, é o sistema operacional mais presente no mercado de dispositivos móveis.

O *retrofit* é um cliente *HTTP* construído em *Java*, o qual permite transformar as rotas de uma *API* em uma *interface* [7] (SQUARE, 2020). Todas as requisições realizadas para a *API* do sistema utilizam-no. É possível visualizar a declaração das rotas *find_user_twitter* (Seção 5.3.1) e *find_user_reddit* (5.3.2) na Figura 30.

A Figura 31 mostra um dos principais fluxos do aplicativo, onde são localizados os perfis dos usuários. Para construí-lo, utilizou-se um *singleton* que, como assegurado por Lima

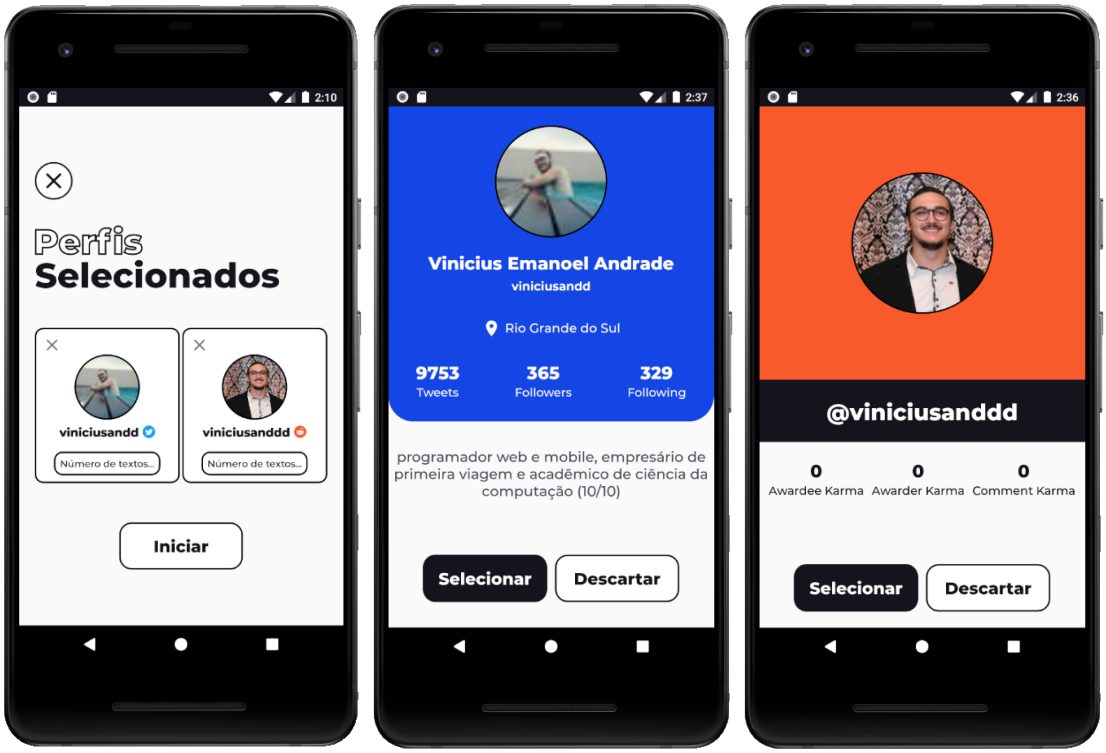
[7] Recurso que permite declarar os métodos que devem ser implementados por algumas classes da aplicação (PADUA, 2017)

```
interface SocialNetworkProfile {
    @GET( value: "/find_user/twitter/{username}")
    fun findOnTwitter(@Path( value: "username") username: String) : Call<TwitterProfile>

    @GET( value: "/find_user/reddit/{username}")
    fun findOnReddit(@Path( value: "username") username: String) : Call<RedditProfile>
}
```

Figura 30 – *Interface* com a declaração das rotas que buscam os perfis.

(2019), é um padrão de projeto, cujo permite que apenas uma instância de um objeto seja criada em toda a aplicação.



(a) Perfis selecionados. (b) Perfil do Twitter. (c) Perfil do Reddit.

Figura 31 – Aplicativo: Fluxo para localizar perfis nas redes sociais.

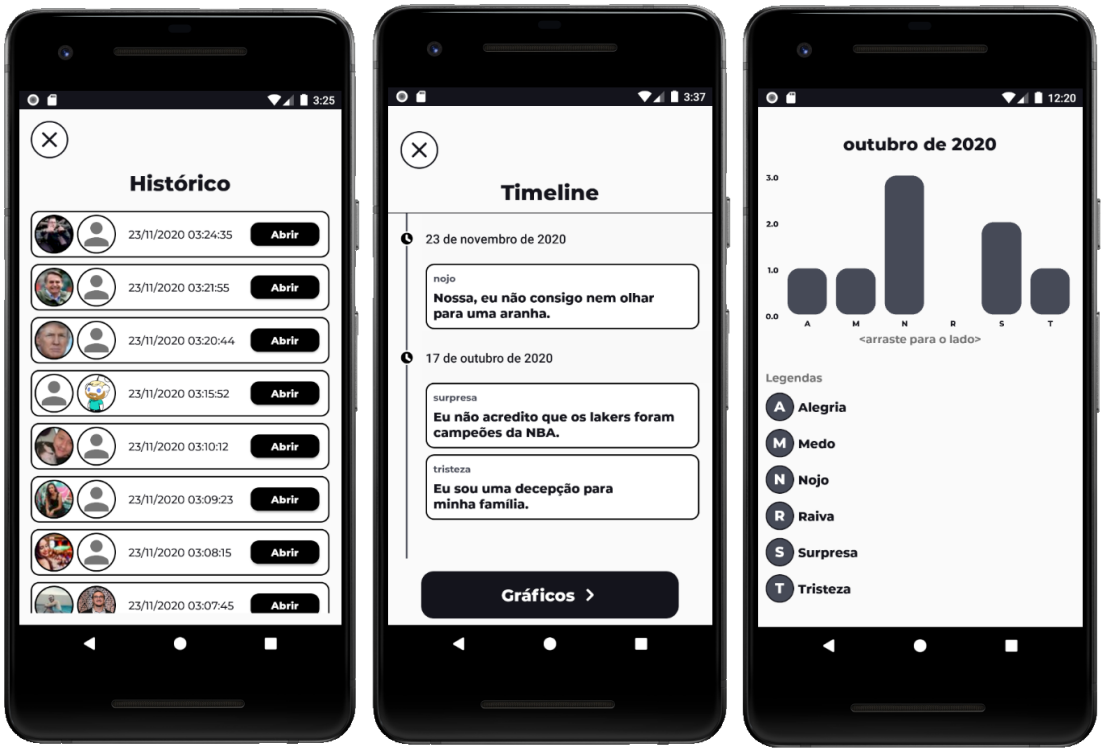
A utilização do *singleton* deu-se pela necessidade de centralizar os perfis selecionados e, além disso, esse objeto precisava ser acessado por mais de uma tela do aplicativo. Logo, considerou-se que essa seria uma boa alternativa para solucionar ambos os problemas.

O *singleton* em questão possui dois atributos (*twitterProfile* e *redditProfile*), cada um deles destina-se em armazenar um objeto que contém as informações de cada um dos perfis selecionados. Ainda na Figura 31, o armazenamento dos atributos do *singleton* ocorre quando o botão “Selecionar” é pressionado. Como pode ser verificado, o botão está presente em (b) e (c). Em (a), quando o “x” de uma das caixas de seleção for pressionado, o objeto que contém as informações daquele perfil é removido do atributo do *singleton* e, em seguida, a caixa de seleção é esvaziada, sinalizando em tela que nenhum perfil daquela rede social está selecionado.

Supondo que o botão “Iniciar”, em (a), seja pressionado e, no mínimo, um dos atributos do *singleton* não esteja vazio, é feita uma requisição para a rota *recognize_emotions* (Seção 5.3.3) e, em seguida, ambos os atributos do *singleton* são esvaziados pois, encerra-se o fluxo de seleção dos perfis e cria-se uma nova tarefa.

Todas as tarefas criadas pelo usuário são armazenadas em um banco de dados no seu próprio dispositivo. Contudo, armazena-se somente o mínimo necessário para que o usuário tenha um histórico de tarefas organizado. Ou seja, no banco de dados estão os links que apontam para as imagens dos perfis selecionados, a data de criação da tarefa e o identificador retornado pela *recognize_emotions* (Seção 5.3.3). O histórico de tarefas pode ser visualizado na Figura 32-(a).

Para tanto, o Android possui uma classe que auxilia no gerenciamento do banco de dados *SQLite* (Seção 5.1.5), isentando os desenvolvedores de inúmeras responsabilidades de versionamento, possibilitando que eles preocupem-se somente em criar as tabelas e métodos necessários para persistência dos dados (Android Developers, 2020). Essa classe foi utilizada no aplicativo, com ela foi possível criar toda a estrutura do banco de dados do mesmo, inclusive a tabela que armazena as tarefas citadas no parágrafo anterior.



(a) Histórico de tarefas. (b) Timeline. (c) Gráficos.

Figura 32 – Aplicativo: Fluxo de visualização dos resultados.

Ainda na Figura 32-(a), sempre que o botão “Abrir” for pressionado, envia-se uma requisição *GET* para a rota *fetch_predictions* (Seção 5.3.4), onde é passado o identificador da tarefa que foi armazenado no banco de dados. Através dos resultados retornados por ela, renderiza-se a *Timeline*, a qual pode ser visualizada na Figura 32-(b).

A Figura 32-(b) mostra o botão “Gráficos” que, por sua vez, quando for pressionado pelo usuário, envia-se uma requisição *GET* para a rota *fetch_predictions_by_month* (Seção 5.3.5), onde também é enviado o identificador da tarefa. Através dos resultados retornados por ela, renderizam-se os gráficos mensais. Um exemplo deles pode ser visualizado na Figura 32-(c).

Esta Seção limitou-se em descrever os dois principais fluxos presentes no aplicativo, juntamente com os principais recursos que proporcionam suas execuções. Mais detalhes sobre o aplicativo estão presentes no Apêndice A, onde encontra-se um manual de uso do sistema, o qual explica como utilizar os recursos do aplicativo e mostra todas as telas existentes.

5.6 Bancos de dados

Nesta Seção estão dois diagramas Entidade-Relacionamento, os quais apresentam a distribuição das tabelas presentes no banco de dados *Postgres* e no banco de dados *SQLite*, respectivamente.

O *Postgres* é o principal banco de dados do sistema, responsável pelo armazenamento das previsões, das tarefas e dos usuários. A única parte do sistema que acessa-o diretamente é a *API* (Seção 5.3). Sua estrutura está exposta na Figura 33. Onde, é possível visualizar a tabela *alembic_version*. Ela é utilizada para controlar a versão atual das suas migrações.

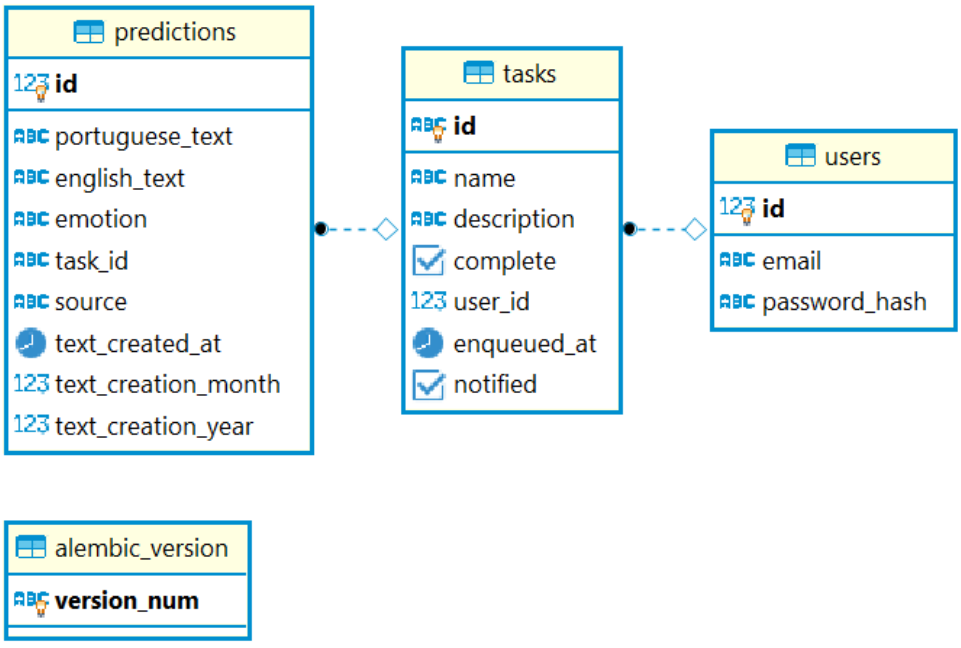


Figura 33 – Diagrama Entidade-Relacionamento do *Postgres*.

O *SQLite* é o banco de dados que fica dentro dos dispositivos dos usuários. Sua criação ocorre na primeira execução do aplicativo. Nele, armazena-se somente algumas informações

das tarefas criadas pelo próprio usuário, as quais são utilizadas para compor o histórico, onde é possível acessar as previsões já realizadas. A sua estrutura pode ser visualizada na Figura 34. Onde, é possível visualizar a tabela *android_metadata*. Ela é utilizada para armazenar informações sobre a estrutura do banco de dados.

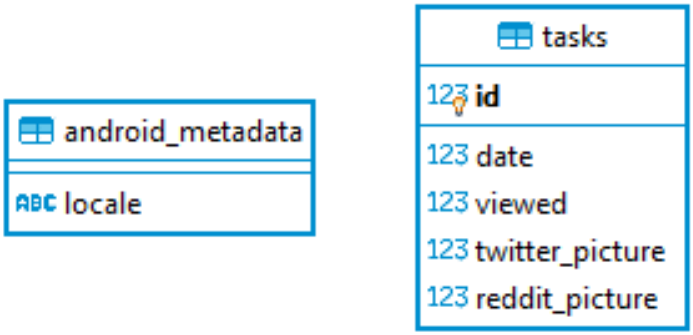


Figura 34 – Diagrama Entidade-Relacionamento do *SQLite*.

A única função do *Redis* é armazenar a fila de tarefas. Como pode ser verificado na Figura 33, a tabela *tasks* no banco de dados *Postgres* possui o atributo *enqueued_at*. Esse atributo faz menção à quando uma determinada tarefa foi inserida na fila (contida no *Redis*).

Por fim, é importante ressaltar que, as únicas informações armazenadas referentes aos perfis localizados nas redes sociais, são as duas *URLs* que apontam para as fotos deles. As quais não são armazenadas no banco de dados do lado do servidor (*Postgres*), somente no *SQLite*. Tomou-se o cuidado de manter no lado do servidor somente o conteúdo textual obtido, sem relação nenhuma com o perfil que os criou.

6 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Esta Seção está dividida em três partes, onde serão discutidos e analisados os resultados obtidos. Será dado ênfase no modelo da Rede Neural. Afinal, detectar as emoções é o primeiro pilar deste trabalho. Para avaliá-la, inicialmente foram utilizados textos curtos criados exclusivamente para essa tarefa. Em seguida, buscou-se textos do Twitter e do Reddit, os quais foram criados por usuários aleatórios. Todos eles possuem contextos diferentes, pois dessa forma é possível mostrar ao leitor o quão ampla é a capacidade do modelo. Por fim, é feita uma análise sobre a acurácia obtida no processo de aprendizagem.

6.1 Previsões em textos curtos

Como citado anteriormente na Seção 5.3.3, o modelo foi treinado com exemplos em inglês, tornando necessário que os textos em português sejam previamente traduzidos. Nesta Seção, todos os textos foram apresentados em português, cujo objetivo é melhorar a compreensão dos resultados. Mas vale ressaltar que, antes de submetê-los ao modelo, todos os textos foram traduzidos utilizando o *Amazon Translate*.

A Tabela 1 mostra cinco textos em que a emoção detectada foi a tristeza. Destacam-se alguns resultados. Por exemplo, no modelo, frases que possuem “câncer” estão propensas à tristeza, independente do sujeito da frase. Em “Minha avó está com câncer” detecta-se tristeza. Substituindo “Minha avó” por “Meu inimigo”, também detecta-se tristeza.

Em conformidade está “Meu cachorro morreu”, a qual também é constatada a tristeza. Quando submete-se “Um cervo morreu”, ainda assim constata-se tristeza. Contudo, ao substituir “Um cervo” por “Uma cobra”, a tristeza deixa de ser predominante e quem assume é o medo, afinal a palavra “cobra” torna propensa a predominância dessa emoção.

Tabela 1 – Previsões em textos curtos que detectaram tristeza.

Texto	**Previsão**
Minha avó está com câncer	Tristeza
Meu cachorro morreu	Tristeza
Sou uma decepção para minha família	Tristeza
Eu tenho péssimas lembranças do passado	Tristeza
Não tenho amigos, sou uma pessoa sozinha	Tristeza

A Tabela 2 mostra cinco textos em que a emoção detectada foi a raiva. É possível notar que ela pode ser predominante em contextos distintos. Entretanto, destacou-se um fator interessante. Como anteriormente citado, na Seção 2.1, uma das principais características da raiva, segundo (EKMAN, 2011), é o fato de um sujeito estar impossibilitado de fazer alguma coisa ou realizar algo.

O modelo conseguiu identificar essa característica da emoção e, é possível visualizar isso quando detecta-se raiva nos seguintes exemplos: “Tem um carro impedindo que eu saia do estacionamento.”, “Estou no trânsito. Vou me atrasar.” e “Não vai ter jogo de futebol”. Além disso, é possível ver que o modelo muda suas previsões de acordo com o motivo inviabilizador na frase. Em “Eu não posso ir na festa porque minha mãe não deixa” detecta-se raiva. Em “Eu não posso ir na festa porque estou me sentindo mal” detecta-se tristeza.

Tabela 2 – Previsões em textos curtos que detectaram raiva.

Texto	**Previsão**
Odeio meus colegas de trabalho. Eles não calam a boca.	Raiva
Minha mãe não deixa eu sair com meus amigos	Raiva
Estou no trânsito. Vou me atrasar.	Raiva
Todas as falas do presidente me irritam	Raiva
Não vai ter jogo de futebol	Raiva

A Tabela 3 mostra cinco textos em que a emoção detectada foi o medo. Em conformidade com o que foi encontrado em (EKMAN, 2011) e descrito anteriormente na Seção 2.1, não existe um limite para o medo, logo, ele pode ser disparado com qualquer objeto existente. Contudo, existem objetos e situações que disparam gatilhos de medo com maior frequência.

Por exemplo, “cobras” dificilmente estarão atreladas em uma situação diferente do medo. Em contrapartida, existem outros objetos que sozinhos não causam medo, mas que podem ser inseridos em um contexto ameaçador. Exemplificando, quando submete-se ao modelo a frase “O Porsche ultrapassou a Ferrari”, detecta-se alegria. Quando contextualiza-se que a ultrapassagem ocorreu de forma “perigosa”, detecta-se o medo. Outro exemplo são as “aranhas” que, quando submetidas ao modelo estão propensas ao nojo. Contudo, quando o interlocutor caracteriza-as como “assustadoras”, constata-se que o medo tornou-se predominante.

Tabela 3 – Previsões em textos curtos que detectaram medo.

Texto	**Previsão**
O Porsche fez uma ultrapassagem perigosa.	Medo
Não vou sair sozinho de noite.	Medo
As aranhas são assustadoras.	Medo
Eu acho que tem cobras aqui	Medo
Estou aguardando o diagnóstico do médico.	Medo

A Tabela 4 mostra cinco textos em que a emoção detectada foi a surpresa. Essa emoção ocorre em momentos que o interlocutor vivencia uma situação inesperada. Referente ao comportamento do modelo, existem alguns textos que quando presentes em frases tornam

propensa a previsão da surpresa, são eles: “Não acredito”, “Isso é sério” e “Não pode ser verdade”.

Tabela 4 – Previsões em textos curtos que detectaram surpresa.

Texto	**Previsão**
Não acredito que os Lakers ganharam a NBA	Surpresa
Recebi um aumento esse mês	Surpresa
Eu fui surpreendido com a notícia do desaparecimento	Surpresa
Não pode ser verdade que eu fui roubado.	Surpresa
É sério que não tinha ninguém em casa?	Surpresa

A Tabela 5 mostra cinco textos em que a emoção detectada foi o nojo. Existem alguns objetos que favorecem a detecção dessa emoção, como, por exemplo “vomito”, “infecção” e “mal cheiro”. Inclusive, referente à todos os exemplos citados, quando eles são submetidos sem nenhum outro texto em conjunto, detecta-se nojo. Entretanto, existem algumas palavras que precisam estar contextualizadas para que o nojo seja detectado. Por exemplo, ao ser submetido “lixo”, detecta-se surpresa. Mas, ao ser submetido “Eu vi uma pessoa comendo lixo”, detecta-se nojo.

Tabela 5 – Previsões em textos curtos que detectaram nojo.

Texto	**Previsão**
Tem uma aranha na parede da cozinha	Nojo
Eu odeio o cheiro do café	Nojo
O ferimento está infeccionado	Nojo
Tem vomito no tapete da sala	Nojo
Tem uma barata na minha comida	Nojo

A Tabela 6 mostra cinco resultados em que a emoção detectada foi a alegria. Essa emoção está propensa em ser detectada quando as frases contém palavras positivas como, por exemplo, “amigos”, “feliz” e “amor”.

Tabela 6 – Previsões em textos curtos que detectaram alegria.

Texto	**Previsão**
Passei o dia com meus amigos	Alegria
Estou feliz	Alegria
Eu amo você!	Alegria
Você é maravilhoso	Alegria
Eu adoro o verão	Alegria

A Tabela 7 mostra cinco resultados ambíguos. Ou seja, as previsões não foram as esperadas. Contudo, os resultados apresentados são compreensíveis. Em “Estou me sentindo

péssimo. Eu acho que todos me odeiam." esperava-se tristeza. Em "Eu odeio você!" esperava-se raiva. Em "Muitas pessoas estão morrendo durante a pandemia" esperava-se medo. Em "Descobri que vou ser pai" esperava-se surpresa. Em "Tem uma mosca na minha comida" esperava-se nojo.

Tabela 7 – Previsões em textos curtos que geraram resultados ambíguos.

Texto	**Previsão**
Estou me sentindo péssimo. Eu acho que todos me odeiam.	Raiva
Eu odeio você!	Nojo
Muitas pessoas estão morrendo durante a pandemia	Raiva
Descobri que vou ser pai	Alegria
Tem uma mosca na minha comida	Raiva

6.2 Previsões em textos do Twitter

Em conformidade com a Seção anterior, os textos foram mantidos em português para uma melhor apresentação dos resultados. Contudo, antes de serem submetidos ao modelo, todos os textos foram traduzidos utilizando o *Amazon Translate*.

Para avaliar o comportamento do modelo perante aos textos presentes no Twitter, buscou-se cinco *tweets* criados por usuários aleatórios. Nenhum critério de busca foi utilizado, a única ressalva era encontrar frases com contextos distintos. As frases e as emoções previstas pelo modelo podem ser visualizadas na Tabela 8.

Os textos obtidos no Twitter (*tweets*) são, em sua grande maioria, curtos, informais, sem contexto, com abreviações e *emojis*. É preciso levar em consideração o limite de 280 caracteres que, consequentemente exige que o autor do *tweet* seja breve. Os *tweets* foram os textos que o modelo teve maior dificuldade para prever as emoções.

6.3 Previsões em textos do Reddit

Em conformidade com as Seções anteriores, os textos foram mantidos em português para uma melhor apresentação dos resultados. Contudo, antes de serem submetidos ao modelo, todos os textos foram traduzidos utilizando o *Amazon Translate*.

Para avaliar o comportamento do modelo perante aos textos presentes no Reddit, buscou-se cinco comentários criados por usuários aleatórios. Nenhum critério de busca foi utilizado, a única ressalva era encontrar frases com contextos distintos. As frases e as emoções previstas pelo modelo podem ser visualizadas na Tabela 9.

Os textos obtidos no Reddit são mais heterogêneos que os do Twitter. Ou seja, existem vários comentários curtos e sem contexto, mas também existem longos comentários onde o contexto é apresentado de forma clara durante a construção do texto. Os comentários do Reddit obtiveram um melhor desempenho nas previsões, quando comparados aos *tweets*.

Tabela 8 – Previsões em textos retirados do Twitter.

Texto	**Previsão**
eu quase não comento sobre esses assuntos aqui pq esse assunto me deixa muito mal por já ter sido vítima de pedófilo mas isso aqui é um absurdo DEMAIS da uma sensação de impotência da uma sensação ruim demais	Medo
Sim..Falei da beleza dela pq isso é algo que contribui mais ainda pro ciúme exagerado desses loucos, infelizmente. Ela era Cabelereira aqui na minha cidade. Estava no estacionamento chegando pra trabalhar quando o cara veio atrás dela e numa discussão a esfaqueou. Que tristeza!	Tristeza
Pessoal, por favor, matem as aranhas... Eu tenho muito medo delas, e sei que isso não é motivo suficiente para sair matando os outros, mas se possível, matem, sem essa de empurra para fora de casa com papel ou vassoura.	Medo
Cara.. se você for parar pra vê as publicações de páginas do Instagram e tudo mais.. você entra em depressão. É uma coisa que te empurra. Te força a ficar triste.. porque é só frase triste, texto triste... Sei lá. Quem me dera alegria fosse espalhado igual...	Tristeza
Tua história, @dale10 , eu vou contar para os meus filhos. Eu vou mostrar pra eles como se comporta um colorado de verdade. Um vou contar que eu te vi fardar mais de 500 vezes. Fazer quase 100 gols e levantar tantas taças. Mas enquanto eu não conto, eu te agradeço. Gigante.	Alegria

6.4 Acurácia do modelo

Como anteriormente citado, o modelo foi treinado utilizando a aprendizagem supervisionada. Todos os dados utilizados no processo, foram retirados do *dataset* descrito na Seção 3.1.2. O treinamento foi realizado utilizando 80% do *dataset*, totalizando 14.961 registros.

Após sua conclusão, o modelo foi avaliado utilizando 20% do *dataset*, totalizando 3.741 registros. A avaliação do modelo resultou em uma acurácia de 57.77%. Para uma melhor representação dos resultados obtidos, na avaliação do modelo, criou-se uma Matriz de Confusão. Ela pode ser visualizada na Figura 35.

Através dela é possível visualizar que o modelo acertou um total de 2.161 previsões, as quais compõem a diagonal principal da matriz. Também é possível calcular qual foi o percentual de acerto que o modelo teve em cada uma das emoções:

- **Alegria**: 63.41%
- **Surpresa**: 58.40%
- **Tristeza**: 58.20%
- **Medo**: 57.26%
- **Nojo**: 54.67%
- **Raiva**: 47.91%

Além desses percentuais, pode-se observar alguns comportamentos do modelo. Por

Tabela 9 – Previsões em textos retirados do Reddit.

Texto	**Previsão**
Gostava muito do The Noite em 2014, não vou mentir, mas com o tempo fui percebendo o quão babaca era o Danilo e o Ultrage, então parei de assistir e peguei muito desgosto do Danilo Gentili, para mim ele era ótimo no CQC e no início do The Noite, mas foi colocando as asinhas de fora e se mostrou um belíssimo babaca que paga de centro, mas todo mundo vê que é um puta direitista que tem ódio da esquerda.	Nojo
Isso é uma das coisas que me irrita, hoje em dia todo mundo fica filmando ao invés de fazer algo, po se uma pessoa ta filmando tu podes ir ali e impedir Edit: não falei para impedir fisicamente, mas chegar ali falando para pararem	Raiva
Não conhecia o caso. Meu deus, isso acabou com meu finalzinho de semana. Na moral, que elas passem o resto dos dias delas na cadeia.	Tristeza
E por que você não tenta mudar de empresa? Tem muita empresa diferente por aí, muito provável que você encontre algo que te deixe mais feliz. Você pode, inclusive, tentar o próprio Nubank! Muitas vezes não é a área que você trabalha e sim o tipo de problema que você resolve. Por exemplo, você pode trabalhar nessa empresa ai e ter que resolver problemas mais complexos de arquitetura de software. Pensar em como melhorar os sistemas, por exemplo, ou pensar se é possível usar um sistema de mensageria para ligar 2 sistemas. Eu trabalho em uma fintech também e , por várias vezes, o mercado financeiro pode ser bem chato, mas o tipo de problema que eu resolvo no dia a dia é bem legal. Ver pessoas sendo impactadas positivamente pelo o que você faz é uma das minhas maiores alegrias.	Alegria
Meu pai sempre nos disse que Diego Maradona era a única esperança que este país tinha nos anos 80. Nossa economia não estava bem, a democracia estava em perigo todos os dias, mas nós o tínhamos. Perdi meu pai em 2018. Com o passar do tempo, percebi que algumas coisas que compartilhei com ele foram desaparecendo aos poucos: lugares que mudaram desde então, políticos que se aposentaram, jogadores em nosso onze inicial que ele nunca viu. Hoje me sinto um pouco mais perdido.	Tristeza

exemplo, muitas das previsões que deveriam detectar a raiva, detectaram medo. O que transparece o motivo do baixo percentual dessa emoção. Em conformidade estão as previsões do medo que, por sua vez, quando errôneas, tendem à raiva. O medo também foi predominante nas previsões erradas da tristeza e da surpresa. Por fim, a surpresa preponderou dentre as previsões erradas da alegria e do nojo.

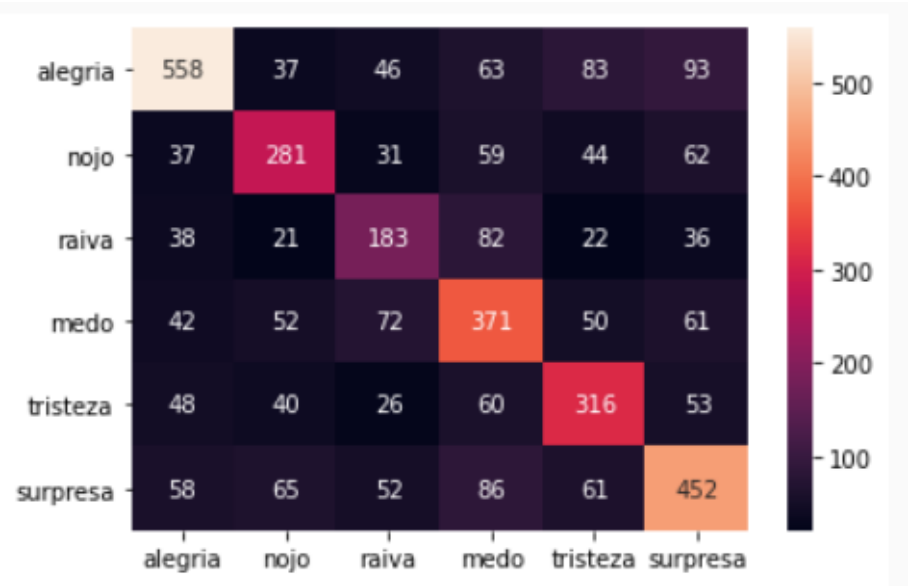


Figura 35 – Matriz de confusão.

7 CONSIDERAÇÕES FINAIS

O desenvolvimento deste trabalho partiu da seguinte hipótese: é possível encontrar indícios de transtornos mentais no conteúdo textual produzido por um indivíduo? Afinal, as emoções propagadas por ele são um reflexo do seu estado de saúde mental. Além disso, existem recursos suficientes para que seja possível encontrar emoções em textos e transformar os resultados em representações de fácil interpretação.

Para validá-la, a primeira etapa do trabalho foi dedicada para pesquisas sobre as emoções. Foi essencial compreender o que elas são e como funcionam no corpo humano. Além disso, foi necessário descobrir qual era a relação das emoções com os transtornos mentais e como as emoções são detectadas em textos.

A partir disso, iniciaram-se as buscas por técnicas computacionais que proporcionassem que o reconhecimento das emoções em textos fosse realizado. Nelas, encontrou-se o PLN, as Redes Neurais e a Aprendizagem Supervisionada.

Em seguida, novas pesquisas foram iniciadas, agora em busca de características dos transtornos mentais, as quais pudessem ser computacionalmente representadas. No momento em que elas foram encontradas, as mesmas serviram de base para a construção de duas representações: uma *timeline* e um gráfico. Ambos, com base na literatura, mostraram ser capazes de apontar indícios de transtornos mentais.

A matéria-prima desse trabalho são as emoções previstas pela Rede Neural. Com base nas avaliações realizadas, o modelo da Rede Neural provou ser eficiente prevendo emoções em diversos contextos. Contudo, a acurácia atingida (57.77%) precisa ser melhorada.

Para isso, como trabalhos futuros, espera-se criar um *dataset* com um número maior de registros. Inclusive, quando comparada à outras abordagens, o Reconhecimento de Emoções ainda possui uma escassez de dados. Além disso, ao invés de descartar uma parte considerável do *dataset* para realizar o balanceamento, almeja-se encontrar alguma técnica que possibilite realizá-lo sem perder tantos exemplos que, devido ao contexto de escassez, deveriam ser melhor aproveitados. Também, imagina-se uma experimentação de outras arquiteturas de redes neurais. Como por exemplo, a proposta por Sosa (2017). Cuja, é constituída por camadas LSTMs e RNCs. Onde, segundo o Autor, a capacidade de localizar elementos das RNCs é complementada pela capacidade de memorização das LSTMs.

Existem outros pontos que podem ser evoluídos neste trabalho. Por exemplo, utilizar um número maior de rede sociais, ampliando a probabilidade de um indivíduo ser localizado e também a quantidade de textos que podem ser analisados. Além disso, novas técnicas de PLN podem ser adotadas ou, simplesmente novos *regex* podem ser acoplados na etapa de pré-processamento, purificando ainda mais o conteúdo textual e, consequentemente, melhorando os resultados das previsões. Acredita-se também que, a construção de uma nova representação facilitaria ainda mais a encontrar os indícios, cuja seria semelhante ao gráfico

mensal, contudo, ela agruparia as emoções semanalmente.

Outra proposta futura, é realizar parcerias com outras áreas acadêmicas. Afinal, adentrar em áreas de pesquisa diferentes das habituais sempre é um desafio e, inclusive, essa foi uma das etapas mais difíceis deste trabalho. O conhecimento de acadêmicos ou profissionais da psicologia seria de grande valia para realizar avanços nesta proposta, principalmente quando trata-se da construção e validação dos métodos utilizados para encontrar os indícios de transtornos mentais.

REFERÊNCIAS

ALLEN, G.; OWENS, M. **The Definitive Guide to SQLite (Expert's Voice in Open Source)**. 2. ed. [S.l.]: Apress, 2010. ISBN 9781430232254. Citado na página 38.

ALM, E. C. O. **Affect in text and speech**. [S.l.]: University of Illinois at Urbana-Champaign, 2008. Citado na página 15.

AMAN, S.; SZPAKOWICZ, S. Identifying expressions of emotion in text. p. 196–205, 09 2007. Citado na página 7.

AMAZON WEB SERVICES. **Amazon Translate: Tradução automática fluente e precisa**. 2020. Disponível em: <https://aws.amazon.com/pt/translate/>. Acesso em: 20 de agosto de 2020. Citado na página 47.

AMERICAN PSYCHIATRIC ASSOCIATION. **Manual Diagnóstico E Estatístico De Transtornos Mentais - 5.ed.** [S.l.]: Artmed Editora, 2014. ISBN 9788582710890. Citado 6 vezes nas páginas 1, 6, 19, 20, 21 e 22.

Android Developers. **SQLiteOpenHelper**. 2020. Disponível em: <https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper>. Acesso em: 20 de setembro de 2020. Citado na página 51.

ARAUJO, G. D. de et al. Análise de sentimentos sobre temas de saúde em mídia social. **Journal of Health Informatics**, v. 4, n. 3, 2012. Citado na página 8.

BARNHILL, J. W. **Transtornos fóbicos específicos**. 2018. Disponível em: <https://www.msdmanuals.com/pt/casa/distúrbios-de-saúde-mental/ansiedade-e-transtornos-relacionados-ao-estresse/transtornos-fóbicos-específicos>. Acesso em: 09 de novembro de 2020. Citado na página 19.

BATISTA, G. E. de A. P. A. **Pré-processamento de dados em aprendizado de máquina supervisionado**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, University of São Paulo, São Carlos, 2003. Citado na página 16.

BEAR, M. F.; CONNORS, B. W.; PARADISO, M. A. **Neurociências : desvendando o sistema nervoso**. [S.l.]: Artmed Editora, 2017. ISBN 9788582714331. Citado 2 vezes nas páginas 4 e 5.

BENGFORT, B.; BILBRO, R.; OJEDA, T. **Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning**. 1. ed. [S.l.]: O'Reilly Media, 2018. ISBN 9781491963043. Citado 2 vezes nas páginas 17 e 18.

BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. 2. ed. [S.l.]: Elsevier, 2002. ISBN 9788535210323. Citado na página 26.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit**. 1. ed. [S.l.]: O'Reilly Media, 2009. ISBN 9780596516499. Citado 2 vezes nas páginas 17 e 35.

BOSTAN, L. A. M.; KLINGER, R. An analysis of annotated corpora for emotion classification in text. In: **Proceedings of the 27th International Conference on Computational Linguistics**. Association for Computational Linguistics, 2018. p. 2104–2119. Disponível em: <http://aclweb.org/anthology/C18-1179>. Citado 2 vezes nas páginas 15 e 38.

BROWNLEE, J. **Multi-Label Classification with Deep Learning**. 2020. Disponível em: <https://machinelearningmastery.com/multi-label-classification-with-deep-learning/>. Acesso em: 27 de outubro de 2020. Citado 2 vezes nas páginas 12 e 14.

BROWNLEE, J. **Softmax Activation Function with Python**. 2020. Disponível em: <https://machinelearningmastery.com/softmax-activation-function-with-python/>. Acesso em: 21 de novembro de 2020. Citado na página 41.

CALABREZ, P. **O Que São Emoções e Sentimentos?** 2016. Disponível em: <https://www.youtube.com/watch?v=SUAQeBKiQk0>. Acesso em: 20 de maio de 2020. Citado na página 5.

CARLSON, J. L. **Redis in Action**. 1. ed. [S.l.]: Manning Publications, 2013. ISBN 9781617290855. Citado 2 vezes nas páginas 37 e 38.

CECCON, D. **Os tipos de redes neurais**. 2020. Disponível em: <https://iaexpert.academy/2020/06/08/os-tipos-de-redes-neurais/>. Acesso em: 27 de outubro de 2020. Citado 2 vezes nas páginas 12 e 13.

CHERRY, K. **The 6 Types of Basic Emotions and Their Effect on Human Behavior**. 2020. Disponível em: <https://www.verywellmind.com/an-overview-of-the-types-of-emotions-4163976>. Acesso em: 02 de novembro de 2020. Citado na página 5.

CHOLLET, F. **Deep Learning with Python**. 1. ed. [S.l.]: Manning Publications, 2017. ISBN 9781617294433. Citado 7 vezes nas páginas 8, 10, 11, 12, 13, 14 e 35.

CLARK, D.; BECK, A. T. **Terapia cognitiva para trastornos de ansiedad**. [S.l.]: Desclée De Brouwer, S.A., 2012. ISBN 9788433025371. Citado na página 22.

DAMASIO, A. R. **E O Cerebro Criou O Homem**. [S.l.]: Companhia das Letras, 2011. ISBN 9788535919615. Citado 2 vezes nas páginas 4 e 5.

DATA SCIENCE ACADEMY. **Deep Learning Book**. 2019. Disponível em: <http://www.deeplearningbook.com.br>. Acesso em: 20 de abril de 2020. Citado 2 vezes nas páginas 10 e 18.

DEITEL, P. J.; DEITEL, H. **Python for Programmers: with Big Data and Artificial Intelligence Case Studies**. [S.l.]: Pearson, 2019. ISBN 9780135224335. Citado 2 vezes nas páginas 18 e 34.

DENG, L.; LIU, Y. **Deep Learning in Natural Language Processing**. 1. ed. [S.l.]: Springer Singapore, 2018. ISBN 9789811052088. Citado 3 vezes nas páginas 9, 12 e 14.

DICIO, Dicionário Online de Português. **Indicio**. 2020. Disponível em: <https://www.dicio.com.br/indicio/>. Acesso em: 09 de novembro de 2020. Citado na página 19.

EKMAN, P. **A Linguagem das Emoções**. [S.l.]: Lua de Papel, 2011. ISBN 9788563066428. Citado 5 vezes nas páginas 5, 6, 7, 54 e 55.

FARIAS, R. **Estrutura de Dados e Algoritmos**. 2009. Disponível em: <https://www.cos.ufrj.br/~rfarias/cos121/filas.html>. Acesso em: 28 de novembro de 2020. Citado na página 46.

FIREBASE. **Configurar um app cliente do Firebase Cloud Messaging no Android**. 2020. Disponível em: <https://firebase.google.com/docs/cloud-messaging/android/client>. Acesso em: 15 de setembro de 2020. Citado na página 47.

FIRST, M. B. **Considerações gerais sobre a doença mental**. 2017. Disponível em: <https://www.msdmanuals.com/pt/casa/distúrbios-de-saúde-mental/consideraç~oes-gerais-sobre-cuidados-com-a-saúde-mental/consideraç~oes-gerais-sobre-a-doença-mental>. Acesso em: 26 de outubro de 2020. Citado na página 7.

FOWLER, M. **UML Essencial: Um breve guia para a linguagem-padrão de modelagem de objetos**. 3. ed. [S.l.]: Bookman, 2005. ISBN 9788560031382. Citado 3 vezes nas páginas 25, 26 e 27.

GHAZI, D.; INKPEN, D.; SZPAKOWICZ, S. Detecting emotion stimuli in emotion-bearing sentences. In: **CICLing (2)**. [S.l.: s.n.], 2015. p. 152–165. Citado na página 15.

GOLDBERG, Y. **Neural Network Methods in Natural Language Processing**. [S.l.]: Morgan Claypool Publishers, 2017. ISBN 9781627052955. Citado 3 vezes nas páginas 9, 12 e 14.

GRINBERG, M. **Flask Web Development: Developing Web Applications with Python**. 2. ed. [S.l.]: O'Reilly Media, 2018. ISBN 9781491991732. Citado na página 35.

GUEDES, G. T. A. **UML2: Uma abordagem prática**. 2. ed. [S.l.]: Novatec Editora, 2011. ISBN 9788575222812. Citado 2 vezes nas páginas 25 e 27.

GUERRA, V. **Figma: 5 motivos para ficar de olho!** 2016. Disponível em: <https://medium.com/ui-lab-school/figma-5-motivos-para-ficar-de-olho-d8cfe3af07a1>. Acesso em: 05 de setembro de 2020. Citado na página 48.

HANCOCK, J.; LANDRIGAN, C.; SILVER, C. Expressing emotion in text-based communication. p. 929–932, 01 2007. Citado na página 7.

HEATON, J. **Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks**. [S.l.]: CreateSpace Independent Publishing Platform, 2015. ISBN 9781505714340. Citado 3 vezes nas páginas 11, 12 e 13.

HOSPITAL SANTA MÔNICA. **Conheça alguns Mitos e Verdades em Saúde Mental**. 2018. Disponível em: <http://hospitalsantamonica.com.br/conheca-alguns-mitos-e-verdades-em-saude-mental/>. Acesso em: 30 de maio de 2020. Citado na página 1.

IMME, A. **Ranking das redes sociais: as mais usadas no Brasil e no mundo, insights e materiais gratuitos**. 2020. Disponível em: <https://resultadosdigitais.com.br/blog/redes-sociais-mais-usadas-no-brasil/>. Acesso em: 20 de abril de 2020. Citado na página 8.

JEMEROV, D.; ISAKOVA, S. **Kotlin em ação**. [S.l.]: Novatec Editora Ltda, 2017. ISBN 9788575226100. Citado na página 36.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 2. ed. [S.l.]: Prentice Hall, 2008. ISBN 9780131873216. Citado na página 10.

KERAS. **Embedding layer**. 2020. Disponível em: <https://keras.io/api/layers/core_layers/embedding/>. Acesso em: 01 de agosto de 2020. Citado na página 41.

KERAS. **Input object**. 2020. Disponível em: <https://keras.io/api/layers/core_layers/input/>. Acesso em: 01 de agosto de 2020. Citado na página 41.

KREIBICH, J. A. **Using SQLite**. 1. ed. [S.l.]: O'Reilly Media, 2010. ISBN 9780596521189. Citado na página 38.

KULKARNI, A.; SHIVANANDA, A. **Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python**. [S.l.]: Apress, 2019. ISBN 9781484242667. Citado 2 vezes nas páginas 17 e 18.

LABADESSA, E. O uso das redes sociais na internet na sociedade brasileira. **RMS – Revista Metropolitana de Sustentabilidade**, v. 2, n. 2, 7 2012. Citado na página 7.

LANE, H.; HAPKE, H.; HOWARD, C. **Natural Language Processing in Action: Understanding, analyzing, and generating text with Python**. 1. ed. Rua Sete de Setembro, 111 – 16º andar 20050-006 – Centro – Rio de Janeiro – RJ – Brasil: Manning Publications, 2019. ISBN 9781617294631. Citado 4 vezes nas páginas 10, 14, 17 e 18.

LI, P. et al. Short text emotion analysis based on recurrent neural network. p. 1–5, 08 2017. Citado na página 14.

LI, Y. et al. Dailydialog: A manually labelled multi-turn dialogue dataset. **arXiv preprint arXiv:1710.03957**, 2017. Citado na página 15.

LIMA, C. **Padrão Singleton: Como funcionam? Onde vivem? Do que se alimentam?** 2019. Disponível em: <https://medium.com/@christianmellolima/padr~ao-singleton-como-funcionam-onde-vivem-do-que-se-alimentam-6291fb72b22d>. Acesso em: 23 de novembro de 2020. Citado na página 50.

MACEDO, T.; OLIVEIRA, F. **Redis Cookbook: Practical Techniques for Fast Data Manipulation**. 1. ed. [S.l.]: O'Reilly Media, 2011. ISBN 9781449305048. Citado na página 38.

MANNING, C. D.; RAGHAVAN, P.; SCHUTZE, H. **Introduction to Information Retrieval**. [S.l.]: Cambridge University Press, 2008. ISBN 0521865719. Citado 2 vezes nas páginas 17 e 18.

MCKINNEY, W. **Python Para Análise de Dados: Tratamento de Dados com Pandas, NumPy e IPython**. 1. ed. [S.l.]: Novatec Editora, 2018. ISBN 9788575227510. Citado 2 vezes nas páginas 34 e 35.

MIGUEL, F. K. Psicologia das emoções: uma proposta integrativa para compreender a expressão emocional. **Psico-USF**, scielo, v. 20, p. 153 – 162, 04 2015. ISSN 1413-8271. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-82712015000100015&nrm=iso>. Citado na página 22.

MINDMINERS; FAAP. **Redes Sociais e Saúde Mental**. 2019. Disponível em: <https://mindminers.com/blog/redes-sociais-saude-mental>. Acesso em: 21 de outubro de 2020. Citado na página 1.

MOTTA, D. **Precisamos conversar sobre UX!** 2018. Disponível em: <https://www.hostgator.com.br/blog/ux-design/>. Acesso em: 22 de novembro de 2020. Citado na página 48.

MOZILLA DEVELOPERS NETWORK. **Autenticação HTTP**. 2019. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Authentication>. Acesso em: 22 de novembro de 2020. Citado na página 43.

MOZILLA DEVELOPERS NETWORK. **JSON**. 2019. Disponível em: <https://developer.mozilla.org/pt-BR/docs/JSON>. Acesso em: 28 de novembro de 2020. Citado na página 44.

MOZILLA DEVELOPERS NETWORK. **Base64**. 2020. Disponível em: <https://developer.mozilla.org/en-US/docs/Glossary/Base64>. Acesso em: 22 de novembro de 2020. Citado na página 43.

MüLLER, A. C.; GUIDO, S. **Introduction to Machine Learning with Python: A Guide for Data Scientists**. 1. ed. [S.l.]: O'Reilly Media, 2016. ISBN 9781449369415. Citado na página 34.

NAGY, Z. **Regex Quick Syntax Reference: Understanding and Using Regular Expressions**. [S.l.]: Apress, 2018. ISBN 9781484238752. Citado na página 16.

NEAPOLITAN, R. E.; JIANG, X. **Artificial Intelligence: With an Introduction to Machine Learning, Second Edition (Chapman Hall/CRC Artificial Intelligence and Robotics Series)**. 2. ed. [S.l.]: Chapman and Hall/CRC, 2018. ISBN 9781138502383. Citado 3 vezes nas páginas 9, 10 e 11.

OBE, R. O.; HSU, L. S. **PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database**. 3. ed. [S.l.]: O'Reilly Media, 2017. ISBN 9781491963418. Citado na página 37.

PADUA, J. T. de. **Interfaces em Kotlin: propriedades e métodos padrão**. 2017. Disponível em: <https://medium.com/@jeffersontpadua/interfaces-em-kotlin-propriedades-e-metodos-padrao-697751f3e26b>. Acesso em: 24 de novembro de 2020. Citado na página 49.

PASCANU, R.; MIKOLOV, T.; BENGIO, Y. On the difficulty of training recurrent neural networks. JMLR.org, p. III–1310–III–1318, 2013. Citado na página 13.

PAULEKMANGROUP. **About Paul Ekman**. 2020. Disponível em: <https://www.paulekman.com/about/paul-ekman/>. Acesso em: 02 de novembro de 2020. Citado na página 5.

PEDRO, M. **GoLang: uma introdução**. 2019. Disponível em: <https://blog.geekhunter.com.br/golang/>. Acesso em: 21 de novembro de 2020. Citado na página 36.

PETIT, U. **Simplify your Dataset Cleaning with Pandas**. 2019. Disponível em: <https://medium.com/analytics-vidhya/simplify-your-dataset-cleaning-with-pandas-75951b23568e>. Acesso em: 07 de novembro de 2020. Citado na página 16.

PRAW. **PRAW: The Python Reddit API Wrapper**. 2020. Disponível em: <https://praw.readthedocs.io/en/latest/>. Acesso em: 10 de julho de 2020. Citado na página 35.

REDDIT. **About**. 2020. Disponível em: <https://www.redditinc.com/>. Acesso em: 20 de abril de 2020. Citado na página 8.

REDDIT. **Api Documentarion**. 2020. Disponível em: <https://www.reddit.com/dev/api/>. Acesso em: 20 de abril de 2020. Citado na página 8.

RESENDE, K. **Kotlin com Android: Crie aplicativos de maneira fácil e divertida**. 1. ed. [S.l.]: Casa do Código, 2018. ISBN 9788594188755. Citado na página 36.

RUSSEL, S.; NORVIG, P. **Inteligência Artificial**. 3. ed. [S.l.]: Elsevier, 2013. ISBN 9788535237016. Citado 3 vezes nas páginas 9, 10 e 13.

SANTANA, V. F. de et al. Redes sociais online: Desafios e possibilidades para o contexto brasileiro. **ERI 2009**, n. XV, 10 2009. Citado na página 8.

SCHERER, K. R.; WALLBOTT, H. G. Evidence for universality and cultural variation of differential emotion response patterning. **Journal of personality and social psychology**, v. 66 2, p. 310–328, 1994. Citado 2 vezes nas páginas 7 e 15.

SEBRAE. **Benefícios dos Aplicativos em 2020**. 2020. Disponível em: <https://respostas.sebrae.com.br/beneficios-dos-aplicativos/>. Acesso em: 21 de novembro de 2020. Citado na página 24.

SEGUIN, K. **The Little Redis Book**. [s.n.]. [S.l.]: [s.n.], [210–]. Citado na página 38.

SHAW, P. **Postgres Succinctly**. 1. ed. [S.l.]: Syncfusion, 2014. ISBN 9781642000696. Citado 2 vezes nas páginas 36 e 37.

SOCIAL, W. A.; HOOTSUITE. **Digital in 2018: The Americas**. 2018. Disponível em: <https://hootsuite.com/resources/digital-in-2018-americas>. Acesso em: 10 de maio de 2020. Citado na página 1.

SOSA, P. M. Twitter sentiment analysis using combined lstm-cnn models. 06 2017. Citado na página 61.

SQUARE. **A type-safe HTTP client for Android and Java**. 2020. Disponível em: <https://square.github.io/retrofit/>. Acesso em: 10 de setembro de 2020. Citado na página 49.

STATCOUNTER. **Mobile Operating System Market Share Worldwide**. 2020. Disponível em: <https://gs.statcounter.com/os-market-share/mobile/worldwide>. Acesso em: 23 de novembro de 2020. Citado na página 49.

STONES, R.; MATTHEW, N. **Beginning Databases with PostgreSQL: From Novice to Professional**. 2. ed. [S.l.]: Apress, 2005. ISBN 9781590594780. Citado 2 vezes nas páginas 36 e 37.

STRAPPARAVA, C.; MIHALCEA, R. SemEval-2007 task 14: Affective text. Association for Computational Linguistics, Prague, Czech Republic, p. 70–74, jun. 2007. Disponível em: <https://www.aclweb.org/anthology/S07-1013>. Citado 2 vezes nas páginas 7 e 15.

SUBRAMANIAM, V. **Programming Kotlin: Create Elegant, Expressive, and Performant JVM and Android Applications**. 1. ed. [S.l.]: Pragmatic Bookshelf, 2019. ISBN 9788594188755. Citado na página 36.

SUNNATOVICH, M. I. **Use cases for hash functions or what is SHA-256?** 2019. Disponível em: <https://medium.com/@makhmud.islamov/use-cases-for-hash-functions-or-what-is-sha-256-83036de048b4>. Acesso em: 20 de setembro de 2020. Citado na página 43.

TRIPATHI, H. **What Is Balanced And Imbalanced Dataset?** 2019. Disponível em: <https://medium.com/analytics-vidhya/what-is-balance-and-imbalance-dataset-89e8d7f46bc5>. Acesso em: 05 de novembro de 2020. Citado na página 16.

TSOUMAKAS, G.; KATAKIS, I. Multi-label classification: An overview. **International Journal of Data Warehousing and Mining**, v. 3, p. 1–13, 09 2009. Citado na página 12.

TSUTSUMI, M. **Você lê termo de uso de redes sociais? Veja o que eles dizem**. 2014. Disponível em: <https://www.terra.com.br/noticias/tecnologia/voce-le-termo-de-uso-de-redes-sociais-veja-o-que-eles-dizem,33432f1dab027410VgnVCM10000098cceb0aRCRD.html>. Acesso em: 20 de abril de 2020. Citado na página 8.

TWEEPY. **Tweepy Documentation**. 2020. Disponível em: <http://docs.tweepy.org/en/latest/>. Acesso em: 10 de julho de 2020. Citado na página 35.

TWITTER. **Get Tweet timelines**. 2020. Disponível em: <https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline>. Acesso em: 20 de abril de 2020. Citado na página 8.

VIANA, S. **Por que usar Jupyter Notebook?** 2017. Disponível em: <https://medium.com/@suzana.svm/por-que-usar-jupyter-notebook-77d5a59b42a1>. Acesso em: 21 de novembro de 2020. Citado na página 34.

WAZLAWICK, R. S. **Análise e projeto de sistemas de informação orientados a objetos: modelagem com UML, OCL e IFML**. 3. ed. [S.l.]: Elsevier, 2015. ISBN 9788535279849. Citado na página 25.

WU, C.-H.; CHUANG, Z.-J.; LIN, Y.-C. Emotion recognition from text using semantic labels and separable mixture models. **ACM Trans. Asian Lang. Inf. Process.**, v. 5, p. 165–183, 06 2006. Citado na página 7.

YOUNGMINDS. **Coronavirus: Impact on young people with mental health needs**: Survey 2: Summer 2020. 2020. Disponível em: <https://youngminds.org.uk/media/3904/coronavirus-report-summer-2020-final.pdf>. Acesso em: 15 de maio de 2020. Citado na página 1.

Apêndices

APÊNDICE A – Manual de uso do sistema

SEJA BEM-VINDO AO MANUAL DO SISTEMA *EMOGNIZER*

Antes de explicarmos como o sistema funciona, precisamos esclarecer algumas coisas importantes:

- O sistema proporciona que dois perfis sejam encontrados em redes sociais distintas (Twitter e Reddit). Mas, espera-se que ambos sejam de uma mesma pessoa. Caso contrário, as informações obtidas serão completamente imprecisas.
- O sistema não fará nenhum tipo de diagnóstico.
- O sistema não foi desenvolvido exclusivamente para profissionais pois, todas suas representações podem ser interpretadas por pessoas que não possuem conhecimento científico. Contudo, recomenda-se fortemente que usuários sem conhecimento de transtornos mentais, ao suspeitarem de alguma adversidade em um indivíduo, busquem auxílio de um profissional da área da saúde mental.

LOGIN

Primeiramente, para utilizar o sistema é obrigatório possuir um cadastro. Caso já possua, insira seu e-mail e senha nos campos da tela de login e, em seguida, pressione o botão “Entrar”. Caso contrário, clique no botão “Cadastre-se agora” para abrir o formulário de cadastro.

CADASTRAR NOVO USUÁRIO

Se ainda não é cadastrado, será necessário preencher o formulário abaixo e enviá-lo. Nele, existem três campos: “Nome Completo”, “Email” e “Senha”. Preencha os três com suas informações e pressione o botão “Cadastrar”. Caso já possua um cadastro e veio parar nessa tela, pressione o botão “Clique aqui” ou o “X” no canto superior esquerdo. Ambos levam para a tela de login.

APRESENTAÇÃO DO SISTEMA

Ao adentrar no sistema, alguns slides de apresentação serão expostos. Neles, não existe nenhum tipo de orientação, somente algumas informações sobre os recursos presentes no sistema. Abaixo, é possível visualizá-los, todos são autoexplicativos.

MENU

Após a apresentação, o sistema apresenta um Menu com três possíveis ações. “Pesquisar” é o ponto de partida do fluxo de reconhecimento de emoções, “Histórico” mostra todas as tarefas que já foram criadas e “Sobre” mostra as informações do sistema.

NENHUM PERFIL SELECIONADO

Essa tela é apresentada após o botão “Pesquisar” ser selecionado no Menu. Nela, existem dois botões denominados “procurar”. Ambos os botões redirecionam para a mesma tela de pesquisa. Entretanto, o sistema saberá em qual rede social deve pesquisar de acordo com o botão que foi pressionado. Se estiver familiarizado com o logo das redes sociais, baseie-se por eles. Senão, saiba que se a intenção for procurar um perfil no Twitter, o primeiro botão deve ser pressionado, caso a intenção for procurar um perfil no Reddit, o segundo botão deve ser pressionado. **Antes de pesquisar, lembre-se: Para obter resultados verdadeiros, ambos os perfis precisam ser da mesma pessoa. Se ela não possuir perfis nas duas redes sociais, utilize apenas um.**

PROCURAR USUÁRIO

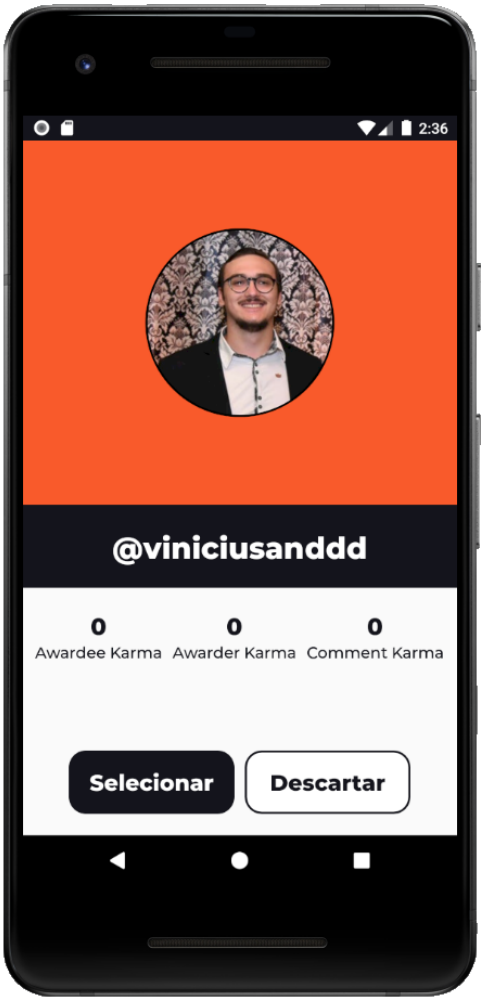
Como anteriormente citado, o sistema apresentará essa mesma tela para ambas as buscas. Logo, só será necessário informar o *username* do usuário que deve ser encontrado na rede social. Lembre-se de informar o *username* corretamente, caso contrário o sistema irá lhe avisar que nenhum usuário foi encontrado.

PERFIL DO TWITTER ENCONTRADO

Se sua busca foi por um perfil do Twitter e ele foi encontrado, você será redirecionado para a tela abaixo. Perceba que são apresentadas as informações do usuário, como, por exemplo sua foto, seu nome, seu *username*, sua localização, o número de *tweets*, o número de seguidores e o número de amigos (pessoas que ele segue). Essas informações são apresentadas para que você tenha certeza que esse é o perfil desejado. Se for, pressione “Selecionar”, caso contrário, pressione “Descartar” e busque novamente.

PERFIL DO REDDIT ENCONTRADO

Se sua busca foi por um perfil do Reddit e ele foi encontrado, você será redirecionado para a tela abaixo. Perceba que são apresentadas as informações do usuário, como, por exemplo sua foto, seu *username* e seu *Karma* (interações do perfil). Em conformidade com a tela anterior, essas informações são apresentadas para que você tenha certeza que esse é o perfil desejado. Se for, pressione “Selecionar”, caso contrário, pressione “Descartar” e busque novamente.



PERFIS SELECIONADOS

Abaixo, é possível visualizar como a tela é apresentada quando ambos perfis são encontrados e selecionados. Para remover algum deles, pressione o botão “x”. Os dois podem ser removidos mas, lembre-se, só será possível iniciar as previsões com, no mínimo, um perfil selecionado. Caso esteja cumprindo esse requisito e deseje começar, preencha os campos com o número de textos que devem ser buscados e pressione “Iniciar”. Vale ressaltar que os campos não são obrigatórios, se eles não forem preenchidos o sistema trará os últimos dez textos de cada perfil.

TAREFA CRIADA COM SUCESSO

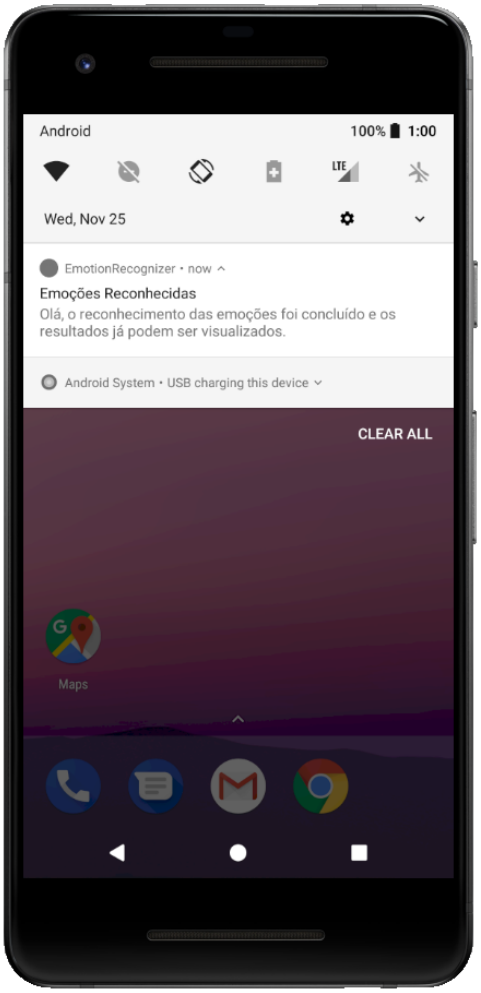
Após ter pressionado o botão “Iniciar”, se tudo ocorreu como esperado, a seguinte tela será apresentada. Nesse momento, nossa Inteligência Artificial esta detectando quais são as emoções dos textos. Aguarde-a, assim que elas forem finalizadas, uma notificação será encaminhada para o seu dispositivo. Caso pressione o botão “Confirmar”, será redirecionado para o Menu.

PREVISÕES CONCLUÍDAS: MENSAGEM NA TELA

Recebeu essa notificação? Isso significa que as previsões foram concluídas e você pode examinar os resultados obtidos. Basta pressionar o botão “Sim” e você será redirecionado. Caso pressione “Não”, o sistema lhe manterá na tela atual. **Não recebeu essa notificação?** Quem sabe as previsões ainda estão ocorrendo. Contudo, lembre-se de conferir sua bandeja de notificações. Para uma melhor compreensão, **veja a próxima página**.

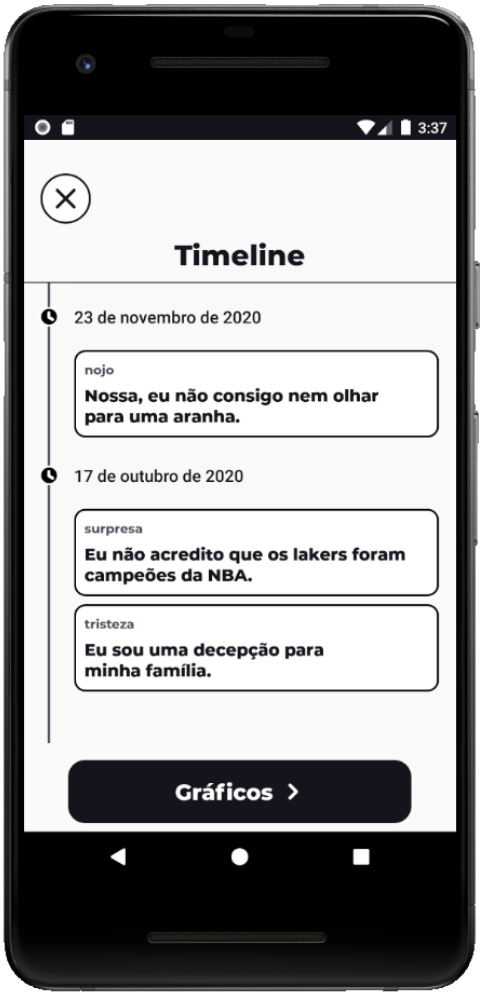
PREVISÕES CONCLUÍDAS: BANDEJA DE NOTIFICAÇÕES

A mensagem mostrada na página anterior aparecerá se o sistema estiver aberto em primeiro plano no seu celular. Caso contrário, uma notificação aparecerá na bandeja de notificações, de forma semelhante ao que é mostrado na imagem abaixo. Para visualizar os resultados, clique na mensagem e, imediatamente o aplicativo será aberto. Será necessário fazer login. Após o login ser efetuado, você será redirecionado para as previsões.



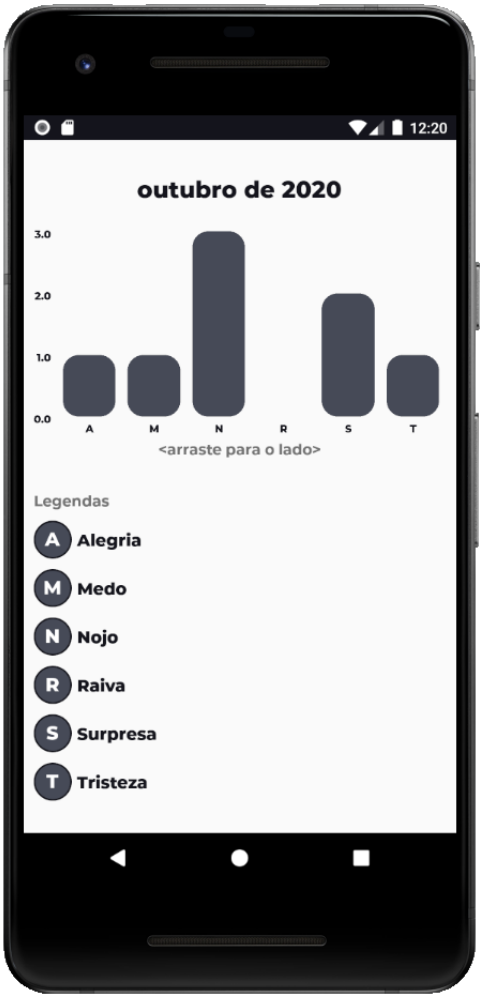
TIMELINE

A primeira forma de visualizar as previsões é utilizando a *timeline*. Ela agrupa os resultados por dia. Nela, é possível visualizar o texto e a emoção prevista.



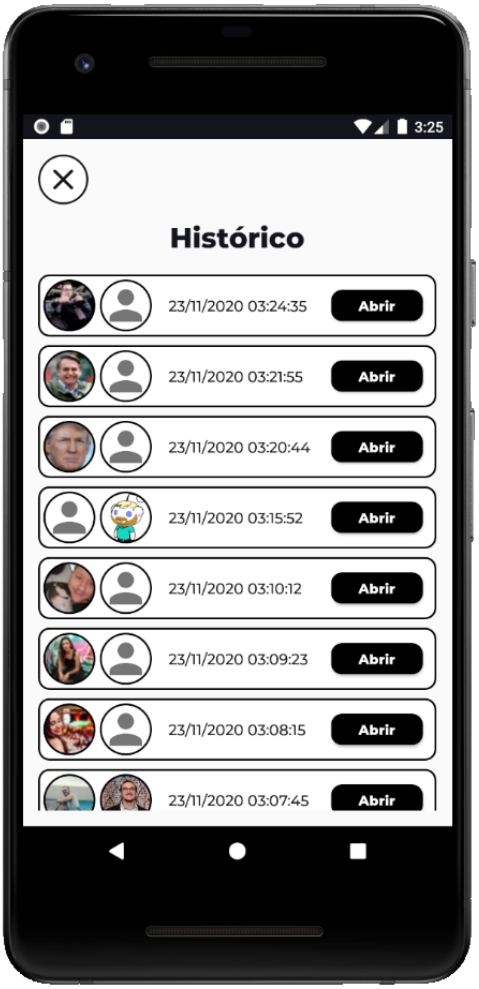
GRÁFICOS

A segunda forma de visualizar as previsões é utilizando os gráficos. Eles agrupam os resultados por mês. Neles, é possível visualizar a quantidade de vezes que cada uma das seis emoções apareceu durante aquele período.



HISTÓRICO DE TAREFAS

Todas suas previsões ocorrem através de tarefas criadas pelo sistema. Logo, todas as previsões estão atreladas a uma tarefa em específico. Para visualizar todas as tarefas criadas, no Menu, acesse “Histórico”. Abaixo, é possível visualizar a tela do histórico de tarefas, onde são mostradas as fotos dos perfis utilizados e a data de criação da tarefa. Para visualizar as previsões, pressione “Abrir”.



ERROS DO USUÁRIO

Essa mensagem sinaliza que a tentativa de localizar um perfil fracassou. Certifique-se que o *username* foi informado corretamente e a busca foi feita na rede social correta.

Essa mensagem sinaliza que os perfis selecionados não possuem textos. Logo, o sistema fica impossibilitado de realizar previsões.

ERROS DO SISTEMA

Essa mensagem ocorre quando o sistema está sobrecarregado, ou seja, algum recurso está sendo utilizado de forma excessiva. Quando ela aparecer, aguarde um curto período de tempo e faça uma nova tentativa.

Essa mensagem não significa que todo o sistema está indisponível, apenas um serviço em específico. Se ele for necessário, em conformidade com a mensagem anterior, aguarde um curto período de tempo e faça uma nova tentativa.

Essa mensagem sinaliza que o sistema não conseguiu prever algum possível erro. Ao contrários das mensagens anteriores, será necessária uma intervenção para corrigir esse problema. Aguarde um período de tempo maior e faça uma nova tentativa.