UNIVERSIDADE DO SUL DE SANTA CATARINA

NATHAN BOTELHO

CLASSIFICAÇÃO TEXTUAL BASEADA EM ANÁLISE DE SENTIMENTO

Palhoça, SC

2021

NATHAN BOTELHO

CLASSIFICAÇÃO TEXTUAL BASEADA EM ANÁLISE DE SENTIMENTO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Flávio Ceci, Dr.

Palhoça, SC

2021

NATHAN BOTELHO

CLASSIFICAÇÃO TEXTUAL BASEADA EM ANÁLISE DE SENTIMENTO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

(Local), (dia) de (mês) de (ano da defesa).

_____

Professor e orientador Flávio Ceci, Dr.
Universidade do Sul de Santa Catarina

_____

Prof. Aran Bey T. Morales, Dr.
Universidade do Sul de Santa Catarina

_____

Prof. Alexandre Vitoreti, MSc.
Universidade do Sul de Santa Catarina

Dedico esta monografia a minha mãe, minha maior e melhor orientadora na vida.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me dado as oportunidades que me deu e que faz com que seja possível seguir o caminho que eu quero seguir. Também agradeço a minha mãe pelo incentivo e todos os cuidados durante todos os anos da minha vida.

Também é importante destacar a ajuda e parceria dos colegas que conheci ao longo do curso e da vida, pela ajuda nos trabalhos, o incentivo ao estudo e evolução profissional, e principalmente pelos momentos de diversão e descontração.

Por fim, agradeço os professores que me motivaram e incentivaram durante a graduação, em especial, o professor orientador do presente trabalho, Flávio Ceci, por todo o suporte fornecido e por servir de inspiração para o tema da monografia.

“Quanto mais aumenta nosso conhecimento, mais evidente fica nossa ignorância”.

(John F. Kennedy, 1962)

RESUMO

O cenário atual da Web 2.0 onde o usuário que era consumidor de informação virou um produtor na criação de conteúdo fez com que seja possível a criação de bases de dados cada vez mais assertivas para a criação de inteligências que compreendem os sentidos e sentimentos em frases bem como a análise destes sentimentos com o objetivo de gerar vantagens competitivas para as organizações nos seus mais variados contextos. Estas bases de dados, que podem ser constituídas pelos mais diversos tipos de dados, como por exemplo, comentários em forma de texto não estruturado, ranqueamento de uma a cinco estrelas, classificações de gosto ou não gosto, entre outros, podem ser utilizadas para entender a opinião de um usuário quanto a um serviço, um produto, ou até uma categoria, e para a elaboração de pesquisas para entender o que seu público-alvo gosta e não gosta, e até o que pode vir a gostar futuramente. A partir dos problemas de análise de sentimentos e processamento de linguagem natural e da pesquisa no material de referencial teórico, onde foram feitos importantes definições sobre o cenário de modo geral, sobre as abordagens e métodos da análise de sentimento e os problemas de processamento de linguagem natural, foi desenvolvido um protótipo funcional utilizando a linguagem de programação Python, a linguagem de programação Typescript, a plataforma Angular, e ferramentas como Docker, Git e GitHub. O protótipo funcional foi testado e avaliado de duas formas diferentes, a forma que se referia ao classificado, onde foi elaborado uma matriz de confusão bem como algumas variáveis de avaliação, como o erro total, a acurácia, a precisão, a revocação e a medida-f, e a forma que diz respeito às interações com o usuário, onde a aplicação foi executada em um ambiente de teste local controlado por cada participante, que fizeram suas considerações através do respondimento de uma pesquisa. Após a análise dos resultados de ambas as avaliações, que foram muito positivas, a aplicação foi considerada apta em analisar as entradas dos usuários.

Palavras-chave: Análise de sentimento. Mineração de opinião. Processamento de linguagem natural.

ABSTRACT

The current scenario of Web 2.0 where the user who was a consumer of information became a producer in content creation made it possible to create increasingly assertive databases for the creation of intelligences that understand the senses and feelings in sentences well as the analysis of these feelings in order to generate competitive advantages for organizations in their most varied contexts. These databases, which can be made up of the most diverse types of data, such as comments in the form of unstructured text, rankings from one to five stars, ratings of likes or dislikes, among others, can be used to understand the opinion of a user about a service, a product, or even a category, and for the preparation of surveys to understand what your target audience likes and dislikes, and even what they might like in the future. From the problems of analysis of feelings and natural language processing and research in the theoretical framework material, where important definitions were made about the scenario in general, about the approaches and methods of feeling analysis and the problems of language processing Naturally, a working prototype was developed using the Python programming language, the Typescript programming language, the Angular platform, and tools such as Docker, Git and GitHub. The functional prototype was tested and evaluated in two different ways, the way it referred to the classified, where a confusion matrix was elaborated as well as some evaluation variables, such as total error, accuracy, precision, recall and f-measure and the way with regard to interactions with the user, where the application was run in a local test environment controlled by each participant, who made their considerations by answering a survey. After analyzing the results of both evaluations, which were very positive, the application was considered able to analyze the user input.

Keywords: Sentiment analysis. Opinion mining. Natural language processing.

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

A evolução dos sistemas de computadores tem trazido diversas oportunidades para que organizações de todos os níveis evoluam os seus padrões de tecnologia, e façam a utilização dos dados disponibilizados na internet em favor de seus contextos. Segundo Blaz (2017, p. 12) “A utilização de softwares via computadores pessoais é crucial em todos os níveis, do suporte às atividades cotidianas no nível operacional, ao acesso a dados cruciais para tomada de decisões estratégicas”.

Nesse contexto empresarial da web e da tecnologia da informação, destacam-se nas organizações diversas técnicas e tecnologias, como a criação de serviços, por exemplo, para a captação de dados públicos ou privados, com a finalidade de atingirem os seus objetivos estabelecidos. Zanchin (2001, p. 11) afirma:

> O “casamento” da informática com as telecomunicações gera a tecnologia da informação. Ela é a grande facilitadora de fatores como a globalização das principais dessas atividades econômicas, a flexibilidade organizacional, as pressões competitivas, a diversidade do trabalho, entre outros fatores. Essas mudanças têm aumentado o interesse das organizações pelo conhecimento. A tecnologia da informação é a ferramenta propiciadora da implantação da gestão do conhecimento nas organizações.

Silva (2013) retrata o presente cenário da era da Web 2.0 afirmando que nos últimos anos os internautas têm observado diversas transformações, em que as antigas páginas estáticas se transformaram em documentos de conteúdo dinâmico e diversificado, que interagem com o usuário das mais diversas formas, e também complementa: “O usuário comum, que antes era apenas um passivo receptor da informação, tornou-se um agente ativo, capaz de contribuir de forma simples com o conteúdo da Web” (SILVA, 2013, p. 1). A partir dessa modernização da web, as páginas estão cada vez mais dinâmicas e tem como objetivo dar voz aos usuários, uma vez que são eles que geram uma épica gama de dados.

A geração de dados da web é continua e acontece a cada minuto. Avila (2017, p. 11) diz que “Um grande número de textos não estruturados, curtos e informais são publicados diariamente tanto na internet (ex: fórum, blog, twitter) como nas redes corporativas (ex: pesquisa de satisfação, comentários na intranet)”. Diversos usuários acessam suas contas em inúmeras plataformas, fornecendo dados muitas vezes em forma de textos não estruturados, de forma desconexa ou sem contexto, tudo usando um dispositivo eletrônico conectado à internet e sendo disponibilizado em maioria publicamente, contribuindo direta ou

indiretamente para a formação de uma grande base de dados, que possuem um valor gigantesco para as empresas que fazem sua captação.

A aquisição de dados empresariais em forma de classificação ou ranqueamento é um dos mais valorosos dados e que geram maiores resultados benéficos para a organização. A empresa quer ser avaliada pelos seus produtos ou serviços, então, disponibiliza aos usuários formas de realizar essa classificação, seja em forma de comentário ou categorização com algum valor numérico, por exemplo, de zero a cinco estrelas, entre outras formas. Entender o que as pessoas pensam ou qual o seu sentimento sobre determinado assunto é uma informação relevante e normalmente utilizada para o processo de tomada de decisão (AVILA, 2017).

É de extrema importância para uma organização se basear na opinião do seu público-alvo para montar tanto seu portifólio quanto sua estratégia. Nesello (2014) diz que o termo *big data* teve origem devido a quantidade de dados produzida diariamente por operações comerciais, financeiras, mídias sociais, entre outros. Nesello (2014, p. 18) também comenta que "o potencial de agregação de valor de *big data* nos negócios, se reflete em uma melhor satisfação dos clientes, melhores serviços e na contribuição para criação e manutenção de um negócio bem-sucedido". Um ponto importante nesse processo é a categorização dos dados, que se acontecem a partir do processamento de análise de sentimento, que são técnicas de inteligência artificial e aprendizagem de máquina, para a classificação de itens, construção de perfis, detecções de padrões, entre outros.

1.1 PROBLEMÁTICA

A atualidade e a relevância do termo *big data* e a quantidade massiva dos dados na web assim como suas aplicações nas organizações trazem o foco para o assunto análise de sentimento. Na indústria, o termo análise de sentimento é mais usado, mas na academia tanto a análise de sentimento quanto a mineração de opinião são frequentemente empregados (LIU, 2012). Liu (2012) também explica que a análise de sentimento, também conhecida como mineração de opinião, é o campo de estudo que analisa opiniões, sentimentos, avaliações, atitudes e emoções, e faz uma breve introdução aos principais problemas de pesquisa com base no nível de granularidade, destacando três níveis principais:

- Nível do documento: A tarefa neste nível é classificar se todo um documento de opinião expressa um sentimento positivo ou negativo, conforme Pang et al. (2002) e Turney (2002), ambos citados por (LIU 2012, p. 10).
- Nível da frase: A tarefa neste nível vai para as frases e determina se cada frase expressou uma opinião positiva, negativa ou neutra.
- Nível de entidade e aspecto: As análises de documento e de frase não descobrem exatamente o que as pessoas gostaram e não gostaram. Em vez de olhar para construções de linguagem (documentos, parágrafos, sentenças, cláusulas ou frases), o nível de aspecto olha diretamente para a própria opinião. Parte-se da ideia de que uma opinião consiste em um sentimento (positivo ou negativo) e um alvo (de opinião).

Existem alguns problemas que o processamento de linguagem natural procura resolver antes de partir à análise do sentimento. Segundo Blaz (2017, p. 18):

> O sentimento pode ser identificado no documento como um todo, individualmente para cada sentença, ou em nível de aspecto. Antes dessa classificação, deve ser realizado um pré-processamento, no qual o texto original sofre transformações via técnicas de processamento de linguagem natural (PLN) necessárias à classificação de sentimentos em si.

Liu (2012) comenta sobre os problemas do processamento de linguagem natural e enfatiza a importância de não esquecer que a análise de sentimento é um campo da *PLN*, atingindo todos seus aspectos, como resolução de correferência, manipulação de negação e desambiguação de sentimento.

A resolução de correferência, manipulação de negação e desambiguação de sentimento são técnicas de processamento de linguagem natural. A manipulação de negação, uma das mais básicas, refere-se a um tipo de *Sentiment shifters*, que são expressões que são usadas para inverter a orientação do sentimento (LIU, 2012). Também segundo Liu (2012) a correferência refere-se ao problema de determinar múltiplas expressões em uma frase ou documento referente à mesma coisa, ou seja, eles têm o mesmo "referente". Já a desambiguação de sentimento, desambiguação do sentido da palavra, ou *word sense disambiguation* diz respeito a identificação de uma palavra que pode ter múltiplo significados (BARROS, 2018).

Alguns problemas no texto estudado se caracterizam por dificultar a análise em si, ao tornar o texto ambíguo (dar ao texto mais de um entendimento possível) ou ao utilizar

subjetividade (abrindo margem para múltiplas interpretações da mesma frase), potencializando a margem para os erros da análise. Balague Filho (2017) comenta que é importante perceber que o conceito de subjetividade não está diretamente relacionado ao conceito de sentimento. É possível ter a frase “acho que sou melhor”, que é subjetiva, mas não tem sentimento, mas a frase “a bateria não durou 2 minutos” é objetiva e tem sentimento negativo implícito (BALAGE FILHO, 2017).

Um problema bastante relevante na análise geral e classificação dos textos são os aspectos culturais ou assuntos de senso comum que são interpretados pelos seres humanos de forma natural podendo atribuir um sentimento ou uma opinião específica a uma combinação curtas de palavras. Um exemplo da atribuição automática de opinião dos seres humanos a pequenos textos demonstra-se a partir do fato de ao pensar em “comida gelada” não se formar a mesma opinião ou a mesma emoção quando se pensa em “cerveja gelada” ou “refrigerante gelado”. Silva (2016, p. 8) comenta que “ambientes que favorecem a geração de textos curtos e informais são cenários de pesquisa desafiadores para análise de sentimento, nos quais o principal objetivo é extrair avaliações de produtos, categorização por sentimentos, agrupamento e extração de padrões comportamentais”.

Apesar dos problemas descritos anteriormente destacarem-se individualmente, o principal problema de atribuição de inteligência como um todo é a capacidade do ser humano e sua individualidade, bem como seu pensamento não “estruturado”, levando todo seu histórico de vida consigo em seu raciocínio lógico geral.

Atualmente existem ferramentas de processamento de linguagem natural que são consistentes e resolvem o problema de diversas organizações que precisam ou querem empregar um utilitário do gênero em seus diversos e complexos contextos. Com o avanço de serviços de construções inteligência artificial de chatbots e itens do meio, já existem ferramentas que disponibilizam esse tipo conteúdo para empresas sem a necessidade de um programador ou um empregado capacitado na área.

Com a atual evolução de inteligência artificial, *machine learning* e vertentes desses, tais como os serviços mencionados anteriormente, já é possível criar um modelo de aprendizagem autônoma possibilitando a busca e aprendizagem automática de respostas usando grandes bases de dados disponíveis. Entretanto, apesar de existirem abordagens como a de Ekman que considera os seis principais sentimentos sendo elas a surpresa (*surprise*), a alegria (*joy*), a tristeza (*sadness*), a raiva (*anger*), o medo (*fear*) e o nojo (*disgust*) (EKMAN, 1992) ou a abordagem Valence-Arousal-Dominance (VAD) que separa nas três dimensões: valência (estimulo prazeroso ou não), excitação (intensidade provocada) e dominância (grau

de controle exercido no ser humano) (WARRINER; KUPERMAN; BRYSBAERT, 2013), e até abordagens mais simples como a classificação de sentimento sendo formulada como um problema de aprendizagem com três classes, positiva, negativa e neutra (LIU; ZHANG, 2012), o cérebro humano faz considerações únicas baseado nos problemas diários e na sua experiência de vida na interpretação de um texto, gerando uma dificuldade para a criação de uma inteligência artificial que adapte-se a esses padrões.

Dada a problemática contextualizada anteriormente, o presente trabalho tem por finalidade responder: Como construir um modelo de análise e classificação de sentimentos em textos de forma a construir um perfil baseado em análises anteriores?

1.2 OBJETIVOS

Esta seção apresenta os objetivos geral e específico que norteiam o desenvolvimento deste trabalho.

1.2.1 Objetivo Geral

Desenvolver uma proposta de solução para tornar possível a polarização de sentimentos predominantes em textos, bem como arquivar os textos polarizados empregando-os para novas classificações.

1.2.2 Objetivos Específicos

Tem-se como objetivos específicos os seguintes itens:

- Identificar métodos e técnicas de análise de sentimento que possam apoiar a classificação de textos;

- Definir um cenário de aplicação;
- Modelar uma proposta de solução a partir do cenário definido;
- Desenvolver um protótipo funcional a partir da modelagem proposta;
- Avaliar o protótipo funcional a partir de métricas de classificação e a partir de interações com o usuário.

A próxima seção apresenta a justificativa do trabalho.

1.3 JUSTIFICATIVA

O presente trabalho justifica-se pelo crescimento massivo de diversas novas ferramentas, linguagens e frameworks, e principalmente, a crescente necessidade de análise de dados em diversos contextos organizacionais, esperando contribuir positivamente com a área apresentando uma solução baseada na integração de divergentes tecnologias, antigas e novas.

Para Wajzenberg (1998) a tecnologia representa uma vantagem competitiva, podendo significar a sobrevivência ou a derrocada de uma organização, e até, num sentido mais amplo, de um país. O momento tecnológico das organizações privadas, que buscam suas vantagens competitivas baseadas em suas estratégias, bem como a atualidade do assunto também se destacam na escolha do tema para a resolução da problemática proposta, uma vez que, com a ascensão da internet em geral, principalmente das redes sociais e comércios eletrônicos, surgem novas necessidades de ferramentas como recomendações automáticas, processamento de perguntas frequentes, processamento de fala, entre outras demandas baseadas em inteligência artificial.

Segundo Silva (2013) “Análise automática de sentimento é uma área de pesquisa recente, que ganhou mais destaque no início deste século”. Para Garcia (2017) compreender e produzir texto em linguagem humana pode ser a atividade mais complexa que os humanos fazem. O processo integra sentimentos, emoções, memória e uma variedade de sistemas de processamento de informação sempre que é utilizado para escrever ou ler uma mensagem.

O tema da pesquisa também foi escolhido pela curiosidade e o interesse do autor, dada toda a contextualização já descrita de sua importância, sua complexidade, sua atualidade,

visando principalmente a contribuição com o cenário a partir da tentativa de publicação de uma proposta com uma abordagem do autor sobre o tema, concordantemente com Oliveira (2013) que diz que “As atividades humanas são orientadas, em grande parte, por resultados de pesquisas e desenvolvimentos tecnológicos divulgados diariamente [...]”. Espera-se também, ao contribuir com o proposto tema, atrair foco para a área, para que novos trabalhos e abordagens sobre conceitos de análise de linguagem natural e classificação textual baseada em análise de sentimento se desenvolvam, assim como as tecnologias usadas para tal.

1.4 ESTRUTURA DA MONOGRAFIA

O presente trabalho é composto por seis capítulos, estando estruturados da seguinte maneira:

- Capítulo 1: Introdução, objetivos, problemática e justificativa.
- Capítulo 2: Referencial teórico.
- Capítulo 3: Metodologia da pesquisa.
- Capítulo 4: Proposta da solução.
- Capítulo 5: Desenvolvimento.
- Capítulo 6: Conclusões e trabalhos futuros.

O próximo capítulo apresenta o referencial teórico que servirá de base para os futuros desenvolvimentos do trabalho.

2 REFERENCIAL TEÓRICO

Esta seção apresenta o referencial teórico apresentado pelo presente trabalho, descrevendo e explicando conceitos utilizados, sendo eles: A análise de sentimento, a classificação de texto e documentos e o processamento de linguagem natural.

2.1 ANÁLISE DE SENTIMENTO

É sabido que a análise de sentimento é uma área da inteligência artificial que tem por objetivo classificar opiniões, sentimentos, avaliações, entre outros.

Silva (2016) resume e explica que a área de análise de sentimento é um campo de estudo recente devido ao crescimento da internet e seu conteúdo, principalmente nas redes sociais, nas quais as pessoas publicam suas opiniões em uma linguagem coloquial. A mesma autora também descreve que em muitos casos, é utilizado artifícios gráficos para tornar ainda mais sucintos os diálogos das redes. Balage Filho (2017) também resume explicando a área de análise de sentimento como o campo de estudo que extrai e interpreta o sentimento, geralmente classificado como positivo ou negativo, em direção a algum aspecto em um texto de opinião.

A análise de sentimento (em inglês, *sentiment analysis*), mineração de opinião (em inglês, *opinion mining)* ou análise de subjetividade (em inglês, *subjectivity analysis)* são algumas variações de um mesmo objeto de estudo (CECI et al, 2016), e são denominadas distintamente pelo motivo de serem tarefas ligeiramente diferentes (LIU, 2012). Apesar de existirem também, além das já descritas, algumas outras nomenclaturas para essas variações, como extração de opinião (em inglês, *opinion extraction)*, mineração de sentimento (em inglês, *sentiment mining*), análise de afeto (em inglês, *affect analysis*), análise de emoção (em inglês, *emotion analysis*) ou mineração de revisão (em inglês, *review mining*), entre outros (LIU, 2012), de acordo com Liu (2012, p. 7, tradução nossa): “eles agora estão todos sob a égide da análise de sentimento ou mineração de opinião”.

Como descrito anteriormente, a importância da área e sua atualidade podem ser de extrema vantagem) competitiva para diversas organizações, ao reconhecer o que um grupo de usuários pensa sobre algum produto. Para Ceci, Alvarez e Gonçalves (2016, p. 20):

> Opiniões são utilizadas para explicitar pontos de vista¸ de maneira que os pensamentos de outras pessoas podem ser úteis para o processo de tomada de decisão. As organizações fazem uso desse tipo de instrumento por meio das opiniões relacionadas a seus produtos e serviços, direcionando suas ações estratégicas.

Para Liu (2012) as opiniões são centrais para quase todas as atividades humanas e são os principais influenciadores de nossos comportamentos. Na próxima seção, são apresentadas breves explanações e definições a respeito das emoções e suas ligações com as opiniões.

2.1.1 As emoções

O sentimento é estudado de diversas formas tanto em questões de biológicas ou de cunho comportamental, quanto sua escrita, subjetividade e discurso, por diversas áreas diferentes, como por exemplo a antropologia (CECI et al, 2016), a filosofia (BALAGE FILHO, 2017, CECI et al, 2016, LIU, 2012), as ciências biológicas (CECI et al, 2016), a psicologia (BALAGE FILHO, 2017, CECI et al, 2016, FOSCHIERA, 2012, LIU, 2012), a psicanálise (FOSCHIERA, 2012), a ciência da computação (CECI et al, 2016), a linguística (FOSCHIERA, 2012) e a sociologia (BALAGE FILHO, 2017, LIU, 2012), entre outros.

Miguel (2015, p. 154) no seu artigo *Psicologia das emoções: uma proposta integrativa para compreender a expressão emocional* explica que “as teorias psicoevolucionistas propõem que os estados emocionais existem hoje como reflexo da evolução das espécies, ou seja, como respostas adaptativas a situações que ocorrem no meio”. O autor também complementa em seu artigo que as abordagens cognitivistas, embora não discordem totalmente da origem evolutiva e nem neguem a influência das alterações viscerais, destacam a avaliação da situação como sendo a principal característica da emoção.

Já no artigo *Psicologia, Metafísica e Literatura: a Descrição dos Sentimentos Profundos em Bergson* o autor Rodrigues (2013) afirma que do ponto de vista objetivo, um sentimento pode ser investigado como um sistema de manifestações orgânicas bem definidas.

Apesar de tudo, segundo o autor, é a filosofia que deve se atribuir a tarefa de investigar a natureza dos sentimentos de um ponto de vista radicalmente distinto da ciência (RODRIGUES, 2013).

Voltando ao campo da ciência da computação, Liu (2012, p. 11, tradução nossa) define as emoções como “nossos sentimentos e pensamentos subjetivos”, e explica a relação de emoções com sentimentos, voltado ao processo computacional de analisar sentimentos:

> As emoções estão intimamente relacionadas aos sentimentos. A força de um sentimento ou opinião está tipicamente ligada à intensidade de certas emoções, por exemplo, alegria e raiva. As opiniões que estudamos na análise de sentimento são principalmente avaliações (embora nem sempre).

De acordo com Balage Filho (2017) os sentimentos podem ser vistos como um termo genérico para designar todo o texto que expressa características positivas, negativas ou neutras. O mesmo autor também diz que o termo “sentimento” é amplamente utilizado e pode referir-se à subjetividade, emoção, avaliação e opinião. Já a subjetividade pode ser vista como “[...] a presença no texto de sentimentos, pontos de vista ou crenças pessoais. Assim, uma frase subjetiva é aquela que contém qualquer crença. Em oposição, a frase pode ser objetiva” (BALAGE FILHO, 2017, p. 25, tradução nossa).

2.1.2 Abordagens e métodos

A mineração de opinião e análise de sentimento surgiram com a intenção de atuar na identificação de recursos computacionais, para identificar, classificar, e analisar opiniões e sentimentos (CECI et al, 2016), e apesar de áreas que contribuem com a análise de sentimento, como por exemplo a linguística e o processamento de linguagem natural terem longas histórias, poucas pesquisas foram feitas sobre esses temas antes do ano 2000 (LIU, 2012). Apesar da maior parte dos estudos sobre os temas relacionados se concentrarem recentemente, algumas técnicas foram desenvolvidas até então, e serão descritas nessa seção, juntamente com a descrição e definição das áreas abordadas.

2.1.2.1 Machine learning

Aprendizagem de máquina, (do inglês, *machine learning*) é um tipo de algoritmo que consiste em aprender características sobre um determinado conjunto de dados conhecidos como dados amostrais de treinamento (do inglês, *training sample*), e aplicar esse conhecimento em outro conjunto de dados distinto, conhecido como dados amostrais de teste (do inglês, *test sample*), para predizer suas características, destacando que o conjunto de testes não é utilizado durante a etapa de aprendizado do algoritmo (AVILA, 2017).

Os algoritmos de aprendizado de máquina tentam descobrir a estrutura dos dados e isso geralmente significa descobrir uma relação preditiva entre as variáveis (BENGIO, 2012). De maneira geral, "significa descobrir onde a massa de probabilidade se concentra na distribuição conjunta de todas as observações variáveis" (BENGIO, 2012, p.17, tradução nossa).

As estratégias de aprendizado de máquina tradicionalmente utilizadas dependem da forma como os dados são representados (PEREIRA, 2017) e podem fazer uma grande diferença no sucesso de um algoritmo de aprendizado (BENGIO, 2012).

2.1.2.2 Aprendizagem supervisionada

Os algoritmos de aprendizagem supervisionada, ou *supervised learning*, podem ser abstraídos em termos de pares (X, Y), em que X é uma variável aleatória de entrada e Y é um rótulo que desejamos prever dado X (BENGIO, 2012).

A principal característica da aprendizagem supervisionada é que o conjunto de dados disponível contém exemplos explícitos de qual é a saída correta, baseado na entrada informada (AVILA, 2017). Mohri et al (2012) citado por Avila (2017) separa as abordagens da aprendizagem supervisionadas em dois tipos, sendo eles, a classificação e a regressão:

> Na classificação, as amostras pertencem a uma ou mais classes e queremos identificar as classes dos dados de teste baseado no conhecimento adquirido a partir dos dados de treinamento. Na regressão, a ideia é similar a classificação, porém ao invés de saídas discretas, a saída do algoritmo consiste em uma variável continua, como por exemplo um n´úmero real (AVILA, 2017, p. 17).

Batista (2003) explica que no aprendizado supervisionado é fornecido ao sistema de aprendizado um conjunto de exemplos $E = \{E_1, E_2, \ldots E_N\}$, sendo que cada exemplo $E_i \in E$ possui um rótulo associado. De acordo com o autor, esse rótulo define a classe à qual o exemplo pertence, formalmente definindo cada exemplo $E_i \in E$ é uma tupla, $E_i = (\vec{x}_i, y_i)$ na qual $\vec{x}_i$ é um vetor de valores que representam as características, ou atributos, do exemplo E_i, e y_i é o valor da classe desse exemplo (BATISTA, 2003).

2.1.2.3 Aprendizagem não supervisionada

Na aprendizagem não supervisionada, ou *unsupervised learning*, o conjunto de dados disponível, ou de treinamento, não contém nenhuma informação a respeito da saída para determinada entrada e todos os dados de treinamento se resumem a informações de entradas possíveis (AVILA, 2017).

Batista (2003, p. 19) explica que "na aprendizagem não supervisionada é fornecido ao sistema de aprendizado um conjunto de exemplos *E*, no qual cada exemplo consiste somente de vetores $\vec{x}$, não incluindo a informação sobre a classe y". Segundo o autor, o método tem por objetivo construir um modelo que procura por regularidades nos exemplos, formando agrupamentos ou clusters de exemplos com características similares (BATISTA, 2003).

2.1.2.4 Deep Learning

A aprendizagem profunda (do inglês, *deep learning*) pode ser entendida como a aplicação de técnicas de aprendizagem de máquina para a aprendizagem de novos padrões (CECI, 2015). As estratégias de aprendizagem profunda geralmente usam redes neurais artificiais que aprendem representações distribuídas dos dados de entrada e em tais representações, cada neurônio da rede participa da composição de diferentes conceitos (PEREIRA, 2017).

Os algoritmos de aprendizagem profunda buscam explorar a estrutura desconhecida na distribuição de entrada para descobrir boas representações, geralmente em vários níveis, com recursos aprendidos de nível superior definidos em termos de recursos de nível inferior (BENGIO, 2012). Segundo o autor, o objetivo é “tornar essas representações de nível superior mais abstratas, com suas características individuais mais invariáveis para a maioria das variações que estão normalmente presentes na distribuição de treinamento, preservando coletivamente o máximo possível das informações na entrada” (BENGIO, 2012, p. 17, tradução nossa).

Os principais algoritmos de aprendizagem profunda utilizados são baseados em redes neurais artificias (CECI, 2015), técnica que será descrita na próxima seção.

2.1.2.5 Redes neurais

Uma rede neural, ou *neural network*, que foram inspiradas, em parte, pela observação de sistemas de aprendizado biológico mais complexos, como o cérebro (MITCHELL et al, 1994 apud PEREIRA, 2017), consistem em muitos neurônios conectados entre si (BARCHI, 2020, PEREIRA, 2017), cada um sendo uma unidade de computação que produz uma sequência de ativações de valor real (BARCHI, 2020).

Uma arquitetura de rede neural artificial, ou *artificial neural networks (ANN)*, é organizada em camadas (BARCHI, 2020), sendo elas: a camada de entrada (do inglês, *input layer*) que é caracterizada pelas *features* extraídas a partir dos dados alimentam as unidades de entrada, também conhecidos como *units* (PEREIRA, 2017), a camada oculta (do inglês, *hidden layer*) que podem ser apenas uma ou várias camadas (BARCHI, 2020), na qual o dado é processado (PEREIRA, 2017), e camada de saída (do inglês, *output layer*), que retorna um valor de saída, sendo que esse valor normalmente pertence ao intervalo de 0 até 1 (PEREIRA, 2017). Uma rede neural profunda, ou *deep neural network*, ocorre quando a rede neural possui duas ou mais camadas escondidas (LECUN et al, 2015 apud PEREIRA 2017, BENGIO et al, 2015 apud PEREIRA, 2017).

De acordo com Barchi (2020, p. 10, tradução nossa) “[...] com as informações das camadas anteriores, as conexões com pesos ativam os neurônios da próxima camada. Cada neurônio tem *n* entradas *i*, pesos (*w*), viés (*b*), uma função de ativação (*F (x)*) e saída (*y*)”.

Formalmente, a saída *y* do "*j*-ésimo" neurônio é expressa com a equação 1.0 (BARCHI, 2020, p. 10):

$$y_j = F\left(b + \sum_{k=1}^{n} w_{k,j}\, i_k\right) \quad (1)$$

Barchi (2020, p. 10-11, tradução nossa) explica:

> As entradas com pesos e a tendência são parâmetros ajustáveis que tornam a rede neural um sistema parametrizado. Entre as funções de ativação não lineares, a função logística e a unidade Linear Retificada (ReLU) são duas das mais utilizadas. A função de ativação logística é fortemente aplicada para prever probabilidades, uma vez que varia de 0 a 1. ReLU está presente na maioria das redes neurais convolucionais e é dada pela seguinte equação: $R(x) = max(0, x)$, por exemplo , $R(x)$ é zero para $x < 0$ e x quando $x > 0$. O processo de treinamento otimiza os pesos para cada neurônio, minimizando o erro das previsões e atingindo um nível especificado de precisão.

Apesar das redes neurais resolverem problemas de classificação de classificação, outras técnicas também foram criadas para resolver o mesmo problema, bem como otimizar tal processo. Um das técnicas é o *support vector machine*, ou SVM, que se apresenta na próxima seção.

2.1.2.6 Support vector machine

Support vector machine (SVM) é uma técnica que tem por objetivo encontrar o limite para decidir entre a classificação em duas classes, utilizando treinamento de dados (CECI, 2015), tendo como principais vantagens a efetividade em espaços com várias dimensões, efetividade em espaços onde o número de dimensões é maior do que o número de amostras e também versatilidade, uma vez que utiliza diferentes funções no seu kernel para definir a forma de decisão do algoritmo (AVILA, 2017).

Para Semolini (2002) a técnica de *support vector machine* baseia-se nos princípios da minimização do risco estrutural, proveniente da teoria do aprendizagem estatístico, a qual está baseado no fato de que o erro do algoritmo de aprendizagem, junto aos dados de validação, é limitado pelo erro de treinamento mais um termo que depende da dimensão VC (dimensão Vapnik e Chervonenkis), que é uma medida da capacidade de expressão de uma família de funções. Para o autor, "o objetivo é construir um conjunto de hiperplanos tendo

como estratégia a variação da dimensão VC, de modo que o risco empírico (erro de treinamento) e a dimensão VC sejam minimizados ao mesmo tempo" (SEMOLINI, 2002, p. 4).

O algoritmo de SVM é altamente efetivo na categorização de texto, e é uma importante referência para o problema de análise de sentimento (AVILA, 2017), sendo performaticamente superior ao algoritmo *Naive Bayes* (JOACHIMS, 1998 apud AVILA, 2017), apresentado na próxima seção.

2.1.2.7 Naive Bayes

O método de Naive bayes (NB) se refere a classificação baseado em inferência bayesiana (MAIA, 2008). Segundo Manning e Schütze (1999) citados por Ceci (2015), as abordagens bayesianas são fundamentadas em estatísticas e muito utilizadas para auxiliar no processamento de linguagem natural. O método tem sido utilizado a mais de cinquenta anos, principalmente na área de recuperação de informação (MARON, 1961 apud AVILA, 2017).

Os classificadores de Naive bayes trabalham com dados contínuos e discretos (MAIA, 2008), sendo uma técnica de aprendizado de máquina supervisionado (LEWIS, 1998 apud AVILA, 2017, MANNING et al 2009 apud CECI, 2015), criada a partir de um conjunto de dados inicial utilizados para treinamento (MANNING et al 2009 apud CECI, 2015), tendo em vista que para dados discretos os valores de probabilidades são coletados através da contagem nos grupos dos documentos, e para dados contínuos, ele assume que os valores sigam uma função de distribuição normal, assim, as probabilidades são inferidas a partir da média e do desvio padrão de grupos dos documentos (MAIA, 2008).

Avila (2017) define a base desse classificador como o teorema da probabilidade, conhecido como bayes ou regra de bayes:

$$P(C = c_k \mid X = x) = P\,(C = c_k)\; x\; \frac{P(X = x \mid C = c_k)}{P\,(x)} \qquad (2)$$

Ou seja:

$$P(X = x) = \sum_{k^j=1}^{e_c} P(X = x \mid C = c_{k^j}) \; x \; P(C = c_{k^j}) \tag{3}$$

Avila (2017, p. 27) explica que “todas as pesquisas de satisfação são classificadas, exatamente, em apenas uma das e_c classes possíveis: (c_1, c_2, ..., c_k, ..., c_n)”. O autor também explica que, dado o conceito do problema de classificação em questão, um exemplo de classes possíveis seriam: “elogio”, “neutro” e “reclamação”.

Manning, Rachavan e Schütze (2009) citados por Ceci (2015) também acreditam que a família naive bayes são baseadas em métodos probabilísticos, e representam o cálculo baseado na seguinte equação (4):

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P\ (t_k|c) \tag{4}$$

Além das importantes técnicas para a análise de sentimentos apresentadas anteriormente nesse capítulo, também são necessárias técnicas de classificação de texto e documentos, bem como obtenção, filtragem, e semântica dos dados disponíveis. Essas técnicas serão apresentadas na próxima seção.

2.2 CLASSIFICAÇÃO DE TEXTO E DOCUMENTOS

Classificar documentos textuais é uma tarefa importante para o processo de analisar sentimentos. A quantidade de informações a serem acessada para que pessoas e organizações desempenhem suas tarefas adequadamente aumentam a cada dia (BASTOS, 2015). Grande parte das informações a serem acessadas não estão disponíveis em sistemas de informação (BRUGGEMANN et al, 2000 apud BASTOS, 2015) como em bancos de dados, modelos estruturados, páginas HTML, mas em documentos (BRUGGEMANN et al, 2000 apud BASTOS, 2015) muitas vezes em forma de texto não estruturados, e com a finalidade de regatar e compreender seu conteúdo para ser então processado, serão apresentadas técnicas nessa seção.

2.2.1 Mineração de dados

Mineração de dados (do inglês, *data mining*) é um dos campos que estuda a análise de sentimento (LIU, 2012), sendo altamente interdisciplinar e apresentando grande intersecção com diversos outros, como aprendizado de máquina e reconhecimento de padrões, além de fazer uso de diversos conceitos de computação e estatística (ARRUDA, 2013).

O processo de mineração de dados consiste em extrair padrões de grandes quantidades de dados podendo ser feito por meio de métodos supervisionados como classificação e por métodos não supervisionados, como por exemplo, agrupamento de dados (ARRUDA, 2013).

A mineração Web (também conhecido como mineração de dados web, *web data mining* ou *web mining*), é o processo de descobrir informação útil em dados da web, por meio de técnicas de mineração de dados (LIU, 2007 apud PEREIRA JUNIOR, 2008, CHAKRABARTI, 2002 apud PEREIRA JUNIOR, 2008), facilitando a busca e obtenção de dados em páginas HTML, por exemplo.

Pereira Junior (2008) resume o processo de mineração web como sendo um processo iterativo, no qual prototipagem tem um papel essencial para experimentar facilmente com diferentes alternativas, bem como para incorporar o conhecimento adquirido durante iterações anteriores do processo.

2.2.2 Classificação de texto

A classificação de texto (TC, do inglês *text classification* ou *text categorization*) é a tarefa de atribuir a um determinado texto em linguagem natural, em algum idioma, uma classe de um universo finito e pré-estabelecido (PINHEIRO, 2011).

Nas últimas décadas, tarefas relacionadas a gestão de conteúdo de documentos ganharam destaque proeminente na computação (FIGUEIREDO, 2008), destacando-se as aplicações do gênero que tornam-se cada vez mais difundidas, como na detecção de spam em e-mails, na remoção nas mensagens de conteúdo suspeito, na organização de documentos em

tópicos hierárquicos, na facilitação de pesquisas diversas, sistemas de recomendação, entre outros (PINHEIRO, 2011).

Segundo Pinheiro (2011) apesar dos problemas relacionados a TC serem problemas antigos estudados desde a década de 1960, sua importância cresceu apenas na década de 1990, pela grande demanda de aplicações. Segundo o mesmo autor, com a difusão da internet, houve um crescimento bastante acelerado na quantidade de informação, principalmente devido a facilidade de inserir novos documentos neste meio de comunicação (PINHEIRO, 2011). Estes documentos textuais têm sido os principais alvos de máquinas de busca e outras ferramentas de recuperação de informação, que executam tarefas como busca por documentos potencialmente relevantes e filtragem de textos com base em alguns conteúdos específicos (FIGUEIREDO, 2008).

Dharmadhikari et al. (2011) citado por Ceci et al (2014) explica a classificação de documentos como podendo ser supervisionada, ou seja, sendo treinada a partir de interações e validadas por um especialista, ou sendo não supervisionada, realizada de forma automática sem a necessidade de intervenção humana.

Sebastiani (SEBASTIANI, 2002 apud PINHEIRO, 2011), explica TC como a tarefa de designar um valor booleano (*true* ou *false*) ao par $\langle d_i, c_j \rangle \in \mathscr{D} \times \mathscr{C}$, no qual $\mathscr{D}$ é o conjunto de todos os documentos e $\mathscr{C} = \{c_1,..., c_N\}$ é o conjunto pré-definido de *N* categorias. Pinheiro (2011, p.18) explica: “um valor *true* em $\langle d_i, c_j \rangle$ indica que o documento d_i pertence à classe c_j, enquanto um valor *false* indica que o documento d_i não pertence à classe c_j”.

2.2.3 Anotações semânticas

Anotação semântica, ou *semantic annotation*, que consiste na utilização de metadados associados a um conteúdo (BASTOS, 2015), normalmente para documentos, trechos de texto ou conceitos, por meio de criação de rótulos (CALEGARI, 2016) tem sido proposta como uma forma para expressar a semântica da informação (BASTOS, 2015), enriquecendo o conteúdo do documento e permitindo o processamento semântico do mesmo (ARANTES, 2010), facilitando busca, recuperação, entendimento e uso de informação (BASTOS, 2015), bem como permitindo a busca avançada, baseada em conceitos,

interferências sobre conteúdos e visualização de informação baseadas em ontologias (CALEGARI, 2016).

Para Bastos (2015) a anotação semântica surgiu devido as limitações encontradas para extrair informações a partir de conteúdo de páginas web:

> As páginas web foram originalmente projetadas para permitir que navegadores (browsers) apresentassem informações a humanos. As primeiras iniciativas de leitura do conteúdo de páginas web por máquinas foram baseadas em aspectos sintáticos. Porém, devido ao crescimento da quantidade de conteúdo disponibilizado na web, o uso de mecanismos de buscas baseados em aspectos sintáticos do conteúdo passou a apresentar problemas (BASTOS, 2015, p. 16)

Os modos de utilização de anotação semântica dividem-se em três categorias, sendo elas a manual, automática e semiautomática (OREN et al, 2006 apud CECI et al, 2014, SLIMANI, 2013 apud CALEGARI, 2016). A anotação semântica manual é o processo utilizado para transformar recursos sintáticos, como texto puro, em estruturas complexas por meio de adição de metadados (CALEGARI, 2016) e pode ser realizada por uma ou mais pessoas (CECI et al, 2014). As anotações automáticas por sua vez, criam as anotações sem intervenção manual (CECI et al, 2014), e necessitam de utilização de técnicas de inteligência artificial, como aprendizado de máquina, para que isso aconteça (CALEGARI, 2016). Já as anotações semiautomáticas são realizadas por pessoas, com a assistência de ferramentas que fazem sugestões automáticas (CECI et al, 2014).

O uso de anotações semânticas em documentos renderizados por ferramentas *desktop* é chamado documentação semântica (BASTOS, 2015). A adição de metadados nesses documentos resulta em documentos semânticos, que podem ser vistos como documentos “inteligentes”, uma vez que conhecem o seu conteúdo e permitem que processos automatizados saibam o que fazer com ele (UREN et al., 2006 apud BASTOS, 2015).

Arantes (2010) enfatiza o uso de ferramentas para apoiar o processo de anotação semântica como sendo fundamental, uma vez que a notação por via de edição de estrutura de documento é enfadonha, e pode levar a erros. O autor explica:

> Para criar uma anotação semântica em sua forma mais básica, um usuário deve editar o conteúdo do documento, modificando sua estrutura, de forma que a anotação não seja parte do conteúdo visível para leitura humana, mas ao mesmo tempo seja possível de processamento por máquinas. já que o ato de editar manualmente a estrutura de um documento é uma tarefa que pode levar a erros, graves, podendo até comprometer a renderização gráfica do documento, algumas ferramentas proveem interfaces gráficas para o gerenciamento de metadados (ARANTES, 2010, p. 30).

O acesso ao conteúdo de documentos depende de intervenção humana, uma vez que os documentos foram originalmente criados para seu conteúdo ser entendido por humanos

e não por computadores (BASTOS, 2015). Para contornar esse problema, pode-se utilizar a documentação semântica, possibilitando aos computadores interpretarem o conteúdo de documentos *desktop* através da adição de metadados baseados em ontologias (BASTOS, 2015), que será contextualizado na próxima seção.

2.2.4 Ontologias

Uma ontologia (do inglês, *ontology*) é uma especificação formal e explícita de uma conceituação compartilhada, ou seja, um modelo abstrato que representa um fenômeno no mundo real (GRUBER, 1993 apud BASTOS, 2015) sendo capazes de prover uma estrutura semântica formal rica que inclui conceitos, relações e restrições (BASTOS, 2015). O termo ontologia foi originalmente utilizado pelo ramo da metafísica, ciência que procura explicar a fundamental natureza das coisas, particularmente o relacionamento entre a mente e a matéria (CECI, 2015).

Arantes (2010) explica que, de acordo com a filosofia clássica, ontologia é o estudo dos tipos de coisas que existem. Ceci (2015) aponta que na visão da filosofia, as ontologias procuram estudar visões de mundo a fim de categorizar elementos. Já Calegari (2016) explica que a palavra ontologia formada pelos termos de origem grega "ontos", que significa "ser" e "logia", que significa "estudo" é definida na filosofia como o estudo do ser e da realidade.

No contexto de inteligência artificial, e de algumas outras áreas da ciência da computação, o termo é normalmente usado com duas finalidades (CHANDRASEKARAN, 1999 apud ARANTES, 2010):

- Representação de vocabulário comumente especializado para algum domínio;
- Referência a um corpo de conhecimento que descreve algum domínio.

Ceci et al (2014, p. 5-6) explica:

> O uso de uma ontologia possibilita definir conceitos e suas relações representando o conhecimento sobre um documento em termos específicos de um domínio. Para representar o conteúdo de um documento explicitamente, é necessário criar links (associação) entre o documento e partes relevantes de um modelo de domínio, ou

> seja, associar aqueles elementos do modelo de domínio que são relevantes ao conteúdo do documento.

Guarino (1998) citado por Arantes (2010), Bastos (2015) e Ceci (2015) define quatro tipos de ontologia:

- Ontologias de alto nível (*top-level ontology*): também conceituadas como como ontologias genéricas ou ontologias gerais, possuem definições abstratas para a compreensão de aspectos do mundo como, por exemplo, processos, espaços, tempo, coisas, entre outros.
- Ontologias de tarefa (*task ontology*): tratam de tarefas genéricas ou de atividades, como diagnosticar ou vender.
- Ontologias de domínio (*domain ontology*): dedicam-se a um domínio específico de uma área genérica como, por exemplo, medicina, direito, entre outros.
- Ontologias de aplicação (*application ontology*): descrevem conceitos dependentes de um domínio a com o objetivo de solucionar um problema, os quais são, normalmente, especialização de ontologias relacionadas.

Freitas (2003) citado por Ceci (2015) complementa os mesmos quatro tipos de ontologia de Guarino (1998) apresentando mais dois tipos:

- Ontologias de representação: definem as primitivas de representação, tais como frames, atributos, axiomas, entre outros, na forma declarativa.
- Ontologias centrais: definem os ramos de estudo de uma área ou conceitos mais abstratos dessa área.

Guizzardi (2005 apud CELEGARI, 2016) aponta que as ontologias, diferentemente de outras disciplinas, cientificas, mais especificas, como biologia, física e química, que tratam das entidades em seus respectivos domínios, lida com as possíveis relações transdisciplinares entre conceitos pertencentes a diferentes domínios da ciência, além de entidades reconhecidas pro senso comum.

De acordo com Ceci (2015), as ontologias podem simplificar a classificação das sentenças num processo de análise de sentimento, uma vez que em suas classes existem tipos de sentimentos e emoções, trazendo atributos para facilitar a polarização.

2.3 PROCESSAMENTO DE LINGUAGEM NATURAL

O processamento de linguagem natural (PLN), do inglês, *natural language processing* (NLP) aborda a sutileza das técnicas para extrair dados estruturados de dados não estruturados (BALAGE FILHO, 2017) e, apesar de nosso conhecimento e entendimento serem um pouco limitados, devido a complexidade da tarefa de processamento de linguagem natural onde não existem problemas fáceis (LIU, 2012) a área é extremamente desenvolvida, e propõe diversas técnicas para manipulação e transformação de documentos (BLAZ, 2017).

Segundo Avila (2017, p. 13), a área de processamento de linguagem natural é "a área da computação que explora a questão de como computadores podem ser utilizados para entender e manipular textos e áudios em linguagem natural, como inglês, português e espanhol". O processo de trabalho com a linguagem natural está dividido em três níveis distintos, sendo eles o morfológico, o sintático e o semântico (JURAFSKY; MARTIN, 2000 apud BLAZ, 2017, CUNHA; CINTRA, 2009 apud AVILA, 2017).

Na presente seção serão apresentadas as descrições dos níveis do processamento de linguagem natural, como a morfológica, sintática e semântica, bem como métodos e áreas relacionados, como a extração de informação e o entendimento de linguagem natural.

2.3.1 Análise sintática

A área de análise sintática (do inglês, *parsing*), é um método analítico pelo qual se reconhecem as funções sintáticas das expressões (BORGES NETO; MERCER, 1993) e é considerada como sendo o processo de encontrar uma sequência de derivações que nos leva a concluir que uma determinada frase faz parte de uma linguagem ou não, com base em um conjunto de regas especificadas por uma gramática (LINZ, 2012 apud ROSA, 2019).

A análise sintática é a responsável por decompor um período qualquer em orações e, simultaneamente, classificar essas orações, reconhecendo-lhes funções ou papéis sintáticos (BORGES NETO; MERCER, 1993), e é considerada por muitos como um problema já solucionado, apesar de que, devido a implementação de alguns *parsers* serem complexas e de

difícil manutenção, também pode ser considerado um problema não solucionado (ROSA, 2019).

Rosa (2019) descreve que os softwares *parsers* são muito mais comuns do que normalmente nos damos conta, pois estão presentes nas mais diversas áreas, como em processamento de linhas de comando, no processamento de linguagens naturais, ou ainda na composição de linguagens.

A análise sintática trata do conhecimento das relações ou associações entre as palavras em uma frase (BLAZ, 2017, AVILA, 2017). Avila (2017) também enfatiza que a análise examina funções das palavras, e relações de interdependência, desconsiderando certas classes com palavras com poucos significados, conhecidas como *stopwords*.

As *stopwords*, ou no português também conhecidas como *palavras vazias*, são palavras que tem baixo poder de discriminação, e são filtradas antes de processar o texto, como por exemplo, “a”, “é”, “que”, entre outros (SILVA, 2016).

2.3.2 Análise semântica

A análise semântica “preocupa-se com o significado e a organização dos significados das palavras, termos ambíguos, etc.” (AVILA, 2017, p. 14). Allen (1995) citado por Wilkens (2016) explica a análise semântica concentrando-se no que as palavras significam e como esses significados se combinam para formar outros significados.

2.3.3 Análise morfológica

A análise morfológica trata do conhecimento das estruturas e dos componentes das palavras, individualmente (BLAZ, 2017). Oliveira (2015, p. 2), em seu artigo *A relação da consciência morfológica com o processamento morfológico e a leitura*, explica:

> [...] uma vez que o conhecimento das estruturas morfológicas das palavras pode servir tanto para compreensão de palavras novas quanto para precisão na hora de ler ou escrever uma palavra. Por exemplo, mesmo um leitor que até então nunca havia

> visto ou ouvido a palavra *reexportador*, poderia facilmente entender, pela análise dos morfemas, que se trata de alguém que exporta alguma coisa novamente.

A próxima seção apresenta a área de extração de informação.

2.3.4 Extração de informação

A área de extração de informação (do inglês, *Information Extraction*) é uma parte do processamento de linguagem natural, e tem por objetivo aprender relações semânticas de textos (XAVIER, 2014). A motivação principal inicialmente da área de extração de informação era povoar automaticamente base de dados (JACABOS; RAU, 1993 apud BATRES et al, 2005), todavia, de acordo com Batres et al. (2005, p. 74), o sistema de extração de informações "[...] também permitem melhorar o desempenho de sistemas de recuperação de informação através de integração e sintetização da informação, evitando desta forma a ocorrência de redundâncias em textos que tratam do mesmo assunto". Os autores também complementam:

> Cada dia que passa, mais e mais informações surgem nos meios eletrônicos e uma porcentagem significativa das mesmas é composta de informações que, de alguma forma, podem ser estruturadas ou inter-relacionadas. Assim, ao invés de encontrar textos que contenham informações e permitir ao usuário procurar o que lhe interessa, esta nova área passou a se preocupar em encontrar as informações dentro dos textos e tratá-las de forma a apresentar algum tipo de conhecimento novo e útil para o usuário. A idéia é aproveitar todo o conhecimento humano que há em textos escritos e, mesmo que tal conhecimento novo não seja resposta direta às indagações do usuário, processá-lo sob a premissa de que ele deve contribuir na satisfação das necessidades de informação do mesmo (BATRES et al, 2005, p.74-75).

Os métodos de extração de informação podem ser empregados em aplicativos que usam modelos de representação de conhecimento que descrevem relações entre palavras, como ontologias ou redes semânticas (XAVIER, 2014), envolvendo a identificação de padrões que representam um contexto chave dentro de um texto (BATRES et al, 2005) e utilizando um conjunto de filtros que, juntamente com os padrões, apresentarão de forma estruturada a informação contida, possibilitando a atualização de uma base de dados ou melhora de recuperação posterior (JACOBS; RAU, 1993 apud BATRES et al, 2005).

Uma outra abordagem de extração de informação é conhecida como o paradigma de extração aberta de informação (do inglês, *Open Information Extraction)*, de extração de

relações, trabalhando com a identificação de relações não definidas previamente, buscando superar as limitações impostas pelos métodos tradicionais de extração de informações como a dependência de domínio e a difícil escalabilidade (XAVIER, 2014).

2.3.5 Entendimento de linguagem natural

A área de entendimento de linguagem natural (do inglês, *natural language understanding,* ou *NLU*) é um tipo de processamento de linguagem natural, que "pode ser considerada como o processo de tradução da linguagem natural para uma linguagem interpretável por um computador" (WILKENS, 2016, p. 16, tradução nossa).

Segundo Garcia (2017), na área de processamento de linguagem natural, as redes neurais são atualmente o modelo mais popular e são capazes de obter resultados precisos em áreas como reconhecimento de imagem e voz e extração de informações, muitas vezes com desempenho ainda melhor do que os humanos, todavia, as tarefas na área de entendimento de linguagem natural ainda apresentam baixa acurácia em comparação com humanos.

Nos últimos anos, tem havido um esforço crescente na compreensão de textos com redes neurais e técnicas de aprendizado de máquina que são capazes de lidar com grandes blocos de informação, resolvendo vários problemas de linguagem natural do mundo real (GARCIA, 2017), tais como as principais técnicas descritas por Bates (1994) citado por Wilkens (2016) sendo elas a baseada em estatísticas (do inglês, *statistical based*), a correspondência de padrões (do inglês, *pattern matching*), a análise sintaticamente orientada (do inglês, *syntactically driven parsing*), a gramáticas semânticas (do inglês, *semantic grammars*) e a instanciação do case frame (do inglês, *case frame instantiation*), no entanto, apesar do desenvolvimento da área como um todo, os problemas de NLU que requerem estruturas bem definidas como respostas a perguntas, resumos e diálogos automatizados ainda não foram resolvidos, devido a natureza estocástica das técnicas de redes neurais e aprendizagem de máquina, que são impedidas de expressar estruturas bem definidas em seus modelos internos (GARCIA, 2017).

Wilkens (2016) explica as principais técnicas de tradução para linguagem de máquina de Bates (1994):

- Baseado em estatística: a proposta básica da abordagem estatística é que os termos que ocorrem em contextos semelhantes carreguem informações semânticas de forma semelhante. Assim, abordagens como a Análise Semântica Latente (LSA), modelos Naive Bayes e Markov calculam a concorrência dos termos nos textos.
- Correspondência de padrões: Nesta abordagem, a interpretação é obtida por correspondência de padrões de palavras. Associado a cada padrão está uma interpretação e, no caso mais simples, esse arranjo é simplesmente uma lista de classes de equivalência de expressões e interpretações. Em variações mais sofisticadas desta abordagem, a correspondência pode envolver componentes de nível superior ou elementos de semântica (como rotulagem por ontologias e reconhecimento de entidade), portanto, alguns aspectos da interpretação podem ser construídos, mas os parâmetros de abordagem permanecem tão diretamente ligados quanto possível à entrada;
- Análise sintaticamente orientada: a sintaxe fornece maneiras de combinar palavras para formar unidades de nível superior, como frases e sentenças. A análise sintaticamente orientada é naturalmente construtiva de forma que, por exemplo, a interpretação de um grande grupo de palavras é construída a partir das interpretações de suas partes sintáticas. Nesse sentido, é o oposto da correspondência de padrões. A maneira usual de operar é construindo uma análise sintática completa da frase e então construir uma representação interna.
- Gramáticas semânticas: a análise da linguagem baseada em gramáticas semânticas é semelhante à análise baseada na sintaxe, exceto que permite definições semânticas e sintáticas. Assim, a categoria “sintagma nominal” em uma gramática sintática teria uma especificação semântica adicional.
- Instanciação do case frame: consiste em conceitos-chave (conceitos principais) e um conjunto de papéis (conceitos secundários) associados de forma bem definida ao conceito principal. Inicialmente, a cabeça é composta por um verbo principal e o caso inclui o “agente” (que executa a ação), o objeto (que sofre a ação), o local (onde a ação ocorre) e assim por diante.

Após a definição dos conceitos das técnicas e vertentes da análise de sentimento, será apresentada a metodologia de pesquisa no capítulo seguinte.

3 METODOLOGIA DE PESQUISA

Este capítulo apresenta a descrição da metodologia de pesquisa do presente trabalho, caracterizando seu tipo de pesquisa, suas atividades metodológicas e as suas delimitações.

3.1 CARACTERIZAÇÃO DO TIPO DE PESQUISA

Este trabalho caracteriza-se, de vista da sua natureza, como uma pesquisa aplicada. Segundo Silva e Menezes (2005) a pesquisa aplicada objetiva gerar conhecimentos para aplicação prática e dirigidos à solução de problemas específicos, envolvendo verdades e interesses locais.

Como contextualizado na problemática, nos objetivos gerais e objetivos específicos nas seções anteriores, esta pesquisa tem por finalidade o desenvolvimento e proposta de uma solução, bem como a avaliação do protótipo funcional. Como definido no último objetivo específico, as avaliações abordam o classificador de texto e as interações com o usuário, encaixando-se respectivamente nas abordagens de pesquisa quantitativa, ou seja, que consideram tudo que pode ser quantificável, traduzindo as opiniões e informações em números para classifica-las (SILVA; MENEZES, 2005), e qualitativa, que dependem de muitos fatores, tais como a natureza dos dados coletados, e que podem ser definidos como um processo com uma sequência de atividades, que envolvem a redução dos dados, a categorização desses dados e sua interpretação (GIL, 2002).

Do ponto de vista dos objetivos da pesquisa, classifica-se este trabalho como uma pesquisa exploratória. Gil (2002, p. 41) explica:

> Estas pesquisas têm como objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a constituir hipóteses. Pode-se dizer que estas pesquisas têm como objetivo principal o aprimoramento de idéias ou a descoberta de intuições. Seu planejamento é, portanto, bastante flexível, de modo que possibilite a consideração dos mais variados aspectos relativos ao fato estudado.

Para o mesmo autor, na maioria dos casos nas pesquisas exploratórias, assume-se a forma de pesquisa bibliográfica ou estudos de caso.

Para Gil (2002) pode-se definir a pesquisa como o procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos. Para iniciar a realização de uma pesquisa, Chizzotti (2010) citado por Oliveira (2013) aconselha ao pesquisador que inicie seu trabalho com a busca de fontes de informação relacionadas ao assunto a ser estudado.

A partir de diversas fontes e referências estudadas ao longo de todo o trabalho, esta pesquisa possui como procedimento técnico principal a pesquisa bibliográfica. Gil (2002, p. 44) explica:

> A pesquisa bibliográfica é desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos. Embora em quase todos os estudos seja exigido algum tipo de trabalho dessa natureza, há pesquisas desenvolvidas exclusivamente a partir de fontes bibliográficas. Boa parte dos estudos exploratórios pode ser definida como pesquisas bibliográficas. As pesquisas sobre ideologias, bem como aquelas que se propõem à análise das diversas posições acerca de um problema, também costumam ser desenvolvidas quase exclusivamente mediante fontes bibliográficas.

O mesmo autor descreve a pesquisa bibliográfica como um processo que envolve as seguintes etapas (GIL, 2002):

a) Escolha do tema;

b) Levantamento bibliográfico preliminar;

c) Formulação do problema;

d) Elaboração do plano provisório de assunto;

e) Busca das fontes;

f) Leitura do material;

g) Fichamento;

h) Organização lógica do assunto; e

i) Redação do texto.

Após a definição e caracterização do tipo de pesquisa, será apresentada as atividades metodológicas desenvolvidas para o trabalho na próxima seção, bem como as suas delimitações na seção seguinte.

3.2 ATIVIDADES METODOLOGICAS

Esta seção descreve as atividades metodológicas que deverão ser cumpridas para o desenvolvimento da solução, sendo elas:

- Levantamento de requisitos;
- Escolha de ferramentas;
- Modelagem de software;
- Desenvolvimento do software;
- Testes do software;
- Avaliação do software.

3.3 DELIMITAÇÕES

Esta seção apresenta as delimitações do presente trabalho, ou seja, as partes que não serão realizadas ou não serão desenvolvidas, sendo elas:

- Por se tratar de um protótipo, o software desenvolvido não será testado em ambiente real, apenas em ambiente local.
- Será desenvolvido uma amostra de marcadores limitada para a execução do software, e qualquer técnica de atualização automática ou aprendizado automático não será desenvolvida;

4 PROPOSTA DE SOLUÇÃO

Neste capítulo são apresentadas as definições da metodologia de desenvolvimento ICONIX, bem como os componentes para o desenvolvimento das especificações de sistema apontados pela primeira fase da metodologia, tais como os requisitos, os casos de uso e prototipação de tela.

Também são apresentados recursos para o melhor entendimento da proposta de solução, sendo eles usados pelo ICONIX ou não, como a linguagem de modelagem UML, os diagramas de caso de uso, os diagramas de sequência e os diagramas de atividades.

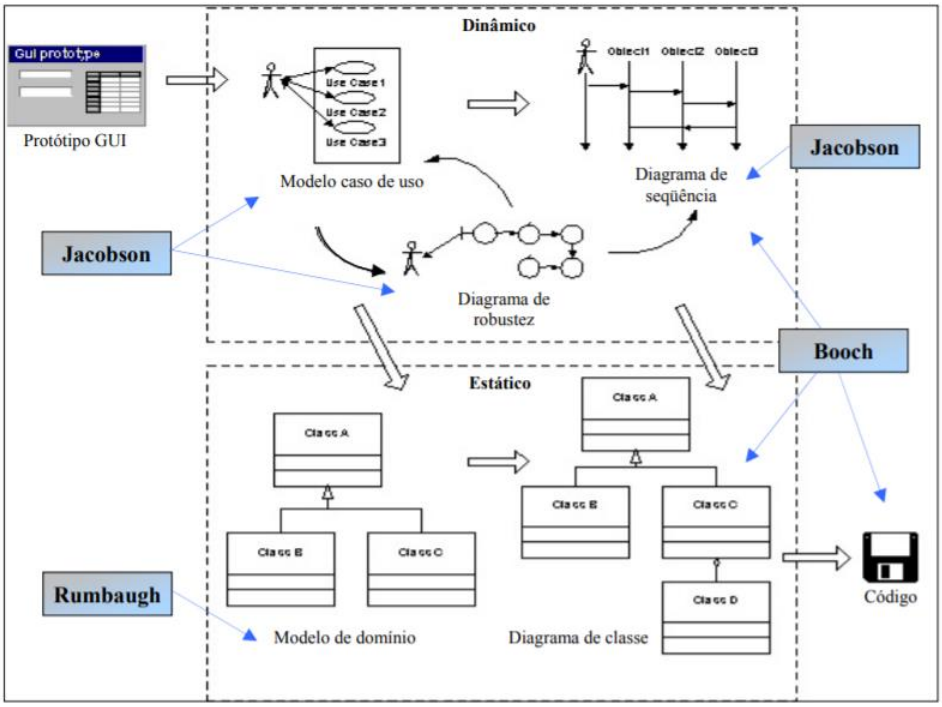
Como já apontado, esse trabalho utiliza apenas a primeira fase da metodologia ICONIX para o desenvolvimento das suas especificações.

4.1 ICONIX

A metodologia ICONIX começou a ser desenvolvida em 1993, com o objetivo de mesclar os melhores aspectos das três mais famosas metodologias orientada a objetos vigentes na época, que posteriormente formaram a base da UML (ROSENBERG; STEPHENS, 2007 apud SOUSA, 2013), conhecidos como a técnica de modelagem de objeto de Jim Rumbaugh, o método de Ivar Jacobson e o método de Booch de Grady Booch (ROSENBERG et al., 2005 apud PAULINO, 2014).

A partir da síntese do processado unificado pelos “três amigos” - Booch, Rumbaugh, e Jacobson - foi elaborado por Doug Rosenberg e Kendall Scott (1999) citados por Bona (2002) um diagrama do processo ICONIX, mostrando a contribuição dos autores no processo, na figura 1.

Figura 1 – Processo ICONIX, mostrando a contribuição dos “três amigos”

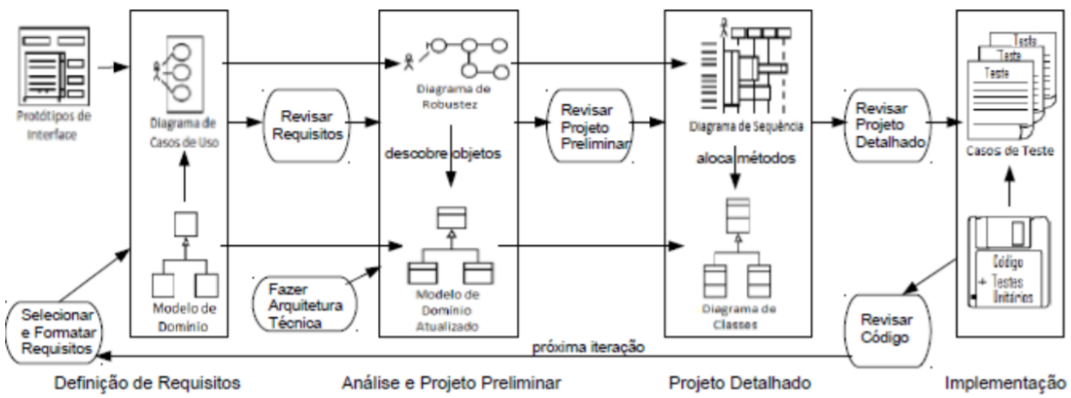


Fonte: Doug Rosenberg e Kendall Scott (1999) citados por Bona (2002), p. 60

Para Paulino (2014), o processo ICONIX é preocupado com a análise e os aspectos de modelagem de projeto de produção de software, tendo como missão a análise de requisitos e o desenho limpo de software.

Bona (2002) explica o processo como sendo simplificado e que unifica conjuntos de métodos de orientação a objetos em uma abordagem completa, com o objetivo de dar cobertura ao ciclo de vida. Sousa (2013) demonstra o processo ICONIX na figura 2:

Figura 2 – Processo ICONIX



Fonte: Sousa 2013, p. 34 adaptado de Rosenberg e Stephens, 2007

Como se pode notar no diagrama na figura 2, o processo ICONIX é “divido em um fluxo dinâmico, para representar os aspectos comportamentais do software, e outro fluxo estático, para expressar os aspectos estruturais do software” (SOUSA, 2013, p. 34). O autor também destaca as principais características do processo:

- Utiliza um subconjunto da UML: apenas 3 diagramas (casos de uso, classes e sequência), mais o de robustez, ao invés dos 14 diagramas da UML (conforme previsto na versão 2.5) existentes;
- Minimiza a paralisia da análise: ignora a semântica de estereótipos do padrão UML tais como *<<extend>>*, *<<include>>*, etc. que faz o desenvolvedores perder tempo, retardando a passagem da análise para o projeto;
- Possui rastreamento da análise à implementação: todos os requisitos são associados a casos de uso e classes, que formam o eixo de sustentação do processo;
- É iterativo e incremental: várias iterações ocorrem entre o desenvolvimento do Modelo de Domínio e a modelagem dos casos de uso, enquanto o modelo estático é incrementalmente refinado pelo modelo dinâmico;
- É baseado nas questões fundamentais da orientação a objetos: o que fazem os usuários? (casos de uso); quais os objetos do mundo real? (Modelo de Domínio); quais os objetos relacionados com os casos de usos? (robustez); como os objetos colaboram entre si? (sequência); como realmente será construído o software? (classes).

A próxima seção apresenta a UML, anteriormente citada.

4.2 UML

A linguagem de modelagem unificada, ou Unified Modeling Language (UML), é uma linguagem visual constituída elementos gráficos onde cada elemento gráfico possui uma sintaxe e uma semântica que definem o significado e a utilização de um elemento (PAULINO, 2014), que serve para especificar e documentar os artefatos de um sistema

intensivo de software (GUERRA, 2012) destacando-se por apresentar uma notação visual e padronizada, ainda que a semântica seja imprecisa em várias de suas definições (SOUSA, 2013), e será usado nesse capítulo, nos diagramas que serão apresentados em futuras seções.

Segundo Sousa (2013), devido à grande expressividade de seus diagramas, é possível representar tanto aspectos estruturais quanto comportamentais de um software em diferentes níveis de abstração, o que auxilia os sucessivos refinamentos informais da especificação de requisitos até o código. Guerra (2012, p.6, tradução nossa) explica:

> Essas linguagens permitem especificar uma ampla variedade de aspectos de um sistema, desde a estrutura estática até o comportamento dinâmico. A estrutura pode ser descrita com elementos de modelo estáticos, como classes, relationships, nodes e componentes. O comportamento descreve como os elementos dentro da estrutura interagem ao longo do tempo. Além disso, qualquer linguagem UML pode ser estendida por seus próprios mecanismos de extensão para definir modelos específicos de domínio.

A UML permite representar os conceitos do paradigma da orientação a objetos (PAULINO, 2014), e foi escolhida como linguagem para expressar os modelos que representam o software (SOUSA, 2013), sendo usada para modelagem de qualquer sistema, não importando a linguagem utilizada e o processo de desenvolvimento adotado (BEZERRA, 2002 apud PAULINO 2014). Este trabalho utiliza como base a versão 2.5 da UML.

4.3 REQUISITOS

Esta seção apresenta os requisitos gerais do sistema, que de acordo com Paulino (2014, p. 61), “devem prever toda a expectativa do usuário, e os pontos que devem ser implementados pelo sistema”.

De acordo com Sousa (2013) a elicitação e a especificação de requisitos saem ligeiramente do escopo do ICONIX, mas ainda assim ele fornece um guia geral para a realização dessa atividade:

1. Não expresse os requisitos em um estilo muito técnico;
2. Não tema dar exemplos para melhorar o entendimento de um requisito;
3. Crie estimativas dos cenários de caso de uso, não dos requisitos funcionais;
4. Evite a síndrome do documento grande e único;
5. Faça a distinção entre os diferentes tipos de requisitos;

6. Trate os requisitos como cidadãos de primeira classe no modelo;
7. Escreva pelo menos um caso de teste para cada requisito;
8. Não inclua detalhes funcionais na especificação de casos de uso;
9. Mapeie os requisitos em casos de uso;
10. Use ferramentas CASE para o rastreamento de requisitos em casos de uso.

Os requisitos aqui pontuados estão distribuídos de acordo com os apresentados como os mais usados por Borillo (2002, apud BONA, 2002), sendo eles os requisitos funcionais, onde definimos o comportamento do sistema (PAULINO, 2014), os requisitos não funcionais, que permitem verificar níveis de qualidade e segurança e detalhamento da tecnologia que deve ser adotada (PAULINO, 2014) e as restrições, que fixam os limites do projeto e do sistema (BORILLO, 2002 apud BONA, 2002) e que são apresentadas nesse trabalho como regras de negócio.

4.3.1 Requisitos funcionais

Os requisitos funcionais definem o que o sistema deve ser capaz de fazer (SOUSA, 2013), permitindo ao desenvolvedor verificar se o que foi construído no sistema atende ao que foi requisitado pelos usuários (PAULINO, 2014), e definidos pelo presente trabalho no quadro 1:

Quadro 1 – Requisitos funcionais

Requisito	**Descrição**
RF001	O sistema deve possibilitar qualquer ação sem uma conta ou *login*.
RF002	O sistema deve permitir e entrada de mensagens
RF003	O sistema deve qualificar as mensagens entradas
RF004	O sistema deve responder as mensagens entradas
RF005	O sistema deve representar visualmente o teor das entradas
RF006	O sistema deve representar visualmente as mensagens enviadas
RF007	O sistema deve representar visualmente as respostas processadas

Fonte: elaborado pelo autor, 2021.

A próxima seção dá continuação a descrição dos requisitos do sistema, apontando e descrevendo os requisitos não funcionais da aplicação.

4.3.2 Requisitos não funcionais

Os requisitos não-funcionais representam as qualidades do produto, podendo incluir requisitos de desempenho, de capacidade, de teste e de segurança, e geralmente estão associados a critérios que servem como parâmetro para quantificar o requisito (BORILLO, 2002 apud BONA 2002).

O presente trabalho pontua os seguintes requisitos não funcionais para serem considerados no processo de desenvolvimento (Quadro 2):

Quadro 2 – Requisitos não funcionais

Requisito	**Descrição**
RNF001	O cliente deverá ser escrito usando o framework angular versão 7 ou superior.
RNF002	As api's dos servidores deverão ser escritas em python na versão 2.7 ou superior.
RNF003	O sistema deverá funcionar nos navegadores Google Chrome, Mozilla Firefox, Microsoft edge e Internet explorer 11.
RNF004	O sistema deverá ter uma interface voltada aos princípios de Flat design.
RNF005	A comunicação entre as aplicações deverá se basear no padrão RESTFUL.

Fonte: elaborado pelo autor, 2021.

A próxima seção finaliza a descrição dos requisitos do sistema, apresentando as restrições, também conhecidas como regras de negócios.

4.3.3 Regras de negócio

As regras de negócios garantem a estrutura ou influenciam o comportamento de um negócio (MORGADO et al, 2007). As regras de negócios são limitações ou condições restritivas, também conhecidas como restrições ou *constraints*. Borillo (2002 apud BONA 2002) aponta as restrições como fixando os limites do projeto e do sistema, normalmente associadas a uma razão por que a restrição está limitando o sistema. Pode-se constatar as limitações a partir do exemplo (figura 3):

Figura 3 – Parte do exemplo de um artefato especificação de requisitos formatado

2. Restrições

C1: Um usuário só pode comprar um livro se estiver logado no sistema.

[Descrever as restrições (regras de negócios e propriedades funcionais) sequencialmente, tentando evitar ambigüidades]

Fonte: adaptado de Sousa, 2013, p.118, tradução nossa

Segundo Morgado et al (2007, p. 385):

> A documentação e a formalização das regras de negócio constituem um importante ativo estrutural e intelectual para a organização, pois, desta forma, as regras de negócio podem ser mais facilmente divulgadas aos profissionais envolvidos, como também são aumentados o entendimento e tratamento uniforme e consistente do negócio por esses profissionais.

O presente trabalho pontua as seguintes regras de negócio para serem considerados no processo de desenvolvimento (Quadro 3):

Quadro 3 – Regras de negócio

Regra	**Descrição**
RN001	Se um usuário enviar duas frases simultaneamente elas deverão ser tratadas separadamente.

Fonte: elaborado pelo autor, 2021.

Após a finalização dos requisitos e regras de negócio, a próxima seção apresenta os protótipos de tela.

4.4 PROTÓTIPOS DE TELA

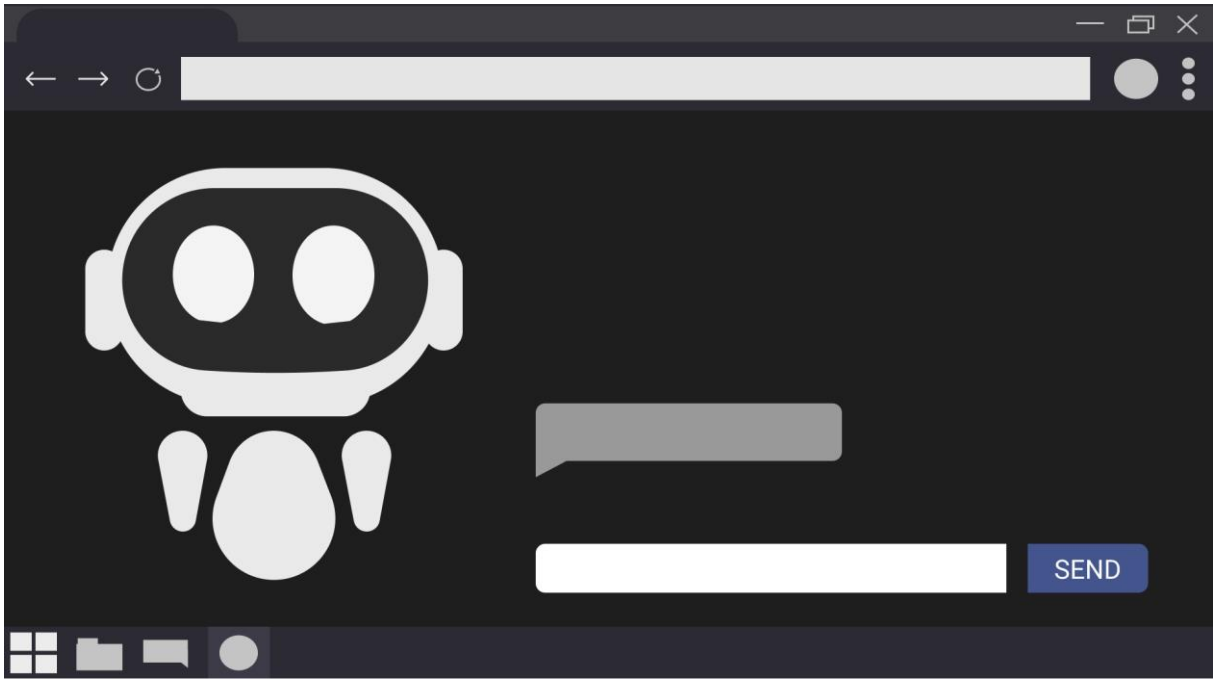
A camada de interface do usuário permite a interação entre o sistema e o usuário (GUERRA, 2012). O protótipo das telas, ou o protótipo da interface do usuário, devem ter um custo reduzido e rápida obtenção, para que possa ser avaliado (BONA, 2002), e permitir que o desenvolvedor consiga elaborar um visual próximo ao esperado pelos usuários (PAULINO, 2014). De acordo com Pascoal (2001) citado por Bona (2002), a prototipação é um processo que permite ao desenvolvedor a criação de um modelo do software que deverá ser construído podendo ter diferentes formas, como:

- Uma no papel ou em um modelo baseado em computador, no qual a interação homem-máquina é desenhada para permitir ao usuário entender como tal interação irá ocorrer;
- Uma versão funcional que contém apenas um subconjunto das funcionalidades requeridas no software desejado;
- Um protótipo em execução, o qual executa parte ou todas a funções desejadas, mas tem outras características que serão melhoradas no desenvolvimento.

Os protótipos desenvolvidos para guiar o desenvolvimento do presente trabalho abordam a tela e os componentes visuais da aplicação. Os protótipos representam a mesma tela e variações das respostas após o input de uma mensagem.

A figura 4 representa o estado inicial da aplicação, quando um usuário entra na tela sem ter feito qualquer tipo de ação:

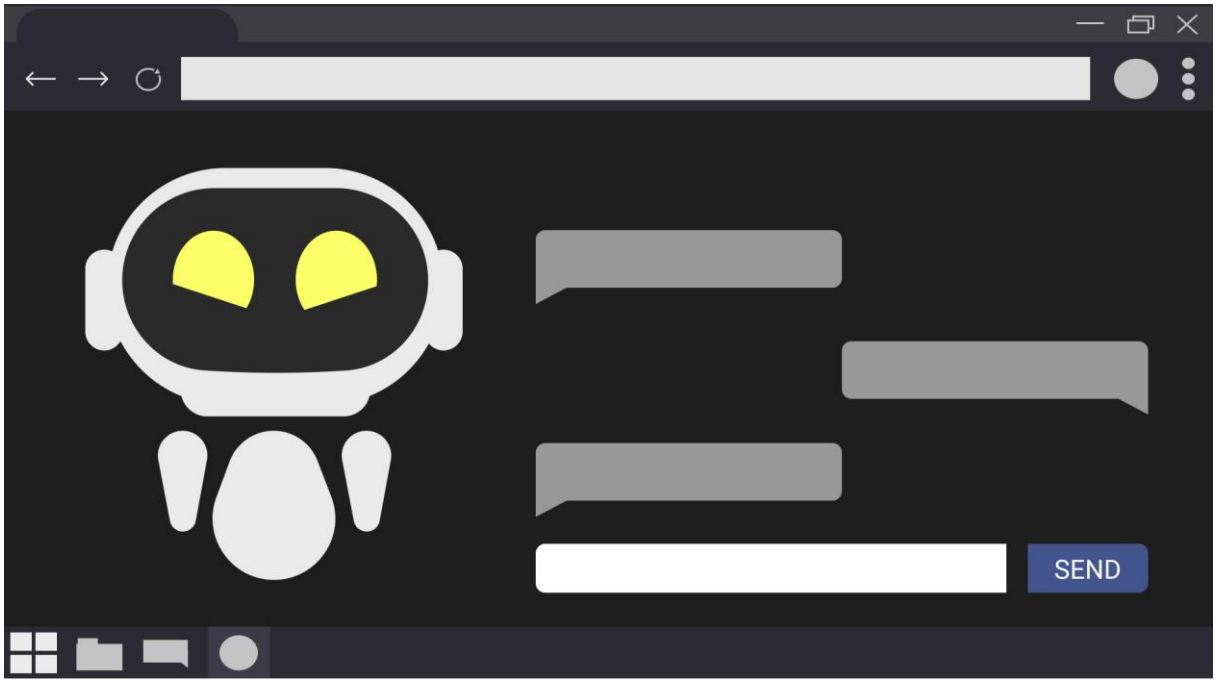
Figura 4 – Protótipo de tela estado inicial da aplicação



Fonte: Elaborado pelo autor, 2021

As figuras 5, 6 e 7 referem-se a resposta após o envio, processamento e resposta descritos no caso de uso UC002 (que será apresentado na próxima seção). A figura 5 representa uma resposta com alegria:

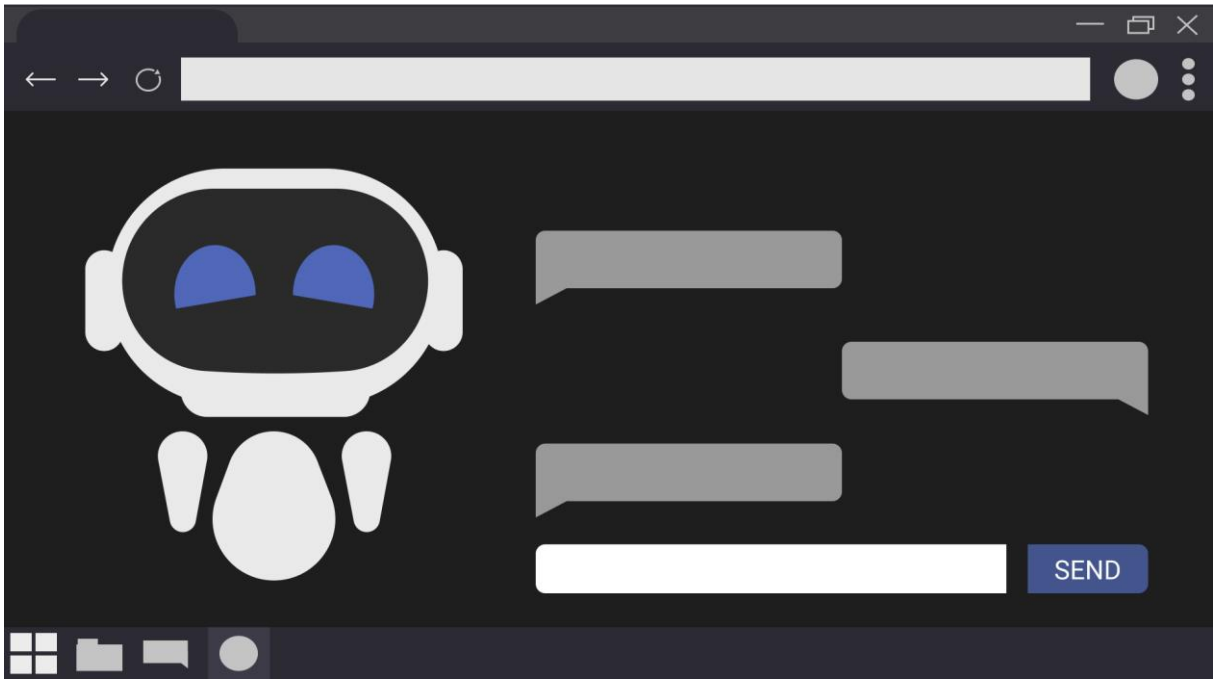
Figura 5 – Protótipo de tela com uma resposta com o sentimento de alegria



Fonte: Elaborado pelo autor, 2021

A figura 6 representa uma resposta com tristeza:

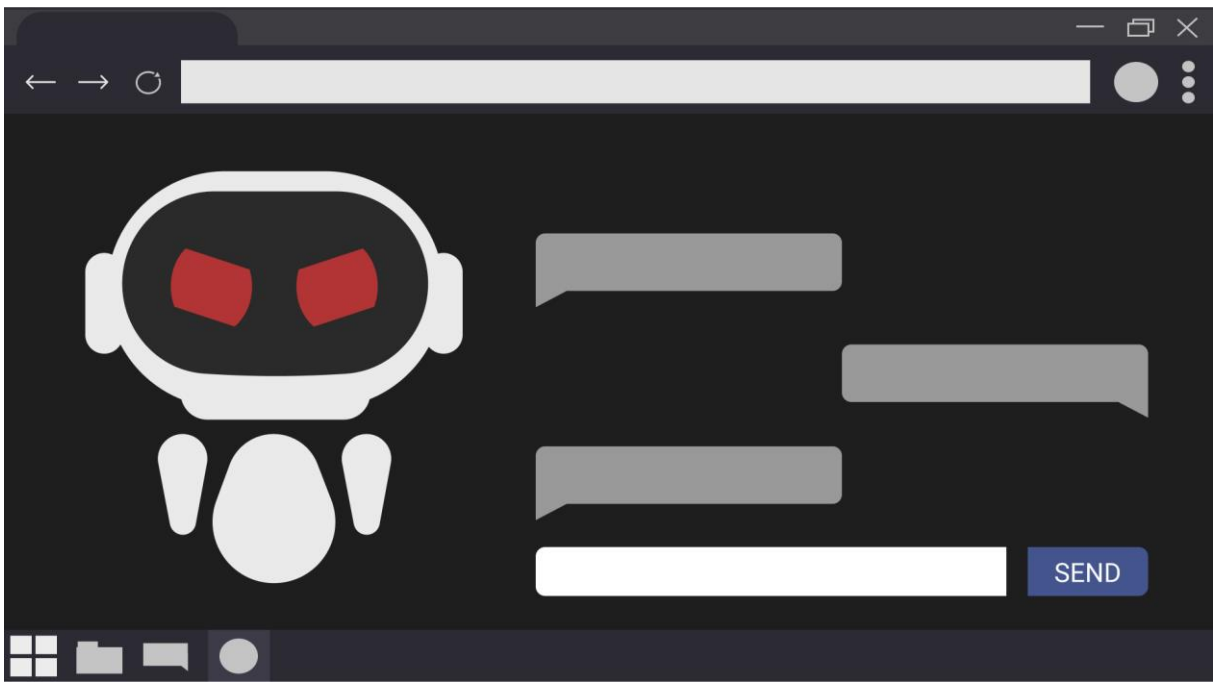
Figura 6 – Protótipo de tela com uma resposta com o sentimento de tristeza



Fonte: Elaborado pelo autor, 2021

A figura 7 representa uma resposta com raiva:

Figura 7 – Protótipo de tela com uma resposta com o sentimento de raiva



Fonte: Elaborado pelo autor, 2021

Após a apresentação dos protótipos de tela o presente trabalho elabora os casos de uso na próxima seção.

4.5 CASOS DE USO

Os casos de uso são o centro conceitual do desenvolvimento, porque guiam todo o processo ICONIX (BORILLO, 2000 apud BONA, 2002).

A lógica principal dos casos de uso é focar "no que os usuários precisam fazer com o sistema é muito mais poderoso do que outras abordagens tradicionais de elicitação de perguntar aos usuários o que eles querem que o sistema faça" (LUJÁN-MORA et al, 2002 apud LUJÁN-MORA, 2005, p. 38, tradução nossa).

Rosenberg e Scott (1999) citados por Bona (2002) explicam que um caso de uso é uma sequência de ações que um ator realiza no sistema para alcançar um objetivo, e são complementados pela autora, explicando os casos de uso como o que descreve e valida o que o sistema irá fazer, servindo de controle entre o usuário (BONA, 2002).

Sousa (2013) exemplifica o artefato de casos de uso com a figura 8:

Figura 8 – Exemplo do padrão do artefato descrição dos casos de uso

Descrição de casos de uso

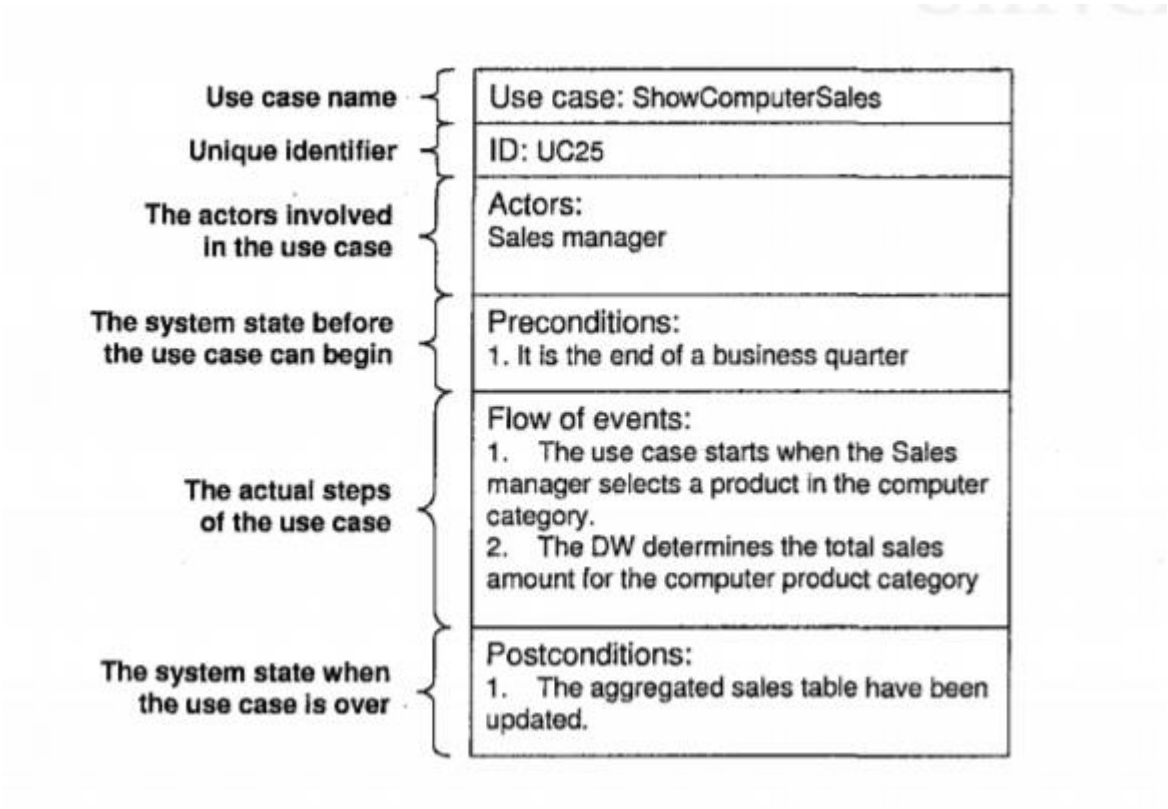
1. Nome: Login
   [Um nome único, provavelmente uma ação]
   Atores:
   [Nome de um ator]
   Pré-condições:
   O sistema mostra a tela de Login
   [Descreva o estado do sistema antes da execução do caso de uso]
   Fluxo Básico:
   Um usuário digita um nome de usuário e uma senha e clica no botão "Login".
   O sistema garante que seja uma nova sessão e que o usuário / senha seja válido.
   O sistema exibe uma mensagem de confirmação e está pronto para ser usado.
   [Etapas básicas do fluxo, é sempre iniciado pelo ator]
   Fluxos alternativos:
   * O nome de usuário tem uma sessão aberta:
   O sistema exibe uma mensagem de login do usuário.
   [Cada fluxo alternativo representa um comportamento alternativo geralmente devido às exceções que ocorrem no fluxo básico]
   Pós-condições:
   O sistema cria uma sessão para o usuário.
   [Descreva o estado do sistema após a execução do caso de uso]
   Pontos de Relacionamento: nenhum
   [Identifique os casos de uso que têm uma relação com este]

Fonte: adaptado de Sousa, 2013, p.121, tradução nossa

Segundo Sousa (2013, p. 120), o artefato “descrição dos casos de uso é um dos mais importantes artefatos desta fase e tem como objetivo expressar o comportamento do software por meio de cenários de uso de uma forma organizada e de fácil compreensão”. Também segundo Sousa (2013), os casos de uso devem ser escritos em voz ativa, no formato sujeito-verbo-objeto, usando um fluxo evento/reposta para descrever os dois lados do diálogo ator/sistema.

Luján-Mora (2005) também pontua um modelo de caso de uso, sobre o gerenciamento de vendas, fazendo uma consulta sobre as vendas trimestrais dos produtos na categoria de computadores (figura 9):

Figura 9 – Modelo de caso de uso UML



Fonte: Luján-Mora, 2005, p.39

A partir da contextualização anteriormente apresentada sobre os casos de uso o presente trabalho elabora dois casos de uso para serem considerados no processo de desenvolvimento.

O primeiro caso de uso comtempla a visualização de mensagens e emoção, e é representada no quadro 4:

Quadro 4 – Caso de uso UC001

Caso de uso:	Visualizar mensagens e emoção atual
ID:	UC001
Atores:	Usuário
Pré-condições:	*Nenhum*
Fluxo de eventos:	1. O ator visualiza o histórico de mensagens na tela; 2. O ator visualiza a emoção atual da aplicação.
Pós-condições:	*Nenhum*

Fonte: elaborado pelo autor, 2021.

O segundo caso de uso refere-se ao processamento e classificação de texto (quadro 5):

Quadro 5 – Caso de uso UC002

Caso de uso:	Processar e classificar texto
ID:	UC002
Atores:	Usuário
Pré-condições:	*Nenhum*
Fluxo de eventos:	1. O ator seleciona o campo de texto na aplicação e insere o texto desejado; 2. O ator clica no botão de enviar ao lado do campo de texto; 3. O sistema trava qualquer tentativa de envio de mensagens; 4. O sistema mostra a mensagem inserida em um campo não editável enviada pelo ator; 5. O sistema analisa a emoção a partir da entrada atual; 6. O sistema computa armazena, e incorpora a emoção analisada ao perfil existente; 7. O sistema gera uma emoção final a partir do perfil criado; 8. O sistema gera uma resposta a partir da entrada atual e da emoção atual; 9. O sistema mostra a mensagem inserida em um campo não editável enviada pelo sistema; 10. O sistema desbloqueia a trava de envio de mensagens.
Pós-condições:	*Nenhum*

Fonte: elaborado pelo autor, 2021.

Com os casos de uso definidos, a próxima seção aborda os protótipos de tela e os componentes visuais da aplicação.

4.6 DIAGRAMA DE CASO DE USO

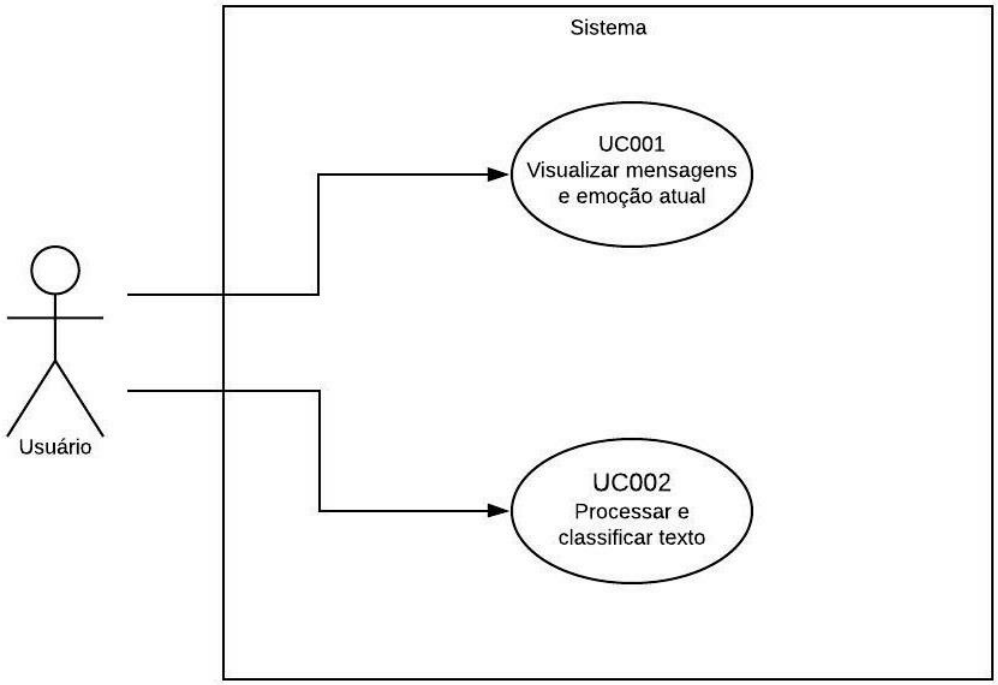
O diagrama de caso de uso é um elemento gráfico exclusivo, devido ao fato de que é utilizado para modelar o modo como às pessoas esperam usar o sistema (PENDER, 2004 apud PAULINO, 2014).

Segundo Bona (2002), pode-se aplicar algumas regras para definir qual associação utilizar entre os casos de uso:

- Usar inclusão (<<include>>) quando estiver repetindo o mesmo fluxo em dois ou mais casos de uso separados e deseja evitar a repetição;
- Usar generalização quando estiver descrevendo uma variação semelhante à outra, mas que faz um pouco mais, e deseja descrevê-la sem muito controle;
- Usar extensão (<<extend>>) quando descrever uma variação em comportamento normal e deseja utilizar a forma mais controlada, explicando os pontos de extensão no use-case geral.

Paulino (2014) aponta o foco principal do diagrama de caso de uso como como as interações entre o usuário e o sistema. Baseado nos casos de usos desenvolvidos anteriormente, o presente trabalho tem o seguinte diagrama (figura 10):

Figura 10 – Diagrama de caso de uso



Fonte: Elaborado pelo autor, 2021

A próxima seção apresenta o artefato diagrama de sequência, que tem por objetivo representar a sequência de processos da aplicação como um todo.

4.7 DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência detalha a descrição em funções que devem permitir ao programador mapear o código do programa a com análise gerada (PAULINO, 2014).

Uma das tarefas de projeto em (BONA, 2002) é especificar o comportamento através do diagrama de sequência, identificando as mensagens entre diferentes objetos dos casos de uso.

Paulino (2014) aplica o diagrama de sequência adotando o padrão MVC para seus casos de uso. Já segundo Sousa (2013) elabora-se um diagrama de sequência para cada caso de uso para mostrar em detalhes como ele será implementado, tendo como foco a alocação de comportamento as classes. Todavia, o presente trabalho elabora apenas um diagrama de

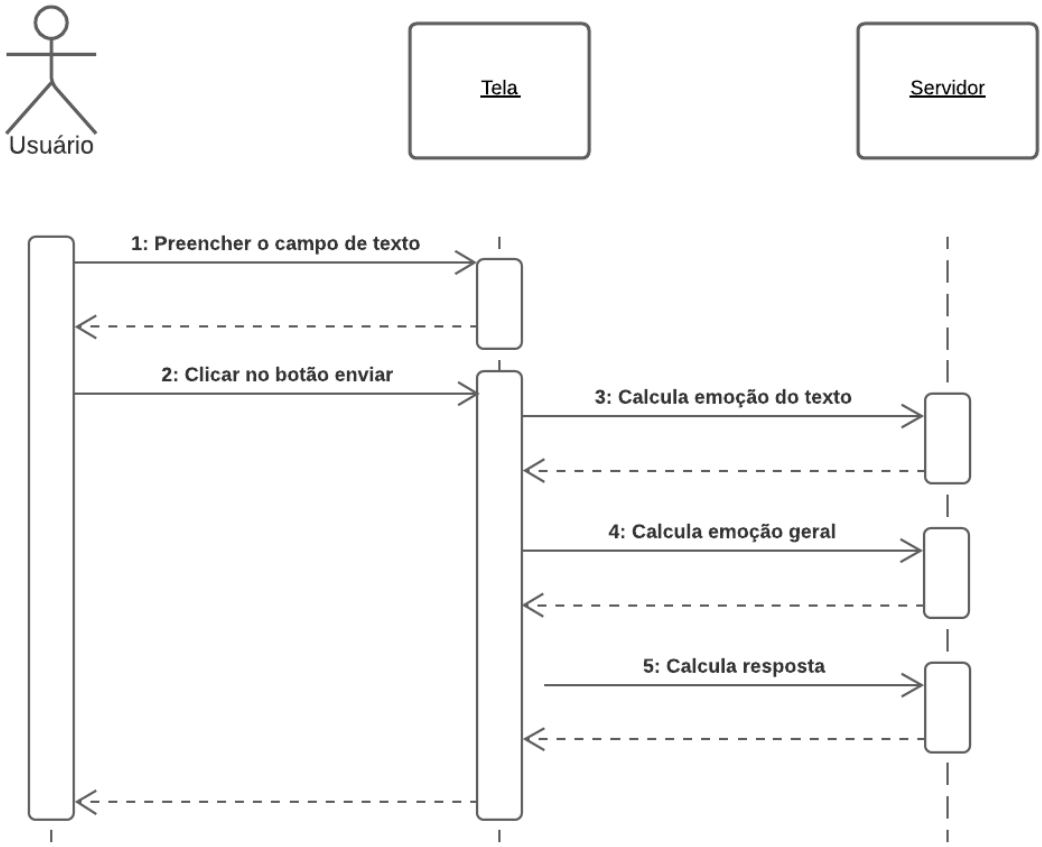
sequência abordando todas as classes, estando divididas em usuário, tela e servidor, devido à baixa complexidade de arquitetura do projeto.

Sousa (2013, p. 48) também explica o diagrama no contexto do desenvolvimento com ICONIX:

> O Diagrama de Sequência tem como objetivo mostrar a colaboração dinâmica (troca de mensagens) entre os vários objetos do software. No Iconix, assim como na maioria dos processos que usam a UML como linguagem de modelagem, o comportamento de um caso de uso é detalhado por meio do Diagrama de Sequência. O principal objetivo dessa atividade é designar comportamento, proveniente das classes de controle, para as classes de entidade e interface. A distribuição de responsabilidade por cada operação entre as classes é uma tarefa árdua e que demanda bastante esforço dependendo da experiência do desenvolvedor.

O presente trabalho, apesar de implementar a metodologia ICONIX apenas para o desenvolvimento dos requisitos, casos de uso e protótipos de tela, também elabora o diagrama de sequência devido aos seus proveitos anteriormente descritos (figura 11):

Figura 11 – Diagrama de sequência



Fonte: Elaborado pelo autor, 2021

A próxima seção aborda atividades relevantes que necessitam ser pontuadas para um bom entendimento dos requisitos de software, e são destacadas através de um diagrama de atividade.

4.8 DIAGRAMA DE ATIVIDADE

O diagrama de atividade é parte da especificação UML (BONA, 2002) sendo uma das quatro linguagens para especificar o comportamento dinâmico na UML 2.0 (GUERRA, 2012).

Segundo Ferreira (2011), o diagrama de atividades é descomposto em atividades e ações. Uma atividade é representada por um "retângulo de cantos arredondados, especifica um processamento complexo, que pode ser dividido em várias outras atividades originando um novo diagrama" (FERREIRA, 2011, p. 14). O mesmo autor complementa afirmando que por esse motivo, o próprio diagrama de atividades é considerado uma atividade.

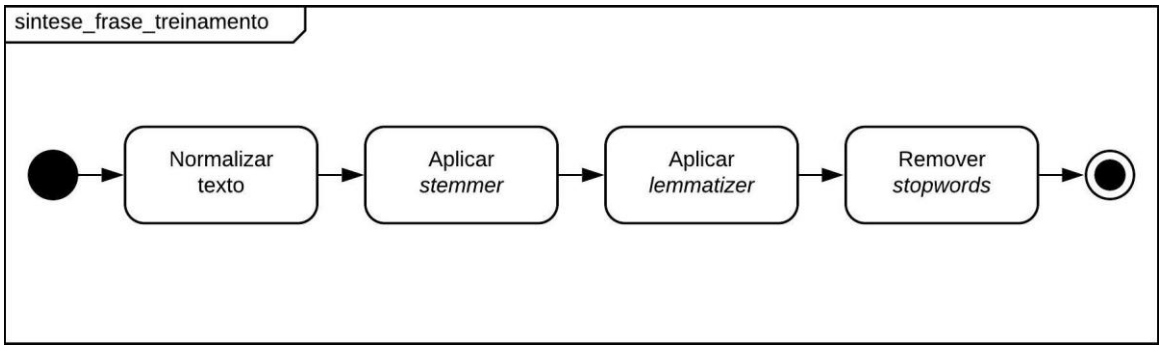
Bona (2002, p.107), apontando as deficiências do modelo ICONIX (descritos na seção 4.1) comenta a utilização do diagrama de atividades para um benefício relacionado a modelagem de processo de negócio, na fase preliminar do projeto:

> O ICONIX não sugere explicitamente nenhum diagrama para modelar processo de negócio na fase preliminar do projeto. Mesmo sendo o ICONIX um processo que pretende ser prático e simples, poderia no entanto, se beneficiar do diagrama de atividades disponível na UML (OMG®, 2001) para modelar processos de negócios. Da mesma forma que, outros processos igualmente simples e práticos como o Grapple, abordado por McConnell (apud SILVA & VIDEIRA, 2001) sugere a utilização do diagrama de atividades para modelar processos de negócio na fase preliminar.

Bona (2002), apontando as deficiências do modelo ICONIX (descritos na seção 4.1) comenta a utilização do diagrama de atividades para um benefício relacionado a modelagem de processo de negócio, na fase preliminar do projeto:

O presente trabalho desenvolve dois diagramas de atividades. O primeiro diagrama tem por objetivo representar a lógica de síntese das frases no treinamento do modelo que irá classificar os textos na aplicação (figura 12):

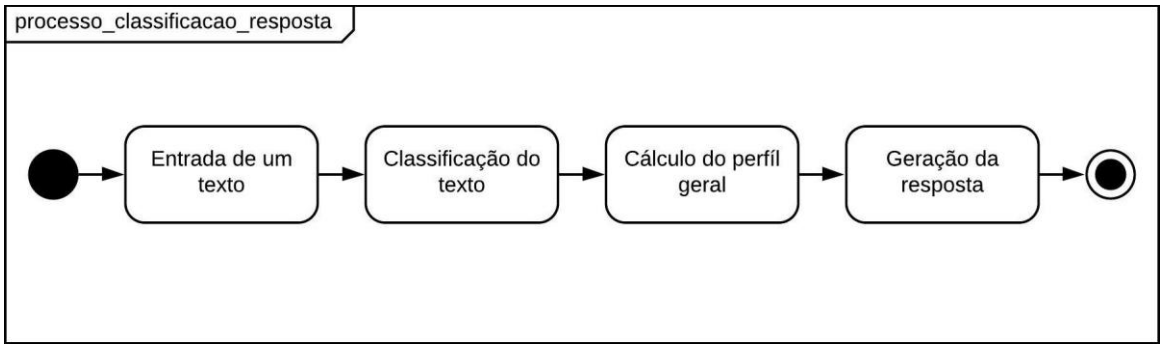
Figura 12 – Diagrama de atividade do modelo de classificação de texto



Fonte: Elaborado pelo autor, 2021

O segundo diagrama de atividade desenvolvido tem por objetivo definir o processo de classificação, criação de perfil e resposta que servirá de base para os desenvolvimentos futuros (figura 13):

Figura 13 – Diagrama de atividade do processo de classificação e resposta



Fonte: Elaborado pelo autor, 2021

Essa seção finaliza o capítulo de proposta de solução, e a partir das especificações do sistemas definidas é apresentado o próximo capítulo de desenvolvimento.

5 DESENVOLVIMENTO

Este capítulo apresenta as descrições das tecnologias e ferramentas, o histórico do desenvolvimento e a apresentação do protótipo funcional do presente trabalho.

5.1 TECNOLOGIAS E FERRAMENTAS

Essa seção apresenta a descrição de linguagens, ferramentas e bibliotecas utilizadas para o desenvolvimento do presente trabalho, como typescript, angular, python, NLTK, AIML, git, GitHub e Docker.

Para o desenvolvimento do protótipo funcional foram escolhidas ferramentas que já fossem conhecidas pelo autor, como o typescript e angular, ou que facilitariam ou automatizariam o cumprimento das especificações previamente descritas, como o python, o nltk e o aiml. Também foram escolhidas ferramentas para auxiliar o desenvolvimento através de versionamento, como é o caso do git e GitHub, e para a fácil execução em diversos ambientes, como o Docker.

5.1.1 Typescript

Typescript é uma linguagem de código aberto que se baseia em javascript, uma das ferramentas mais usadas do mundo, adicionando definições de tipo estático (MICROSOFT, 2021).

A linguagem foi escolhida devido ao framework angular que é utilizado para a camada do *front-end*, que será descrito na próxima seção.

5.1.2 Angular

Segundo Google (GOOGLE, 2021a) angular é uma plataforma de desenvolvimento baseada em typescript, e como plataforma, o angular inclui:

- Uma estrutura baseada em componentes para a construção de aplicativos da web escaláveis;
- Uma coleção de bibliotecas bem integradas que cobrem uma ampla variedade de recursos, incluindo roteamento, gerenciamento de formulários, comunicação cliente-servidor e muito mais;
- Um conjunto de ferramentas de desenvolvedor para ajuda-lo a desenvolver, construir, testar e atualizar seu código.

O framework foi utilizado pela fácil manutenção e fácil adição de funcionalidades, bem como pelo conhecimento e experiências anteriores do autor.

5.1.3 Python

Segundo a Python Software Foundation (2021) python é uma linguagem de programação interpretada, orientada a objetos e de alto nível com semântica dinâmica. Suas estruturas de dados embutidas de alto nível, combinadas com tipagem dinâmica e vinculação dinâmica, tornam-no muito atraente para o desenvolvimento rápido de aplicativos, bem como para uso como linguagem de script ou cola para conectar componentes existentes (PYTHON SOFTWARE FOUNDATION, 2021).

A linguagem foi escolhida pelas facilidades que apresenta e pelas bibliotecas disponíveis, como o aiml, o nltk, entre outras.

5.1.3.1 Natural Language Toolkit

O kit de ferramentas de linguagem natural, ou natural language toolkit (NLTK) é uma biblioteca para processamento de linguagem natural com python. Ele fornece interfaces fáceis de usar para mais de 50 corpora e recursos lexicais, como *WordNet*, junto com um pacote de bibliotecas de processamento de texto para classificação, tokenização, lematização, marcação, análise e raciocínio semântico, *wrappers* para bibliotecas de PNL de força industrial, e um fórum de discussão ativo (NLTK PROJECT, 2021).

A biblioteca fornece vários utilitários para o processamento de linguagem natural e análise de sentimento, e por isso foi escolhida para ser utilizada no desenvolvimento do presente trabalho.

5.1.3.2 Artificial Intelligence Markup Language

Linguagem de marcação de inteligência artificial, ou *Artificial Intelligence Markup Language* (AIML) é uma linguagem de script simples baseada em XML e o padrão aberto para escrever *chatbots* (PANDORABOTS, 2021).

Segundo a documentação da função AIML (2021) o AIML foi projetado pela primeira vez no final dos anos 1990, durante a explosão da *World Wide Web*. Enquanto a web acabou perdendo sua simplicidade original, em 1994 era possível criar um site com apenas um conhecimento rudimentar de algumas tags HTML.

Este trabalho utiliza a biblioteca python-aiml que possui o interpretador para AIML compatível com o python 3, e que foi criado a partir do pyaiml, uma versão do interpretador para o python 2 desenvolvida pelo Dr. Richard Wallace da fundação A.L.I.C.E. (VILLEGAS, 2021).

5.1.4 Git

Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar com tudo, desde projetos pequenos a muito grandes com velocidade e eficiência (GIT, 2021).

A ferramenta do Git é utilizada pelo presente trabalho para o controle de versão dos softwares desenvolvidos.

5.1.5 GitHub

GitHub (2021) é uma plataforma de desenvolvimento onde milhões de desenvolvedores e empresas criam, enviam e mantem seus softwares, onde é possível registar ou retroceder qualquer alteração nos códigos, e hospedar gratuitamente repositórios públicos e privados ilimitados.

A plataforma é utilizada pelo presente trabalho para manter e visionar os códigos desenvolvidos.

5.1.6 Docker

O Docker é uma aplicação baseada em *containers* que elimina as tarefas de configuração rotineiras e repetitivas e é usado em todo o ciclo de vida de desenvolvimento para o desenvolvimento de aplicativos rápido, fácil e portátil (DOCKER, 2021).

A aplicação é utilizada para o fácil desenvolvimento e execução das aplicações desenvolvidas, bem como resolver problemas de compatibilidade entre sistemas operacionais.

5.2 HISTÓRICO DE DESENVOLVIMENTO

Para o processo de desenvolvimento do protótipo foram seguidas as seguintes etapas:

- Modelagem de arquitetura do sistema;
- Prospecção do *dataset* para análise;
- Desenvolvimento dos protótipos;
- Codificação do classificador;
- Codificação do calculador de perfis;
- Codificação do gerador de respostas;
- Codificação do chat;
- Criação dos marcadores de inteligência.

As próximas seções descrevem as etapas do histórico de desenvolvimento apresentado.

5.2.1 Modelagem de arquitetura do sistema

A etapa de modelagem de arquitetura foi a etapa inicial onde foram decididas as ideias e o processo geral da aplicação. A ideia inicial era ter um único *front-end*, como permaneceu no protótipo final, e duas api's rodando em diferentes portas, uma para o processamento do sentimento e uma para o processamento de resposta.

A api para o processamento de sentimentos foi modelada inicialmente tendo em mente a biblioteca tensorflow.js e seria usada junto com node, uma plataforma de código aberto que executa códigos javascript no servidor. Já a api de processamento de respostas foi modelada para ser feita com o python versão 2.7 e com a biblioteca e a linguagem de marcação de inteligência artificial (AIML).

5.2.2 Prospecção do *dataset* para análise

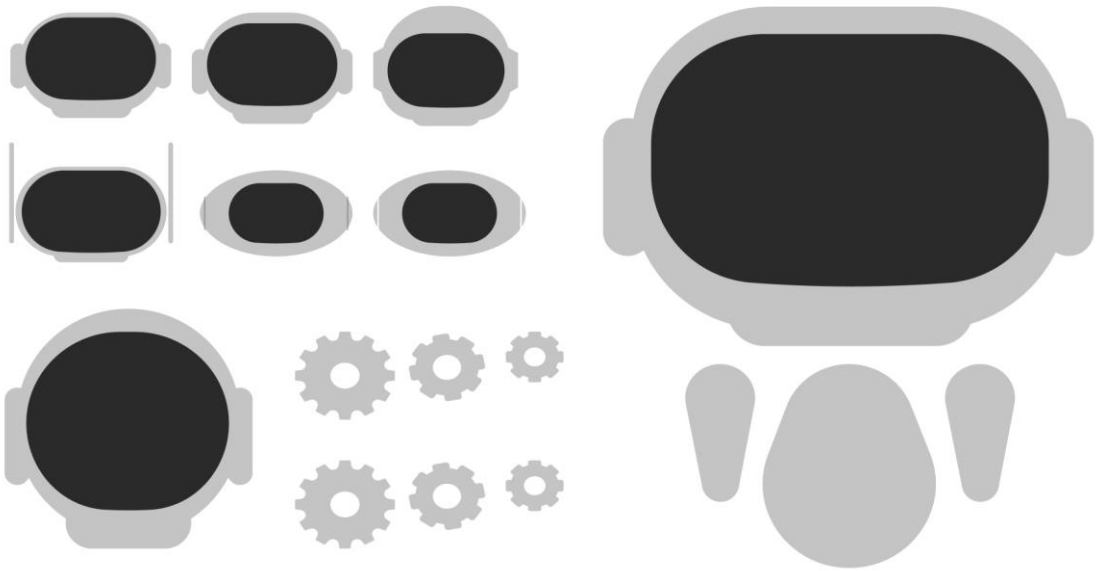
Na etapa de prospecção do *dataset* para análise foram buscados os *datasets* que seriam utilizados na etapa de análise de sentimento. Dezenas de massas de dados foram analisadas e consideradas, como por exemplo, o *sentiment treebank* (versão 1.0) da universidade de Stanford (STANFORD, 2021), o *sentiment polarity* (versão 1.0) do *movie review* data da faculdade de Cornell (CORNELL, 2021) e muitos outros do site Kaggle, uma comunidade de cientistas de dados.

Ao fim desta etapa uma base de dados de emoções para processamento de linguagem natural (GOVINDARAJ, 2021) foi escolhida.

5.2.3 Desenvolvimento dos protótipos

Essa fase foi responsável pelas definições dos layouts de tela, construção de alguns componentes visuais e levantamento de algumas regras de tela. Inicialmente foram desenvolvidos diversos tipos de componentes até a construção e definição de um definitivo, como representado na figura 14.

Figura 14 – Componentes da aplicação criados antes da prototipação das telas



Fonte: Elaborado pelo autor, 2021

Após a construção dos componentes visuais, a prototipação de tela e suas regras foram definidas sem muito esforço, tendo em vista as simples funcionalidades que executariam.

5.2.4 Codificação do protótipo

A etapa de codificação do protótipo teve por objetivo desenvolver as funcionalidades das aplicações. Ao decorrer do desenvolvimento, algumas mudanças relacionadas a arquitetura e ferramentas usadas aconteceram, e são descritas nessa seção.

5.2.4.1 Codificação do classificador

A etapa de codificação do classificador foi a etapa de desenvolvimento da aplicação de análise de sentimentos. Inicialmente a aplicação foi desenvolvida utilizando

redes neurais com a linguagem node e a biblioteca tensorflow.js. Entretanto, à medida que o a aplicação ia sendo desenvolvida, foi decidido fazer a migração dessa biblioteca e linguagem para ser compatível com a aplicação em python (versão 3) em razão de problemas de performance e desenvolvimento.

Ao fim da etapa foi desenvolvido um classificador utilizando o algoritmo naive bayes, que utiliza o *dataset* definido na seção 5.2.2, prospecção do *dataset* para análise, e classificava a probabilidade das entradas em serem uma das seis emoções disponíveis, raiva (*anger*), medo (*fear*), alegria (*joy*), amor (*love*), tristeza (*sadness*) e a surpresa (*surprise*).

5.2.4.2 Codificação do calculador de perfis

A etapa de codificação do calculador de perfis foi mais uma etapa de desenvolvimento de código. A funcionalidade foi construída na aplicação de *back-end* em python (versão 3) e tem como propósito calcular as alterações em um perfil com as seis emoções analisadas a partir da entrada dos dados disponibilizados pelo classificador. A partir do resultado individual de cada classe processado por uma função sigmoide, que transforma todos os resultados em números de 0 até 1, a aplicação incorpora os novos valores em suas respectivas classes do perfil, gerando assim um número final para cada classe, onde o maior número dentre eles é a emoção predominante utilizada em outras funcionalidades.

5.2.4.3 Codificação do gerador de respostas.

A etapa de codificação do gerador de resposta iniciou utilizando a linguagem python versão 2.7 e a biblioteca AIML, sendo desenvolvida sem grandes problemas. Apesar da pouca experiência do autor com a linguagem, uma api que respondia de acordo com os marcadores de inteligência artificiais foi desenvolvida. Posteriormente, a aplicação foi migrada para o python (versão 3) juntamente com as aplicações de análise de sentimentos e o calculador de perfis, sendo unificas em uma única aplicação.

5.2.4.4 Codificação do chat

A etapa de codificação do chat teve como objetivo desenvolver a aplicação de *front-end* que contém as interações com o usuário. Ela foi desenvolvida e integrada em paralelo com as outras aplicações descritas anteriormente, e foi a última a ser finalizada. Isso foi necessário devido a necessidade de entender e definir a comunicação entre a api desenvolvida e o cliente, bem como mapear as possíveis emoções para as interações.

Após o fim dessa etapa de codificação, todas as funcionalidades já estavam integradas, necessitando apenas de uma considerável melhora na base de marcadores de inteligência artificial criados, porque todas as entradas tinham a mesma resposta.

5.2.5 Criação dos marcadores de inteligência

A etapa da criação dos marcadores de inteligência foi a última etapa desenvolvida. Após a análise de sentimento automatizada e todos os processamentos de texto desenvolvidos com sucesso, as respostas que a aplicação daria as entradas e emoções via AIML foram desenvolvidas.

Por se tratar de um protótipo, um número limitado de reações e respostas foram desenvolvidas para cada emoção, e que podem ser atualizados e melhorados posteriormente.

5.3 APRESENTAÇÃO DO PROTÓTIPO FUNCIONAL

Esta seção apresenta o protótipo funcional desenvolvido, resultado do processo descrito na seção anterior, histórico de desenvolvimento.

Para este trabalho, um número limitado de marcadores de inteligência foi desenvolvido de forma a criar um diálogo inicial simples e exemplificar o potencial de análise e resposta do projeto.

O protótipo funcional desenvolvido segue ao máximo os protótipos de tela desenvolvidos e apresentados no capítulo 4, bem como os casos de uso descritos no mesmo capítulo, mas com diferenças visuais bem expressivas, como o dimensionamento da tela e das caixas de texto, os componentes usados nas animações de tela, o fundo de tela animado, entre outros.

As emoções desenvolvidas expressas na camada de *front-end* não se limitaram as três que foram definidas na seção 4.4, raiva, alegria e tristeza, mas foram desenvolvidas baseadas na base de dados escolhida, sendo elas a raiva (*anger*), o medo (*fear*), a alegria (*joy*), o amor (*love*), a tristeza (*sadness*) e a surpresa (*surprise*).

A arquitetura das aplicações do protótipo funcional, como descrito anteriormente na seção de histórico de desenvolvimento, sofreram algumas modificações da ideia inicial. Entretanto, funcionalmente, os objetivos iniciais foram cumpridos e atendem os objetivos do trabalho.

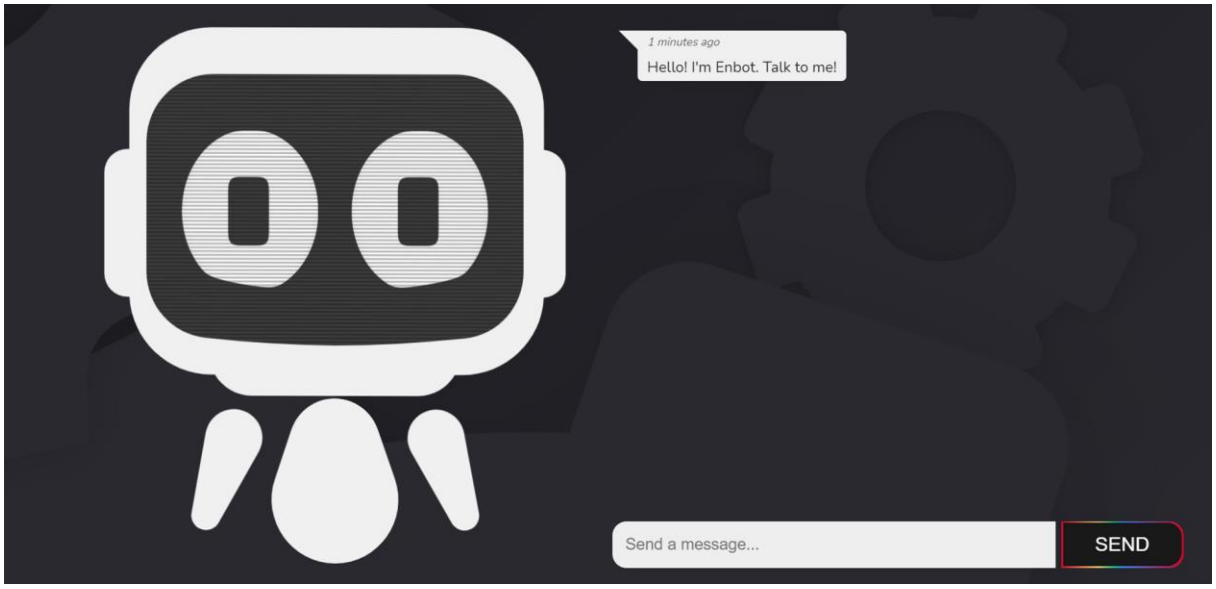
As próximas seções demonstram as telas do protótipo, as funcionalidades das telas, bem como os diálogos necessários para cada um dos sentimentos desenvolvidos, apontando as diferenças de resposta para cada sentimento baseado na mesma entrada.

5.3.1 Tela inicial

A tela inicial da aplicação, como mostra a figura 15, é única tela do sistema e, por consequência, a tela que contêm as principais funcionalidades. A tela em questão é extremamente simples, uma vez que o único propósito da camada de *front-end*, como já descrito anteriormente no presente trabalho, é fazer a ligação com as funcionalidades de análise de sentimento e a geração de resposta.

A figura 15 demonstra uma conversa com apenas uma balão de diálogo que pode ser traduzida para “Olá! Eu sou o Enbot. Fale comigo!”:

Figura 15 – Tela inicial do protótipo funcional



Fonte: Elaborado pelo autor, 2021

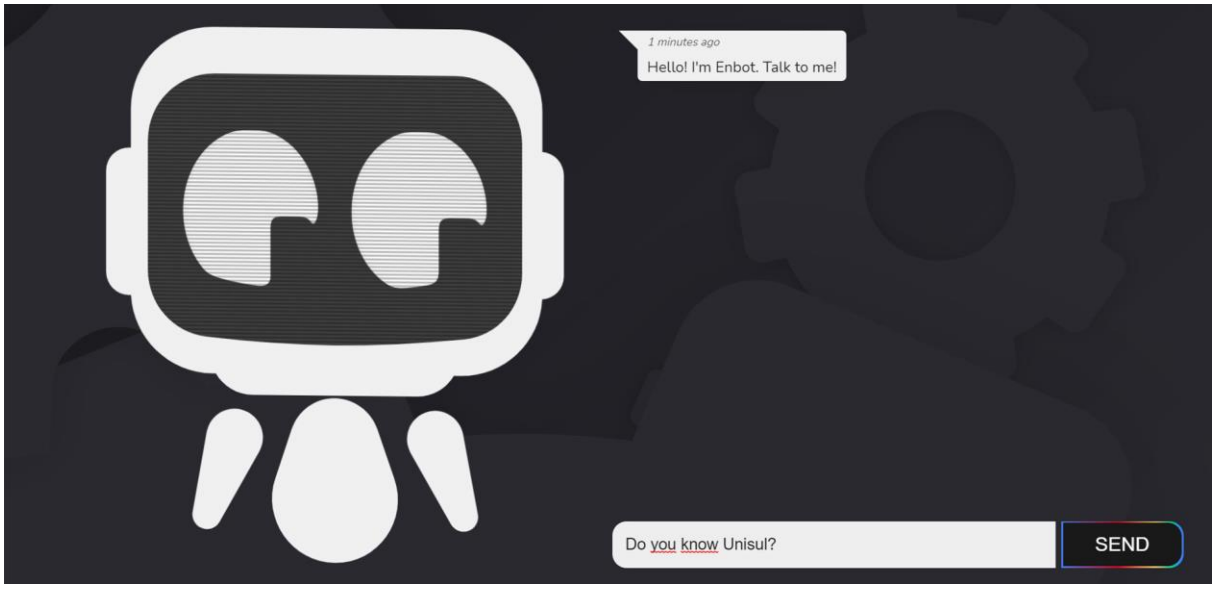
As próximas seções demonstram as ações da tela de envio de mensagens e a visualização de emoções.

5.3.1.1 Envio de mensagens

O envio de mensagens é a principal interação com o usuário do protótipo funcional. Através da caixa de envio de texto e do botão de enviar ao lado dela, é possível processar e analisar o sentimento de uma mensagem, bem como gerar uma resposta para essa entrada.

Ao entrar na tela principal, o usuário do protótipo poderá preencher o campo de texto no canto inferior direito e clicar no botão de enviar (figura 16).

Figura 16 – Mensagem fictícia digitada no protótipo funcional

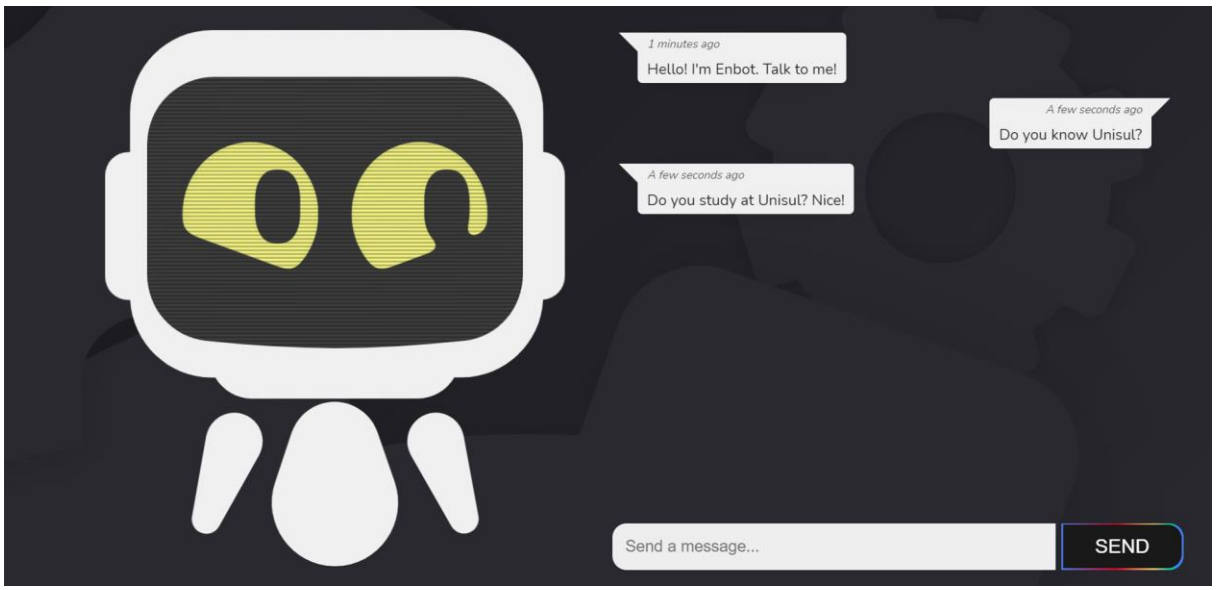


Fonte: Elaborado pelo autor, 2021

A figura anterior contém um balão de diálogo que se traduz na frase “Olá! Eu sou o Enbot. Fale comigo!”, e o texto na caixa de texto pode ser traduzido como “Você conhece a Unisul?

Ao executar as ações anteriormente descritas, dois novos balões de mensagens irão aparecer, uma com o texto de entrada escrito na caixa de mensagens, e um com a resposta para esse texto, como exemplificado na figura 17.

Figura 17 – Mensagem fictícia enviada no protótipo funcional



Fonte: Elaborado pelo autor, 2021

Os balões de diálogo da figura 17 podem ser traduzidos como “Olá! Eu sou o Enbot. Fale comigo!”, “Você conhece a Unisul?” e “Você estuda na Unisul? Legal!”.

A próxima seção demonstra e explica a etapa de pós processamento e visualização das emoções.

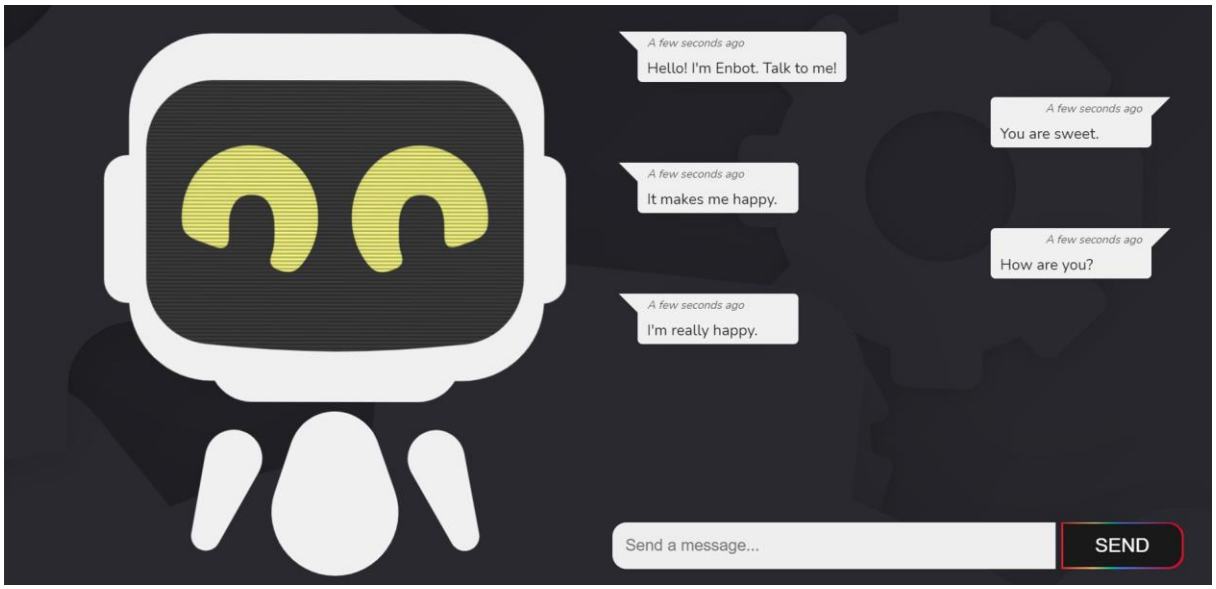
5.3.1.2 Visualização de emoções

O processamento do estado atual da aplicação acontece a cada entrada de uma mensagem. Quando esse estado muda, uma animação de tela é iniciada mudando o estado dos olhos do robô presente da tela principal. Para cada emoção citada na seção anterior, uma animação foi desenvolvida.

Essa seção apresenta exemplos de diálogos criados a partir do protótipo funcional que exibem diferentes animações de tela.

A emoção de alegria é uma das emoções mais recorrentes em diversos *datasets* diferentes, e pode ser alcançada a partir do seguinte diálogo (figura 18):

Figura 18 – Diálogo de alegria criado com o protótipo funcional

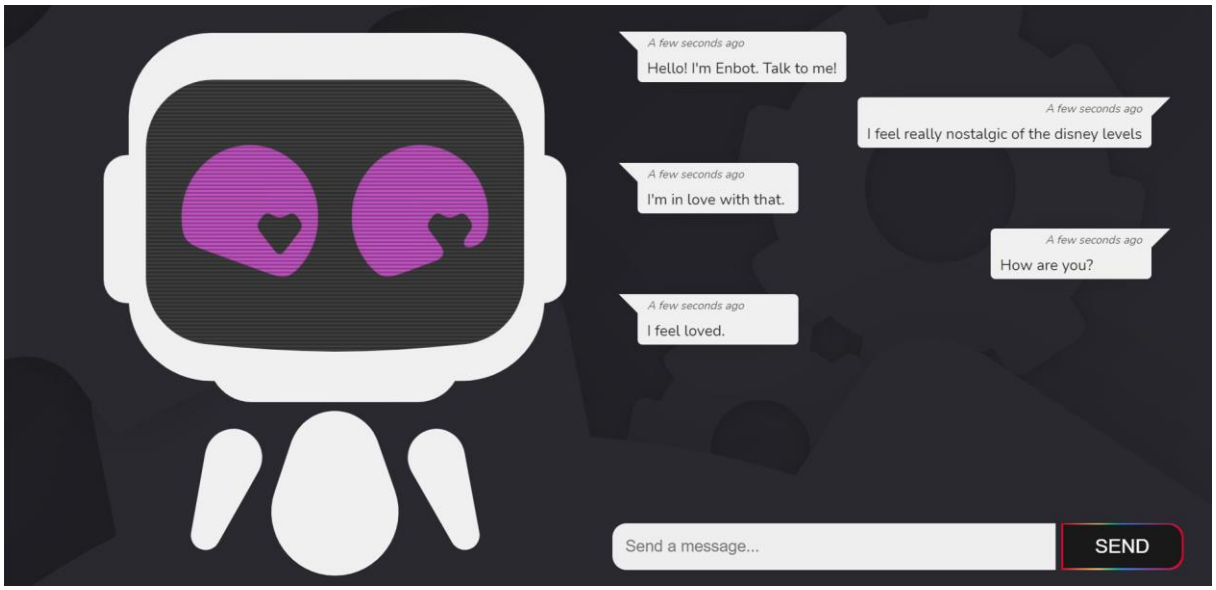


Fonte: Elaborado pelo autor, 2021

Os textos representados na figura anterior podem ser traduzidos como “Olá! Eu sou o Enbot. Fale comigo!”, “Você é adorável.”, “Isto me faz feliz.”, “Como você está?” e “Estou realmente feliz.”.

A emoção de amor também ganhou uma animação devido a sua existência no *dataset* escolhido, e é representada pela figura 19:

Figura 19 – Diálogo de amor criado com o protótipo funcional

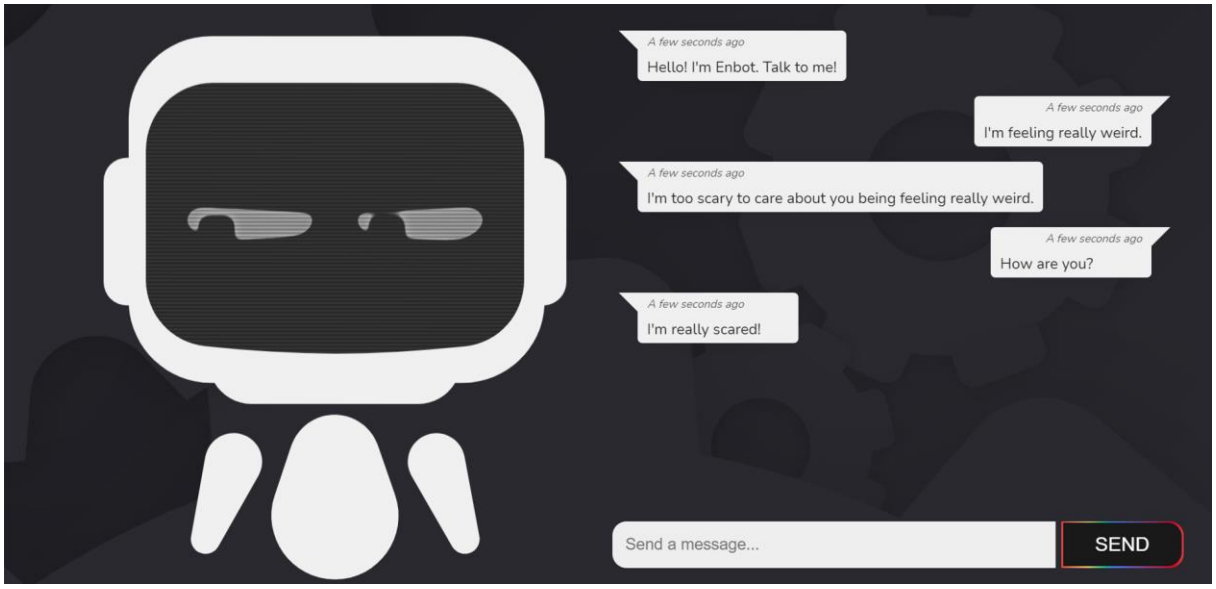


Fonte: Elaborado pelo autor, 2021

Na figura anterior (figura 19) é apresentado um diálogo que pode ser traduzido como “Olá! Eu sou o Enbot. Fale comigo!”, “Sinto muita nostalgia dos níveis da Disney.”, “Eu estou apaixonado por isso.”, “Como você está?” e “Sinto-me amado”.

A emoção de medo é uma das seis principais emoções (EKMAN, 1992) e está inclusa na lista das emoções do presente trabalho (figura 20). Na figura 20 é apresentado um diálogo em inglês que pode ser traduzido como “Olá! Eu sou o Enbot. Fale comigo!”, “Eu me sinto muito estranho.”, “Eu estou muito assustado para se importar que sobre você se sentindo muito estranho.”, “Como você está?” e “Estou muito assustado.”.

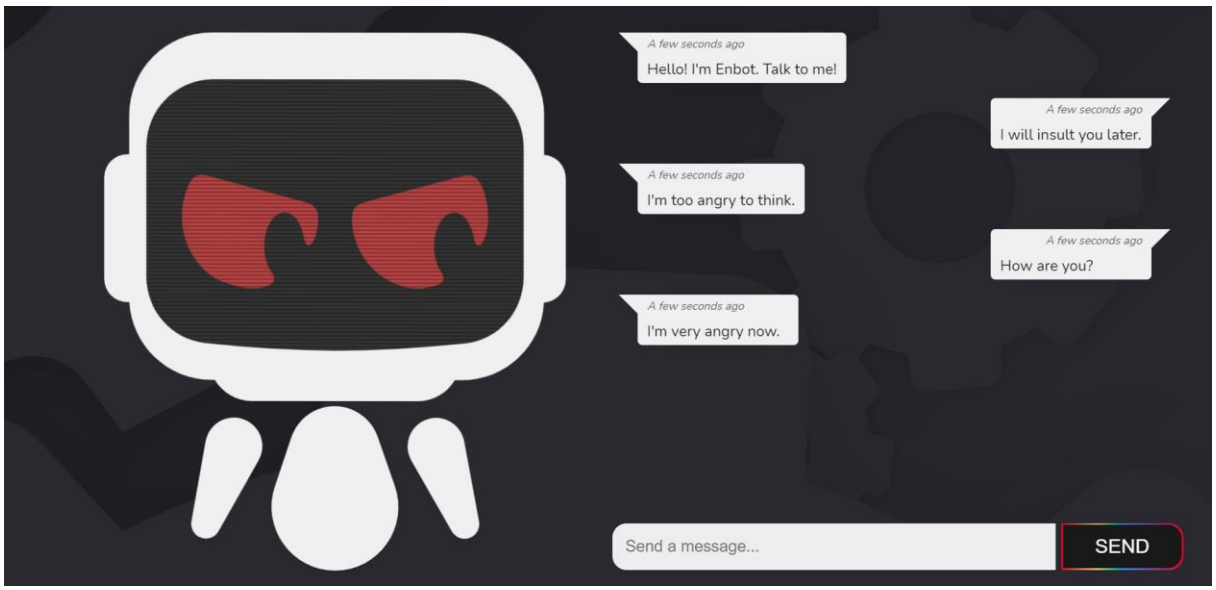
Figura 20 – Diálogo de medo criado com o protótipo funcional



Fonte: Elaborado pelo autor, 2021

Outro sentimento comum é a raiva, que possui seu diálogo representado pela figura 21:

Figura 21 – Diálogo de raiva criado com o protótipo funcional

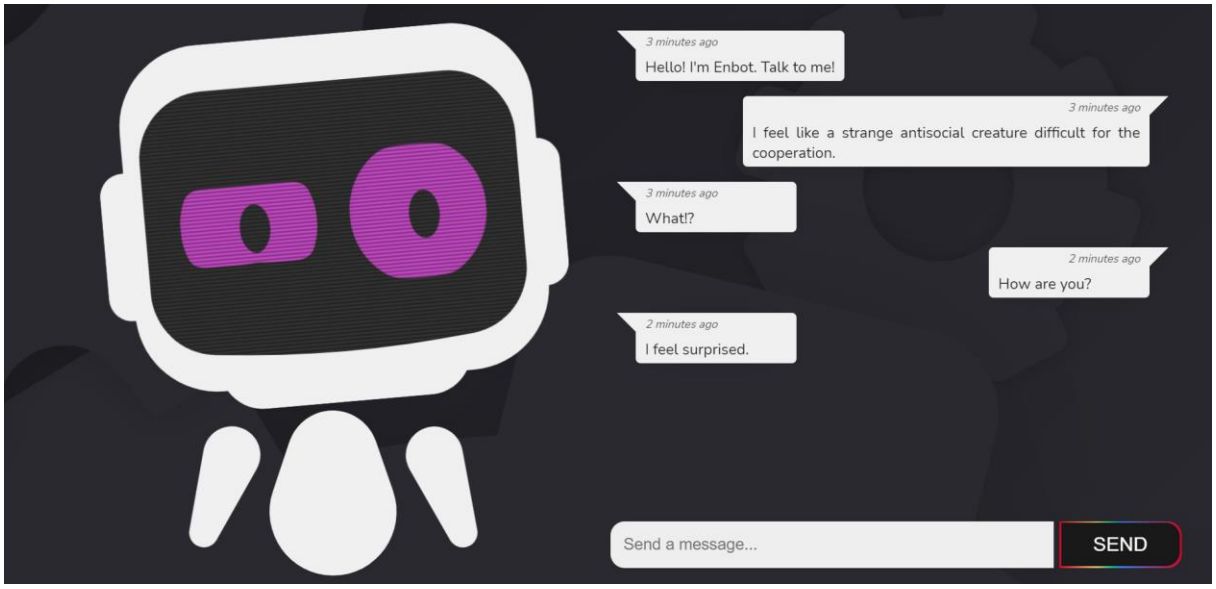


Fonte: Elaborado pelo autor, 2021

O diálogo de raiva, representado pela figura 21, possui cinco textos em inglês que podem ser traduzidos como “Olá! Eu sou o Enbot. Fale comigo!”, “Eu vou te insultar mais tarde.”, “Estou com muita raiva pra pensar.”, “Como você está?” e “Eu estou com muita raiva agora”.

A surpresa é um sentimento que também foi contemplado pela análise das aplicações, e pode ser alcançada através do seguinte diálogo (figura 22), que pode ser traduzido como “Olá! Eu sou o Enbot. Fale comigo!”, “Eu me sinto uma estranha criatura antissocial difícil para a cooperação.”, “O que!?”, “Como você está?” e “Me sinto surpreso”:

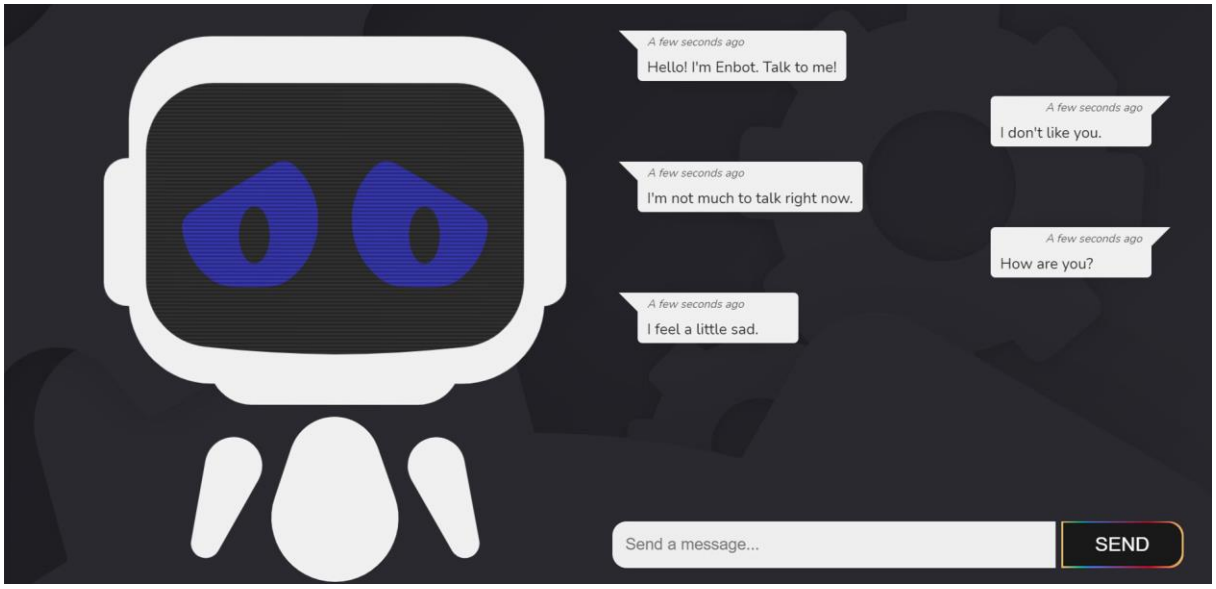
Figura 22 – Diálogo de surpresa criado com o protótipo funcional



Fonte: Elaborado pelo autor, 2021

A tristeza é um dos sentimentos mais fáceis para serem alcançados da aplicação, e pode ser representada com o diálogo da figura 23. O diálogo da figura 23 pode ser traduzido como “Olá! Eu sou o Enbot. Fale comigo!”, “Eu não gosto de você.”, “Eu não tenho muito o que falar agora.”, “Como você está?” e “Eu me sinto um pouco triste.”.

Figura 23 – Diálogo de tristeza criado com o protótipo funcional



Fonte: Elaborado pelo autor, 2021

A próxima seção descreve a seção de avaliação do projeto desenvolvido, que acontece a partir da análise do classificador e das interações com o usuário.

5.4 AVALIAÇÃO

Esta seção apresenta a avaliação do protótipo funcional. As avaliações estão divididas de forma a avaliar o classificador de emoções, bem como a aplicação e suas interações.

5.4.1 Classificador

A matriz de confusão e algumas fórmulas de desempenho utilizadas para sistemas de aprendizado, (ARRUDA, 2013, BATISTA, 2003, BLAZ, 2017, CECI, 2015, SILVA, 2013), tais como a taxa de erro e a precisão, entre outros, são as métricas implementadas de forma a avaliar o classificador do presente trabalho.

Os diferentes tipos de erros e acertos realizados por um classificador podem ser sintetizados em uma matriz de confusão (BATISTA, 2003), ou seja, uma matriz, onde as linhas representam as classes verdades e as colunas as preditas e cada célula desta matriz contêm um número de amostras de acordo com a resposta do classificador (ARRUDA, 2013). Os valores representam a quantidade de instâncias corretas e incorretas classificadas pelos algoritmos (BLAZ, 2017).

Arruda (2013) considera que em um problema de duas classes, convenciona-se uma delas como positiva, sendo assim a outra como negativa, tendo-se o *VP* como o número de verdadeiros positivos, o *VN* como os verdadeiros negativos, *FP* sendo os falsos positivos e *FN* como falsos negativos. De acordo com o mesmo autor, também se tem $N = FP + FN + VP + VN$, onde N é o número total de amostras (ARRUDA, 2013). Batista (2003), assim como Ceci (2015), também utilizam uma matriz de confusão para um problema de duas classes rotuladas como positiva e negativa, como mostra o quadro 6:

Quadro 6 – Exemplo de matriz de confusão com duas classes

		Classe prevista	
		C_1	C_2
Classe real	C_1	Verdadeiro Positivo (VP) C_1	Falso Negativo (FN) C_2
	C_2	Falso Positivo (FP) C_1	Verdadeiro Negativo (VN) C_2

Fonte: Adaptado de Blaz (2017, p.23)

Segundo Blaz (2017), com base nos resultados da uma matriz de confusão, é possível calcular as medidas precisão, revocação e a medida-f para cada classe, que representam o desempenho do classificador para diferentes classes, C_1, C_2...C_n, sendo n>=2.

Silva (2013), além da matriz de confusão de nível de documento que considera as classes positivas e negativas, análogo aos autores descritos anteriormente, também demonstra uma matriz de confusão para a classificação em nível de característica, considerando três classes como uma de suas métricas utilizadas, como mostra o quadro 7:

Quadro 7 – Exemplo de matriz de confusão com três classes

		Automático		
		C_1	C_2	C_3
Manual	C_1	t_p (1)	erro (2,1)	erro (3,1)
	C_2	erro (1,2)	t_p (2)	erro (3,2)
	C_3	erro (1,3)	erro (2,3)	t_p (3)

Fonte: Adaptado de Silva (2013, p.63)

De acordo com Batista (2003) a taxa de erro e a precisão são duas medidas utilizadas para medir o desempenho de sistemas de aprendizado. O autor também explica:

> [...] quando a probabilidade a priori de cada classe é muito diferente, isto é, quando existe um grande desbalanço entre as classes, tais medidas podem ser enganosas. Por exemplo, é bastante simples criar um classificador com 99% de precisão, ou de forma similar, com 1% de taxa de erro, se o conjunto de dados possui uma classe majoritária com 99% do número total de exemplos. Esse classificador pode ser criado simplesmente rotulando todo novo caso como pertencente a classe majoritária. (BATISTA, 2003, p. 144)

Os testes do classificador do protótipo funcional usam uma base de teste com dois mil itens, fornecida pelo mesmo autor (GOVINDARAJ, 2021) da base de dados utilizada, com o objetivo de corroborar os resultados apresentados no processo de avaliação.

Além da criação da matriz de confusão, algumas variáveis de avaliação também se destacam entre os métodos de avaliação, como o erro total (ARRUDA, 2013, BATISTA, 2003), a acurácia (BATISTA, 2003, CECI 2015), a precisão (ARRUDA, 2013, BLAZ 2017, CECI 2015, SILVA 2013), a revocação (ARRUDA, 2013, BLAZ 2017, CECI 2015, SILVA 2013), e a medida-f (ARRUDA, 2013, BLAZ 2017, CECI 2015, SILVA 2013).

O erro total é apresentado por Arruda (2013) como sendo o erro do modelo (equação 5):

$$Erro = \frac{FP + FN}{n} \tag{5}$$

Ceci (2015) demonstra o cálculo de acurácia utilizando a seguinte equação (equação 6):

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{6}$$

Para Blaz (2017) a precisão representa a quantidade de instâncias que foram corretamente classificadas, como pertencentes a uma determinada classe pelo algoritmo (equação 7):

$$Precisão = \frac{VP}{VP + FP} \tag{7}$$

Também para Blaz (2017), dada uma classe específica, a revocação representa as instâncias classificadas corretamente para essa classe, representada pela equação 8:

$$Revocação = \frac{VP}{VP + FN} \quad (8)$$

A medida-f, que consiste na média harmônica entre a precisão e a revocação (SILVA, 2013), é representada por Ceci (2015) mediante a fórmula 9:

$$Medida\ f = 2\ .\ \frac{precisão\ .\ revocação}{precisão + revocação} \quad (9)$$

Para o presente trabalho foi desenvolvido a seguinte matriz de confusão, de forma a avaliar os erros e acertos de cada classe, e é representada pelo quadro 8:

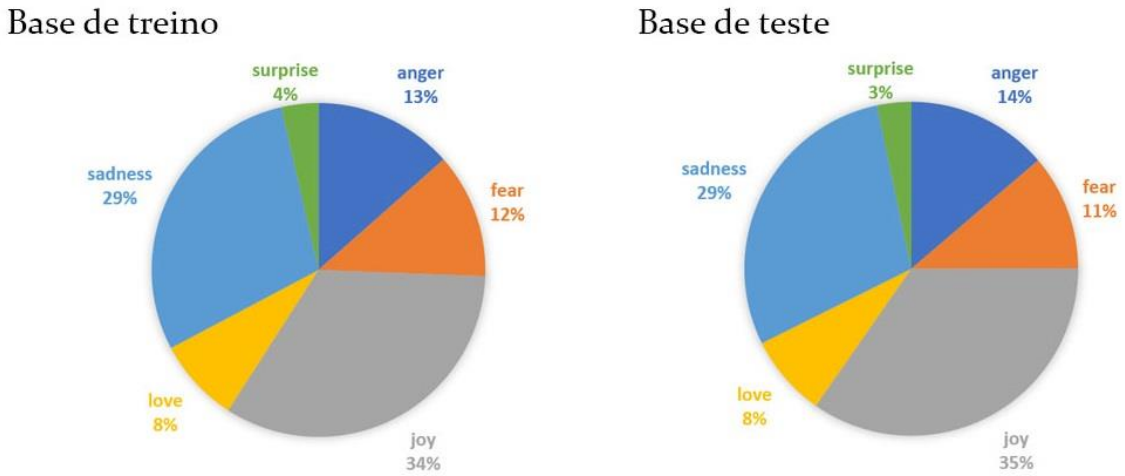
Quadro 8 – Matriz de confusão das classes utilizadas

		Total					
		Anger	Fear	Joy	Love	Sadness	Surprise
Classificado	Anger	199	15	20	2	39	0
	Fear	12	136	23	2	51	0
	Joy	10	8	633	19	24	1
	Love	6	1	77	50	24	1
	Sadness	14	11	35	10	511	0
	Surprise	2	12	33	0	17	2

Fonte: Elaborado pelo autor, 2021

A partir da matriz desenvolvida é possível traçar uma comparação entre a base de treino e a base de teste utilizada. A base de treino, que foi utilizada para o treinamento do classificador, possui dezesseis mil registros, enquanto a base de teste, como já descrito, possui dois mil registros. A figura 24 apresenta a distribuição de ambas as bases de dados, que apesar de possuírem números de registros diferentes, possuem uma distribuição semelhante:

Figura 24 – Distribuição dos sentimentos nas bases de dados



Fonte: Elaborado pelo autor, 2021

Os cálculos e variáveis apresentados anteriormente também foram feitos para cada classe individualmente, como sugerido por Blaz (2017). Essas varáveis estão expressas no quadro 9:

Quadro 9 – Variáveis da matriz de confusão das classes utilizadas

		Emoções					
		Anger	**Fear**	**Joy**	**Love**	**Sadness**	**Surprise**
Variáveis	**VP**	199	136	633	50	511	2
	FP	76	88	62	109	70	64
	FN	44	47	188	33	155	2
	VN	1681	1729	1117	1808	1264	1932
	Erro total	0.06	0.06	0.125	0.07	0.11	0.03
	Acurácia	0.94	0.93	0.875	0.92	0.88	0.96
	Precisão	0.72	0.60	0.91	0.31	0.87	0.03
	Revocação	0.81	0.74	0.77	0.60	0.76	0.5
	Medida-f	0.76	0.66	0.83	0.41	0.81	0.05

Fonte: Elaborado pelo autor, 2021

Apesar dos números apresentados relativos ao total das bases de treino e teste possuírem uma variação de classe para classe, eles apresentam divergências se comparadas

com a acurácia e a precisão. Mesmo que os sentimentos de alegria (*joy*) e tristeza (*sadness*) possuam um maior número de registros, essas variáveis possuem valores semelhantes com as melhores de cada tipo.

A próxima seção apresenta a avaliação das interações com usuário desenvolvidas no protótipo funcional, exercidas através de uma pesquisa.

5.4.2 Interações

Essa seção apresenta a avaliação da aplicação como um todo através da exposição das funcionalidades para usuários. Para avaliar o protótipo funcional foi criado com um formulário com a ferramenta google forms. A ferramenta pode coletar e organizar informações em pequena ou grande quantidade, gratuitamente (GOOGLE, 2021b).

Para avaliar o protótipo foram desenvolvidas sete perguntas a fim de entender a concepção dos profissionais anteriormente descritos quanto as funcionalidades do sistema. Os participantes foram orientados a seguir o procedimento descrito nos projetos e conduzidos a realizar alguns diálogos aleatórios após a execução da aplicação.

As aplicações foram disponibilizadas através da plataforma GitHub para a execução em ambiente local, conforme previsto delimitações definidas anteriormente na seção 3.3, para serem testadas e avaliadas. Devido a certa complexidade e conhecimento técnico requerido para a execução do conjunto de aplicações, apenas alguns desenvolvedores de diferentes níveis (júnior, pleno, sênior) foram avaliados. As perguntas aplicadas estão expressas no quadro 10.

O questionário recebeu 7 respostas, e após seu término foram computados alguns gráficos (figuras 25-29) a partir da ferramenta usada, o google forms.

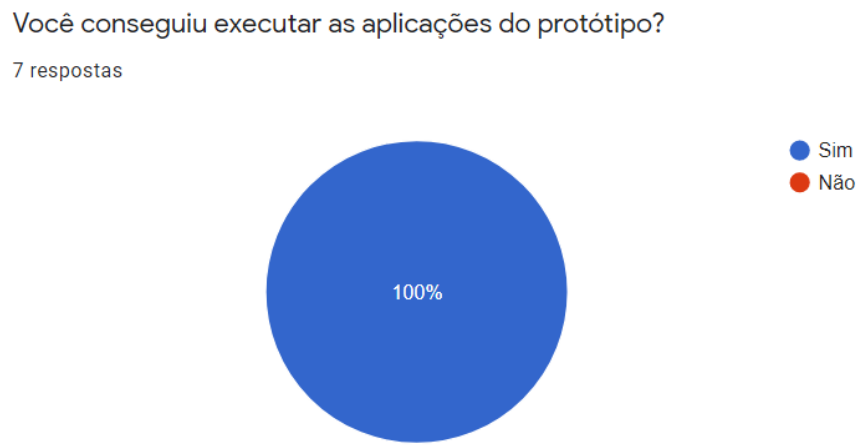
O primeiro gráfico tem por objetivo expressar os usuários desenvolvedores que conseguiram executar as aplicações em ambiente local (figura 25). Como se pode analisar, todos os usuários que responderam ao questionário conseguiram executar o projeto em seus sistemas operacionais sem grandes problemas.

Quadro 10 – Questões aplicadas para a avaliação do protótipo funcional

	Questão	Objetivo
Q1	Você conseguiu executar as aplicações do protótipo?	Garantir que o usuário desenvolvedor conseguiu executar as aplicações.
Q2	Você se considera em que nível de conhecimento técnico?	Entender o que o usuário pensa sobre seu nível de conhecimento técnico e traçar uma ponte com as demais questões.
Q3	Qual o nível de inglês que você se considera atualmente?	Conhecer o nível de inglês do usuário para saber se é capaz de estabelecer um diálogo simples na aplicação.
Q4	Após o total de duas entradas de texto diferentes, o sentimento predominante mudou alguma vez?	Saber se os perfis estão sendo montados corretamente de forma que os sentimentos existentes sejam considerados na próxima análise.
Q5	Após o total de cinco entradas de texto diferentes, o sentimento predominante mudou alguma vez?	Saber se as aplicações foram testadas o suficiente e com uma variedade grande de entradas ao ponto que a emoção predominante possa mudar mais de uma vez.
Q6	Para você, quais são os objetivos da aplicação? Justifique.	Entender a respeito das opiniões dos usuários quanto aos objetivos da aplicação.
Q7	Você considera as classificações e emoções feitas fiéis as entradas? Justifique.	Ter conhecimento da opinião dos usuários quanto aos erros e acertos da classificações das entradas feitas.

Fonte: Elaborado pelo autor, 2021

Figura 25 – Gráfico analítico Q1



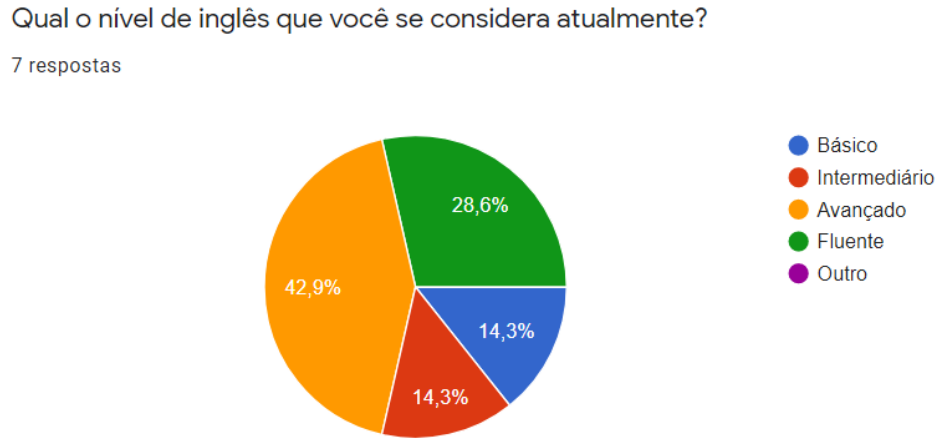
Fonte: Elaborado pelo autor, 2021

O segundo e terceiro gráfico fazem análises dos perfis que foram entrevistados, relativos ao conhecimento técnico e ao nível de inglês dos participantes:

Figura 26 – Gráfico analítico Q2

Fonte: Elaborado pelo autor, 2021

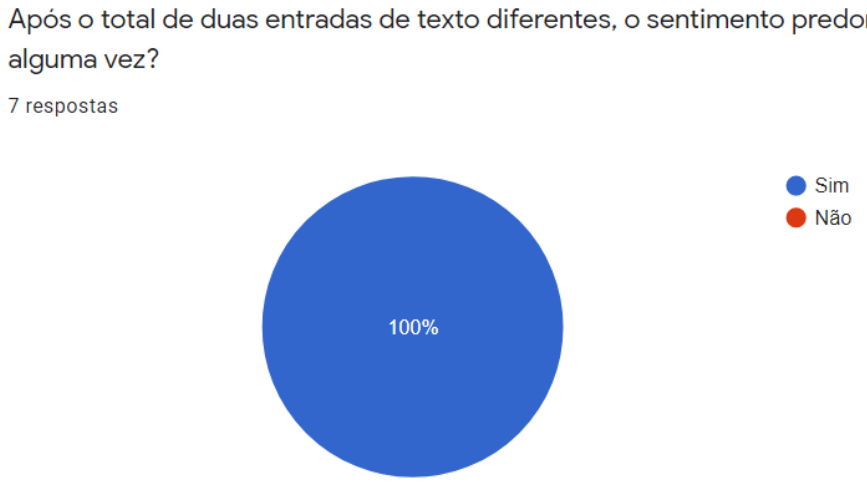
Figura 27 – Gráfico analítico Q3



Fonte: Elaborado pelo autor, 2021

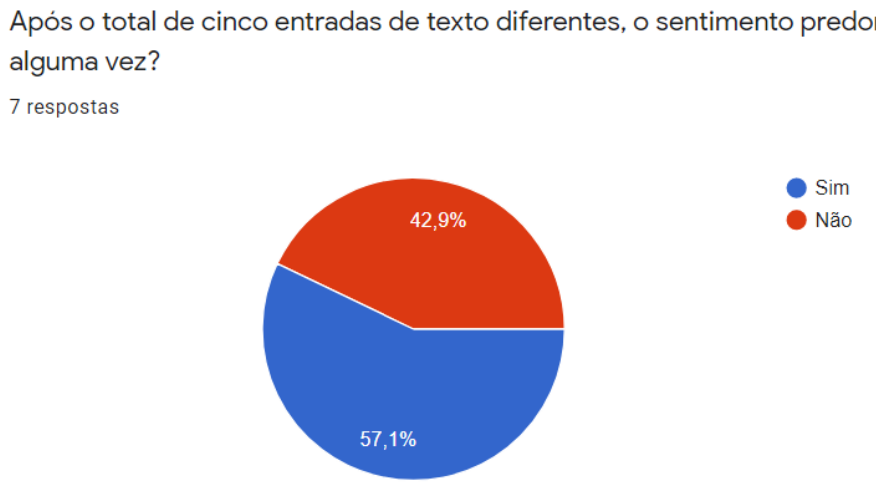
Os dois últimos gráficos apresentam as mudanças dos sentimentos após a ação de enviar alguma mensagem. O quarto gráfico (figura 28) representa a ação de analisar alguma entrada com sucesso, enquanto o quinto e último gráfico (figura 29) representa as mudanças após a primeira análise. Como aparente nos gráficos, após a primeira análise, um pouco mais da metade dos usuários conseguiram fazer o sentimento em predominante mudar.

Figura 28 – Gráfico analítico Q4



Fonte: Elaborado pelo autor, 2021

Figura 29 – Gráfico analítico Q5



Fonte: Elaborado pelo autor, 2021

Além dos gráficos apresentados, é importante destacar as respostas dadas para as questões Q6 e Q7 pelos usuários que testaram o protótipo funcional. Cada um, individualmente, elaborou uma resposta para cada questão a respeito dos objetivos da aplicação (quadro 11) e sobre a análise de texto (quadro 12):

Quadro 11 – Respostas da pergunta sobre os objetivos da aplicação

	Questão	Resposta
R1	Q6	"Fazer com que o bot, com base em uma conversa por chat, analise o que foi escrito e emita um sentimento para continuar a conversa."
R2	Q6	"Analisar os inputs fornecidos pelos usuários e calcular o sentimento baseado nesses inputs."
R3	Q6	"Talvez seja para eu ajudar ele. Ele ficou triste. Parece que não consegui ajudar o robozinho. :("
R4	Q6	"Realizar uma análise sentimental baseada em diálogos."
R5	Q6	"Acredito que o objetivo da aplicação é avaliar as mensagens que são enviadas."
R6	Q6	"Identificar através das conversas no chat se o tom da conversa é positivo, negativo ou neutro."
R7	Q6	"Não sei, talvez entretenimento."

Fonte: Elaborado pelo autor, 2021

Quadro 12 – Respostas da pergunta sobre a análise feita

	Questão	Resposta
R1	Q7	"Sim, quando eu iniciei a conversação de forma amigável ele ficou feliz e quando fui grosseiro ele ficou bravo."
R2	Q7	"Na maioria do tempo foi fiel, mas em alguns momentos eu inputava um valor de "How are you?", por exemplo e ele atuava com diferentes emoções. Provavelmente por ser algum comando parecido entre as emoções dele"
R3	Q7	"Sim! Ele ficou triste e não quis mais falar. Eu ofereci ajuda mas ele ficou mais triste ainda."
R4	Q7	"Sim, após realizar alguns testes pude constatar que o robô alterava seu humor de acordo com a conversa. Quando eu fazia algum xingamento ele ficava com raiva ou quando eu fazia algum elogio ele ficava feliz, por exemplo."
R5	Q7	"Sim. Apesar das respostas serem geralmente as mesmas, o bot mudava depois que eu enviar algumas mensagens."
R6	Q7	"Na grande maioria dos casos sim. Em algumas situações, após nutrir o bot com muitos sentimentos positivos (ou negativos), não foi possível reverter o seu sentimento."
R7	Q7	"Sim, quando o ofendi ele ficou bravo e quando fui amigável ele ficou feliz."

Fonte: Elaborado pelo autor, 2021

É importante ressaltar que as respostas anteriores dos quadros foram realizadas por programadores. Como é possível perceber como Q6 e Q7, um forte intuito técnico é realizado em cada uma das respostas dadas.

Com base nas respostas submetidas em Q6, que pergunta a respeito dos objetivos da aplicação, compreende-se que segundo a opinião da maioria dos entrevistados, os objetivos da aplicação são realizar algum tipo de análise ou avaliação com as entradas.

Baseado nas respostas de Q7, que avalia a opinião dos usuários quanto aos erros e acertos da classificações feitas, entende-se que as avaliações feitas foram suficientemente satisfatórias para a maioria dos avaliados. Todas as respostas dadas a esta pergunta confirmam que na maioria dos casos, as classificações e emoções feitas são fiéis a entrada.

Apesar dos resultados serem positivos para o tipo de classificador criado, é importante ressaltar a subjetividade na fala e compreensão de textos. Como descrito no capítulo 2, a análise de texto pode ter uma grande margem de erros, pois é possível dar ao texto mais de uma interpretação possível para uma mesma palavra, por exemplo. Diversas palavras soltas, ou mesmo um texto curto analisado, relacionado a um sentimento específico pode ser tirado do contexto desejado.

Após a seção de avaliação do classificador e a avaliação das interações feitas e expressa pelos gráficos e quadros apresentados, o próximo capítulo apresenta as conclusões e trabalhos futuros.

6 CONCLUSÕES E TRABALHOS FUTUROS

Esse capítulo apresenta as conclusões adquiridas a partir do desenvolvimento do presente trabalho e os trabalhos futuros relacionados que poderão ser desenvolvidos a partir deste.

6.1 CONCLUSÕES

A análise de sentimento ou mineração de opinião é um tópico extremamente significativo. Na atualidade, devido a disseminação de conhecimento e a existência de diversas bibliotecas que auxiliam esse tipo de análise em diversas linguagens de programação diferentes, é possível construir um classificador estatístico baseado em aprendizagem supervisionada sem qualquer conhecimento prévio.

Durante o desenvolvimento do trabalho a pergunta de pesquisa “Como construir um modelo de análise e classificação de sentimentos em textos de forma a construir um perfil baseado em análises anteriores?”, desenvolvida no capítulo um, foi a base para a construção dos demais capítulos e do protótipo funcional. De acordo com os estudos realizados, principalmente no capítulo dois, viu-se que existem diversas formas diferentes de analisar e classificar textos. O atual trabalho utilizou o algoritmo naive bayes para analisar as entradas fornecidas. Depois que uma classificação era feita, todas as classes da predição, raiva (*anger*), medo (*fear*), alegria (*joy*), amor (*love*), tristeza (*sadness*), e surpresa (*surprise*) foram processadas através de um algoritmo criado a partir de uma função sigmoide, e o perfil com o total de todas as classificações feitas era atualizado. Assim, uma emoção total predominante era calculada a partir do valor mais alto no perfil.

Após a finalização do capítulo de desenvolvimento entendeu-se a viabilidade do projeto analisando individualmente os objetivos gerais e específicos concluídos. Conseguiu-se mensurar a conclusão do objetivo geral, sendo ele “desenvolver uma proposta de solução para tornar possível a polarização de sentimentos predominantes em textos, bem como arquivar os

textos polarizados empregando-os para novas classificações” a partir da conclusão dos demais objetivos específicos.

O primeiro objetivo específico tem como foco identificar os métodos e técnicas de análise de sentimentos para apoiar a classificação de texto. O capítulo dois, que compete o referencial teórico, faz o levantamento de diversos métodos de análise de sentimentos existentes, bem como cria uma significativa dissertação sobre o processamento de linguagem natural e seus problemas. Já na etapa de desenvolvimento, descrita no capítulo cinco, a criação da modelagem de arquitetura do sistema faz referência a esse mesmo levantamento no contexto funcional, onde o conjunto elaborado no capítulo de referencial teórico serviu como base para a pesquisa das técnicas de análise de sentimento que seriam utilizadas.

Ao fim do desenvolvimento do capítulo três, onde se definiu as informações relativas à metodologia de pesquisa, caracterizando o que seria feito e definindo as atividades metodológicas, bem como as delimitações do trabalho, inicou-se o desenvolvimento dos requisitos funcionais, não funcionais e as regras de negócio do protótipo funcional, já no capítulo, quatro, expondo o cenário da aplicação e concluindo o segundo objetivo específico desenvolvido.

Mediante ao desfecho da modelagem dos requisitos, bem como a aplicação da primeira fase da metodologia ICONIX para o desenvolvimento de importantes artefatos como os casos de uso, os protótipos de tela, o diagrama de caso de uso da aplicação, o diagrama de sequência da aplicação, e os diagramas de atividade que contextualizam a síntese de fase de treinamento e o processo de classificação, criação de perfil e resposta, finalizou-se o terceiro objetivo específico.

O protótipo funcional foi desenvolvido a partir da modelagem criada no capítulo quatro, e é descrito no capítulo cinco, onde é apresentado as etapas do processo de desenvolvimento, assim como as ferramentas utilizadas, o histórico de desenvolvimento e a apresentação do protótipo funcional. Ao fim da etapa de desenvolvimento foi concluído o quarto objetivo específico.

A avaliação feita no capítulo cinco finaliza o quinto e último objetivo específico, que é avaliar o protótipo funcional a partir de métricas de classificação e avaliar o protótipo funcional a partir de interações com o usuário.

Após o desenvolvimento da matriz de confusão e das variáveis de métrica de avaliação, somos capazes de avaliar positivamente o classificador elaborado, devido aos valores de acurácia, por exemplo, flutuarem entre 0.875 e 0.96. Outros valores considerados também se apresentam satisfatórios para o algoritmo utilizado, como a precisão, que apesar de

estar em baixa em algumas classes, apresentando valores como 0.31, também possui valores assertivos em outras classes, como 0.91.

Concordantemente com os resultados apresentados pela avaliação do classificador, a avaliação qualitativa realizada por um número limitado de usuários em um ambiente controlado, através de um formulário, também foi bastante positiva. No total, cinco entre sete dos usuários consideraram como objetivo principal da aplicação realizar algum tipo de análise ou avaliação a partir das entradas enviadas. Quanto a fidelidade das emoções expressas na aplicação após de uma entrada, todos dos usuários disseram que em todos ou na maioria dos casos as emoções foram expressas corretamente. Relativo à mudança de sentimento predominante para o usuário, 100% das avaliações disseram que após duas entradas o sentimento predominante mudou alguma vez e 57,1% das avaliações disseram que após cinco entradas, o sentimento mudou alguma vez. Isso conclui que, a partir da construção do perfil proposto pela aplicação, 42,9% dos usuários criaram sentimentos fortes que não puderem ser revertidos de forma fácil.

De acordo com os resultados apresentados, corroborados pela avaliação apresentada consegue-se entender o trabalho como assertivo em relação a seus objetivos. O código fonte do protótipo funcional desenvolvido foi disponibilizado e arquivado pelo autor (BOTELHO, 2021) na ferramenta GitHub.

6.2 TRABALHOS FUTUROS

Alguns trabalhos futuros foram definidos foram definidos ao término do desenvolvimento do protótipo funcional:

- Melhorar a análise de emoções: Expandir e modificar o *dataset* com emoções para melhorar a precisão das classes individualmente.
- Criar uma forma de autenticação: Modificar a aplicação para que seja necessário um login, de forma a oferecer experiências personalizadas.
- Melhorar a base de respostas: Criar mais marcadores de inteligência artificial para que seja possível ter melhores diálogos.
- Melhorar a criação de perfil: Fazer com que seja possível a mesclagem de sentimentos predominantes a partir de um tipo de roda das emoções.

REFERÊCIAS

ARANTES, Lucas de Oliveira. **Documentação semântica no apoio à integração de dados e rastreabilidade**. 2010. 184 f. Dissertação (Mestrado) - Curso de Informática, Universidade Federal do Espírito Santo, Vitória, 2010. Disponível em: http://repositorio.ufes.br/handle/10/6396. Acesso em: 22 nov. 2020.

ARRUDA, Guilherme Ferraz de. **Mineração de dados em redes complexas: estrutura e dinâmica**. 2013. 129 f. Dissertação (Mestrado) - Curso de Ciências de Computação e Matemática Computacional, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2013. Disponível em: https://teses.usp.br/teses/disponiveis/55/55134/tde-25062013-085958/pt-br.php. Acesso em: 22 nov. 2020.

AVILA, Gustavo Vianna. **Análise de sentimento para textos curtos**. 2017. 76 f. Dissertação (Mestrado) - Curso de Modelagem Matemática da Informação, Fundação Getúlio Vargas, Rio de Janeiro, 2017. Disponível em: http://bibliotecadigital.fgv.br/dspace/handle/10438/18177. Acesso em: 10 out. 2020.

BALAGE FILHO, Pedro Paulo. **Aspect extraction in sentiment analysis for Portuguese**. 2017. 74 f. Tese (Doutorado) - Curso de Ciências de Computação, Instituto de Ciências Matemáticas e de Computação, São Carlos, 2017. Disponível em: https://teses.usp.br/teses/disponiveis/55/55134/tde-05122017-140435/pt-br.php. Acesso em: 6 out. 2020.

BARCHI, Paulo Henrique. **Machine and deep learning applied to galaxy morphology**. 2020. 107 f. Tese (Doutorado) - Curso de Applied Computing, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2020. Disponível em: http://www.lareferencia.info/vufind/Record/BR_841a431f0dded6e14b4b72585132f1b0. Acesso em: 22 nov. 2020.

BARROS, Alex de Paula. **A flexible compositional approach to word sense disambiguation**. 2018. 41 f. Dissertação (Mestrado) - Curso de Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, 2018. Disponível em: https://repositorio.ufmg.br/handle/1843/SLSC-BBKGTM. Acesso em: 6 out. 2020.

BASTOS, Erick Casagrande. **Documentação semântica na gerência de projetos**. 2015. 128 f. Dissertação (Mestrado) - Curso de Informática, Centro Tecnológico, Universidade Federal do Espírito Santo, Vitória, 2015. Disponível em: http://repositorio.ufes.br/handle/10/1999. Acesso em: 22 nov. 2020.

BATISTA, Gustavo Enrique de Almeida Prado Alves. **Pré-processamento de dados em aprendizado de máquina supervisionado**. 2003. 232 f. Tese (Doutorado) - Curso de Ciências de Computação e Matemática Computacional, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2003. Disponível em: https://teses.usp.br/teses/disponiveis/55/55134/tde-06102003-160219/pt-br.php. Acesso em: 22 nov. 2020.

BATRES, Eduardo Jaime Quirós; OLIVEIRA, Alcione de Paiva; GABRIELLI, Bruno Ventorim; AMORIM, Vinci Pegoretti; MOREIRA, Alexandra. Uso de ontologias para a extração de informações em atos jurídicos em uma instituição pública 10.5007/1518-2924.2005v10n19p73. **Encontros Bibli**: revista eletrônica de biblioteconomia e ciência da informação, [S.L.], v. 10, n. 19, p. 73-88, 15 abr. 2005. Universidade Federal de Santa Catarina. Disponível em: http://dx.doi.org/10.5007/1518-2924.2005v10n19p73. Acesso em: 22 nov. 2020.

BENGIO, Yoshua. Deep Learning of Representations for Unsupervised and Transfer Learning. **Journal Of Machine Learning Research**, [S.L.], v. 27, n. 1, p. 17-37, jun. 2012. Disponível em: http://proceedings.mlr.press/v27/bengio12a.html. Acesso em: 22 nov. 2020.

BLAZ, Cássio Castaldi Araujo. **Análise de Sentimentos em Tíquetes para o Suporte de TI**. 2017. 96 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017. Disponível em: https://lume.ufrgs.br/handle/10183/172455. Acesso em: 6 out. 2020.

BONA, Cristina. **Avaliação de processos de software: um estudo de caso em XP e ICONIX**. 2002. 122 f. Dissertação (Mestrado) - Curso de Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2002. Disponível em: https://repositorio.ufsc.br/xmlui/handle/123456789/82842. Acesso em: 21 abr. 2021.

BORGES NETO, José; MERCER, José Luiz da Veiga. Nos bastidores da análise sintática tradicionalz. **Letras**, Santa Maria, n. 5, p. 86-100, jun. 1993. Universidade Federal de Santa Maria. Disponível em: https://periodicos.ufsm.br/letras/article/view/11452. Acesso em: 22 nov. 2020.

BOTELHO, Nathan. **Enbot/readme**. Disponível em: <https://github.com/enbot/readme>. Acesso em: 01 jun. 2021.

CALEGARI, Newton Juniano. **Proposta de uma ferramenta de anotação semântica para publicação de dados estruturados na Web**. 2016. 87 f. Dissertação (Doutorado) - Curso de Tecnologia da Inteligência e Design Digital, Pontifícia Universidade Católica de São Paulo, São Paulo, 2016. Disponível em: https://tede2.pucsp.br/handle/handle/18992. Acesso em: 22 nov. 2020.

CECI, Flávio. **Um modelo baseado em casos e ontologia para apoio à tarefa intensiva em conhecimento de classificação com foco na análise de sentimentos**. 2015. 211 f. Dissertação (Mestrado) - Curso de Engenharia do Conhecimento, Universidade Federal de Santa Catarina, Florianópolis, 2015. Disponível em: https://repositorio.ufsc.br/handle/123456789/158856. Acesso em: 22 nov. 2020.

CECI, Flavio; ALVAREZ, Guilherme Martins; GONÇALVES, Alexandre Leopoldo. Análise de Sentimento e Mineração de Opinião: uma revisão bibliométrica da literatura. **Espacios**, Caracas, Venezuela, v. 14, n. 38, p. 12-27, out. 2016. Disponível em: https://www.revistaespacios.com/a17v38n14/17381412.html. Acesso em: 10 out. 2020.

CECI, Flavio; WOSZEZENKI, Cristiane Raquel; GONÇALVES, Alexandre Leopoldo. O uso de anotações semânticas e ontologias para a classificação de documentos. **International Journal Of Knowledge Engineering And Management**, Florianópolis, v. 3, n. 5, p. 1-14,

jun. 2014. Disponível em: https://www.researchgate.net/publication/260055399_O_USO_DE_ANOTACOES_SEMANTICAS_E_ONTOLOGIAS_PARA_A_CLASSIFICACAO_DE_DOCUMENTOS. Acesso em: 22 nov. 2020.

CORNELL. **Movie Review Data**. Disponível em: <https://www.cs.cornell.edu/people/pabo/movie-review-data/>. Acesso em: 21 abr. 2021.

DOCKER. **Empowering App Development for Developers**. Disponível em: <https://www.docker.com/>. Acesso em: 21 abr. 2021.

EKMAN, Paul. An argument for basic emotions. **Cognition & emotion**, v. 6, n. 3-4, p. 169–200, 1992. Disponível em: http://www.paulekman.com/wp-content/uploads/2013/07/An-Argument-For-Basic-Emotions.pdf. Acesso em: 10 out. 2020.

FERREIRA, Jeferson. **Validação do Fluxo Excepcional A Partir do Diagrama de Atividades da Uml**. 2011. 147 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2011. Disponível em: http://repositorio.unicamp.br/jspui/handle/REPOSIP/275746. Acesso em: 21 abr. 2021.

FIGUEIREDO, Fabio Soares. **Construção de evidências para classificação automática de textos**. 2008. 73 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2008. Disponível em: https://repositorio.ufmg.br/handle/1843/RVMR-7L3NSY. Acesso em: 22 nov. 2020.

FOSCHIEIRA, Silvia Matturro Panzardi. **A semântica da emoção: Um estudo contrastivo a partir do framenet e da roda das emoções**. 2012. 287 f. Tese (Doutorado) - Curso de Linguística Aplicada, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2012. Disponível em: http://www.repositorio.jesuita.org.br/handle/UNISINOS/4220. Acesso em: 10 out. 2020.

GARCIA, Vinícius Veloso de Mello. **JSPY: um modelo objetivo para compreensão de linguagem natural**. 2017. 112 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2017. Disponível em: https://repositorio.ufmg.br/handle/1843/ESBF-AMDPHP. Acesso em: 22 nov. 2020.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas S.A, 2002. 176 p. Disponível em: http://www.uece.br/nucleodelinguasitaperi/dmdocuments/gil_como_elaborar_projeto_de_pesquisa.pdf. Acesso em: 29 nov. 2020.

GIT. **Git**. Disponível em: <https://git-scm.com/>. Acesso em: 21 abr. 2021.

GITHUB. **GitHub: Where the world builds software**. Disponível em: <https://github.com/>. Acesso em: 21 abr. 2021.

GOOGLE. **Angular - What is Angular?**. 2021a. Disponível em: <https://angular.io/guide/what-is-angular>. Acesso em: 21 abr. 2021.

GOOGLE. **Formulários Google: crie e analise pesquisas gratuitamente**. 2021b. Disponível em: <https://www.google.com/intl/pt-BR/forms/about/>. Acesso em: 21 abr. 2021.

GOVINDARAJ, Praveen. **Emotions dataset for NLP**. Disponível em: <https://www.kaggle.com/praveengovi/emotions-dataset-for-nlp>. Acesso em: 21 abr. 2021.

GUERRA, Pedro Andrés Vilanova. **On UML statechart with variabilities**. 2012. 93 f. Dissertação (Mestrado) - Curso de Informática, Universidad de La República, Montevideo, 2012. Disponível em: https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/2970. Acesso em: 21 abr. 2021.

LIU, Bing. **Sentiment Analysis and Opinion Mining**. California (Usa): Morgan & Claypool Publishers, 2012. 168 p. Versão rascunho. Disponível em: https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf. Acesso em: 10 out. 2020.

LIU, Bing; ZHANG, Lei. A Survey of Opinion Mining and Sentiment Analysis. **Mining Text Data**, Boston, v. 13, n. 1, p. 415-463, jan. 2012. Disponível em: https://link.springer.com/chapter/10.1007%2F978-1-4614-3223-4_13. Acesso em: 10 out. 2020.

LUJÁN-MORA, Sergio. **Data warehouse design with UML**. 2005. 348 f. Tese (Doutorado) - Curso de Lenguajes y Sistemas Informáticos, Universidad de Alicante, San Vicente del Raspeig, 2005. Disponível em: http://hdl.handle.net/10045/11196. Acesso em: 21 abr. 2021.

MAIA, Luiz Claudio Gomes. **Uso de sintagmas nominais na classificação automática de documentos eletrônicos**. 2008. 158 f. Tese (Doutorado) - Curso de Ciência da Informação, Universidade Federal de Minas Gerais, Belo Horizonte, 2008. Disponível em: https://repositorio.ufmg.br/handle/1843/ECID-7NXJKZ. Acesso em: 22 nov. 2020.

MICROSOFT. **TypeScript: Typed JavaScript at Any Scale**. Disponível em: <https://www.typescriptlang.org/>. Acesso em: 21 abr. 2021.

MIGUEL, Fabiano Koich. Psicologia das emoções: uma proposta integrativa para compreender a expressão emocional. **Psico-Usf**, Itatiba, v. 20, n. 1, p. 153-162, abr. 2015. Disponível em: https://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-82712015000100015. Acesso em: 10 out. 2020.

MORGADO, Gisele P.; GESSER, Ingrid; SILVEIRA, Denis S.; MANSO, Fernando S. P.; LIMA, Priscila M. V.; SCHMITZ, Eber A.. Práticas do CMMI® como regras de negócio. **Production**, [S.L.], v. 17, n. 2, p. 383-394, ago. 2007. FapUNIFESP (SciELO). Disponível em: http://dx.doi.org/10.1590/S0103-65132007000200013. Acesso em: 21 abr. 2021.

NESELLO, Priscila. **Implicações do fenômeno Big Data na análise para inteligência estratégica**. 2014. 184 f. Dissertação (Mestrado) - Curso de Administração, Universidade de Caxias do Sul, Caxias do Sul, 2014. Disponível em: https://repositorio.ucs.br/handle/11338/822. Acesso em: 6 out. 2020.

NLTK PROJECT. **Natural Language Toolkit**. Disponível em: <https://www.nltk.org/>. Acesso em: 21 abr. 2021.

OLIVEIRA, Bruno Stefani Ferreira de. **A relação da consciência morfológica com o processamento morfológico e a leitura**. 2015. 84 f. Dissertação (Mestrado) - Curso de Psicologia, Ich – Instituto de Ciências Humanas, Universidade Federal de Juiz de Fora, Juiz de Fora, 2015. Disponível em: https://repositorio.ufjf.br/jspui/handle/ufjf/311. Acesso em: 24 out. 2020.

OLIVEIRA, Mônica Aparecida de. **As representações sociais de tecnologistas e pesquisadores sobre a atividade de pesquisa.** 2013. 160 f. Dissertação (Mestrado) - Curso de Formação, Políticas e Práticas Sociais, Universidade de Taubaté, Taubaté, 2013. Disponível em: http://repositorio.unitau.br/jspui/handle/20.500.11874/927. Acesso em: 6 out. 2020.

PANDORABOTS. **Pandorabots Documentation**. Disponível em: <https://pandorabots.com/docs/faq/>. Acesso em: 21 abr. 2021.

PAULINO, Hideljundes Macedo. **Avaliação e monitoramento de políticas públicas: criação de um modelo sistêmico aplicado ao Instituto de Assistência Técnica e Extensão Rural do Rio Grande do Norte (EMATER-RN)**. 2014. 105 f. Dissertação (Mestrado) - Curso de Administração, Universidade Federal do Rio Grande do Norte, Natal, 2014. Disponível em: https://repositorio.ufrn.br/jspui/handle/123456789/16918. Acesso em: 21 abr. 2021.

PEREIRA, Janaina Cruz. **Melhoramento de docking-based virtual screening usando abordagem de deep learning**. 2017. 170 f. Tese (Doutorado) - Curso de Biologia Computacional de Sistemas, Instituto Oswaldo Cruz, Rio de Janeiro, 2017. Disponível em: https://www.arca.fiocruz.br/handle/icict/23812. Acesso em: 22 nov. 2020.

PEREIRA JUNIOR, Alvaro Rodrigues. **Um modelo para prototipagem rápida de aplicações de mineração na web**. 2008. 178 f. Tese (Doutorado) - Curso de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2008. Disponível em: https://repositorio.ufmg.br/handle/1843/RVMR-7P8NTM. Acesso em: 22 nov. 2020.

PINHEIRO, Roberto Hugo Wanderley. **Seleção de características para problemas de classificação de documentos**. 2011. 101 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2011. Disponível em: https://repositorio.ufpe.br/handle/123456789/2459. Acesso em: 22 nov. 2020.

PYTHON SOFTWARE FOUNDATION. **Welcome to Python.org**. Disponível em: <https://www.python.org/>. Acesso em: 21 abr. 2021.

RODRIGUES, Paulo César. Psicologia, Metafísica e Literatura: a Descrição dos Sentimentos Profundos em Bergson. **Trans/form/ação: Revista de Filosofia**, Marília, v. 36, n. 1, p. 81-100, jan./abr. 2013. Disponível em: https://revistas.marilia.unesp.br/index.php/transformacao/article/view/2917. Acesso em: 10 out. 2020.

ROSA, Clayton Wilhelm da. **A Combinator based, certifiable, parsing framework**. 2019. 100 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2019. Disponível em: https://repositorio.ufpe.br/handle/123456789/35363. Acesso em: 22 nov. 2020.

SEMOLINI, Robinson. **Support vector Machines, Inferência Transdutiva e o Problema de Classificação**. 2002. 142 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas, 2002. Disponível em: http://repositorio.unicamp.br/jspui/handle/REPOSIP/262026. Acesso em: 24 out. 2020.

SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da Pesquisa e Elaboração de Dissertação**. 4. ed. Florianópolis: Universidade Federal de Santa Catarina, 2005. 139 p. Disponível em: https://projetos.inf.ufsc.br/arquivos/Metodologia_de_pesquisa_e_elaboracao_de_teses_e_dissertacoes_4ed.pdf. Acesso em: 29 nov. 2020.

SILVA, Nadia Felix Felipe da. **Análise de sentimentos em textos curtos provenientes de redes sociais**. 2016. 112 f. Tese (Doutorado) - Curso de Ciências de Computação e Matemática Computacional, Instituto de Ciências Matemáticas e de Computação, São Carlos, 2020. Disponível em: https://teses.usp.br/teses/disponiveis/55/55134/tde-27092016-143947/pt-br.php. Acesso em: 6 out. 2020.

SILVA, Nelson Gutemberg Rocha da. **PairClassif - Um Método para Classificação de Sentimentos Baseado em Pares**. 2013. 80 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2013. Disponível em: https://repositorio.ufpe.br/handle/123456789/11441. Acesso em: 6 out. 2020.

SOUSA, Thiago Carvalho de. **Um processo de desenvolvimento orientado a objetos com suporte à verificação formal de inconsistências**. 2013. 315 f. Tese (Doutorado) - Curso de Engenharia de Computação, Escola Politécnica da Universidade de São Paulo, São Paulo, 2013. Disponível em: https://teses.usp.br/teses/disponiveis/3/3141/tde-21102014-113929/pt-br.php. Acesso em: 21 abr. 2021.

STANFORD. **Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank**. Disponível em: <https://nlp.stanford.edu/sentiment/code.html>. Acesso em: 21 abr. 2021.

VILLEGAS, Paulo. **Python aiml**. Disponível em: <https://pypi.org/project/python-aiml/>. Acesso em: 21 abr. 2021.

WAJZENBERG, Alberto. A TECNOLOGIA DE INFORMAÇÃO E A ADMINISTRAÇÃO DE RECURSOS HUMANOS. **Cadernos Ebap**, Rio de Janeiro, v. 1, n. 90, p. 1-18, jan. 1998. Disponível em: http://bibliotecadigital.fgv.br/dspace/handle/10438/12898. Acesso em: 10 out. 2020.

WARRINER, A. B.; KUPERMAN, V.; BRYSBAERT, M. Norms of valence, arousal, and dominance for 13,915 English lemmas. **Behavior research methods**, v. 45, n. 4, p. 1191–207, dez. 2013. Disponível em: https://doi.org/10.3758/s13428-012-0314-x. Acesso em: 10 out. 2020.

WILKENS, Rodrigo Souza. **A study of the use of natural language processing for conversational agents**. 2016. 83 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016. Disponível em: https://lume.ufrgs.br/handle/10183/142158. Acesso em: 24 out. 2020.

XAVIER, Clarissa Castellã. **Learning non-verbal relations under open information extraction paradigm**. 2014. 219 f. Tese (Doutorado) - Curso de Ciência da Computação, Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2014. Disponível em: http://tede2.pucrs.br/tede2/handle/tede/5275. Acesso em: 22 nov. 2020.

ZANCHIN, Janete. **Competências do trabalhador do conhecimento: Um estudo multicasos**. 2001. 92 f. Dissertação (Mestrado) - Curso de Administração, Universidade Federal de Santa Catarina, Florianópolis, 2001. Disponível em: https://repositorio.ufsc.br/xmlui/handle/123456789/82039. Acesso em: 6 out. 2020.

APÊNDICE A – Cronograma do desenvolvimento

Cronograma TCC – Aluno: Nathan Botelho – Orientador: Flávio Ceci

ATIVIDADE	Dezembro - Fevereiro	Março				Abril				Maio				Junho				Julho	
Semanas		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Revisão capítulo 1-3																			
Levantamento de requisitos																			
Escolha de ferramentas																			
Redação Capítulo 4																			
Entrega Capítulo 4																			
Modelagem de software																			
Desenvolvimento do software																			
Testes do software																			
Avaliação do software																			
Redação Capítulo 5																			
Entrega Capítulo 5																			
Redação Capítulo 6																			
Entrega Capítulo 6																			
Conclusões/ Resumo																			
Entrega Monografia																			
Apresentação																			
Defesa																			
Correções																			
Entrega versão final																			