**UNIVERSIDADE DO SAGRADO CORAÇÃO**

**LEONARDO ROBERTO GAZZIRO**

# SISTEMA DE ANÁLISE DE SENTIMENTOS EM REDES SOCIAIS APLICADA À PRESTAÇÃO DE SERVIÇOS UTILIZANDO O ALGORITMO DE CLASSIFICAÇÃO DE NAIVE BAYES

BAURU
2016

**LEONARDO ROBERTO GAZZIRO**

# SISTEMA DE ANÁLISE DE SENTIMENTOS EM REDES SOCIAIS APLICADA À PRESTAÇÃO DE SERVIÇOS UTILIZANDO O ALGORITMO DE CLASSIFICAÇÃO DE NAIVE BAYES

Trabalho de conclusão de curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob orientação do Prof. Dr. Elvio Gilberto da Silva.

BAURU
2016

Gazziro, Leonardo Roberto

G291s

Sistema de análise de sentimentos em redes sociais aplicada à prestação de serviços utilizando o algoritmo de classificação de Naive Bayes / Leonardo Roberto Gazziro. -- 2016.

73f. : il.

Orientador: Prof. Dr. Elvio Gilberto da Silva.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração – Bauru – SP.

**LEONARDO ROBERTO GAZZIRO**

# SISTEMA DE ANÁLISE DE SENTIMENTOS EM REDES SOCIAIS APLICADA À PRESTAÇÃO DE SERVIÇOS UTILIZANDO O ALGORITMO DE CLASSIFICAÇÃO DE NAIVE BAYES

Trabalho de conclusão de curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob orientação do Prof. Dr. Elvio Gilberto da Silva.

Banca examinadora:

_____
Prof. Dr. Elvio Gilberto da Silva
Universidade do Sagrado Coração

_____
Prof. Me. Patrick Pedreira Silva
Universidade do Sagrado Coração

_____
Prof. Me. Henrique Pachioni Martins
Universidade do Sagrado Coração

Bauru, 30 de Novembro de 2016.

Dedico este trabalho de conclusão de curso à minha família, a minha namorada e amigos.

## AGRADECIMENTOS

Agradeço primeiramente a meus pais, Vanderlei Roberto Gazziro e Eliana Maria Basso Gazziro, e minha irmã Leticia Fernanda Gazziro pelo apoio nos estudos e nos momentos difíceis.

Agradeço ao meu orientador Prof. Dr. Elvio Gilberto da Silva por todo o apoio durante o processo de desenvolvimento deste trabalho e todos os conhecimentos passados. Agradeço também aos outros professores que fizeram parte da minha formação e sempre estiveram comprometidos com passar o conhecimento adiante e incentivar o desenvolvimento pessoal de cada um.

Agradeço a meus amigos Dylan Martins Janine de Andrade, Filipe Rosa da Silva, Camila Pellizon Floret, Victor Grava Leal e Luan Matheus Marchiori, pelo apoio em todos os momentos difíceis, e todos os momentos em que estivemos juntos durante todos esses anos.

"Quando inovamos, devemos estar preparados para ouvir as pessoas dizendo que estamos loucos." (Larry Ellison)

## RESUMO

Diante do cenário atual de grande competitividade entre as empresas, a posse de informações estratégicas tornou-se uma vantagem valiosa para emplacar produtos, marcas e até mesmo prestação de serviços. Nos dias atuais, a internet é o meio em que os consumidores mais expressam suas opiniões sobre a experiência com as organizações, expondo sentimentos positivos e negativos, contidos em cada postagem em sites, redes sociais, fóruns, bate-papos, blogs e outros. Esse novo comportamento, criou um novo nicho a ser explorado pelas empresas e por profissionais da computação, a fim de extrair informações que permitam saber, como por exemplo, como a prestação de serviços de uma empresa está posicionada no mercado, o que ela pode melhorar baseada na opinião de seu cliente, bem como, qual decisão tomar diante do panorama em que está inserida. Para isso, visto a precariedade das ferramentas atualmente disponíveis, faz-se necessário o desenvolvimento de softwares a fim de classificar a polaridade das opiniões sobre prestações de serviços, prover informações estratégicas e táticas para reduzir a incerteza associada a uma decisão e expor diretamente a atitude e pensamento dos clientes. A análise de sentimentos tem se tornado um importante tópico na Web, especialmente em redes sociais, com o desenvolvimento de aplicações para monitoramento de produtos e marcas, assim como a análise da repercussão de eventos importantes. Vários métodos e técnicas vêm sendo propostos de forma independente na literatura. Este projeto teve o objetivo de desenvolver um software para análise de sentimentos de usuários, utilizando o algoritmo de classificação Naive Bayes para fazer a classificando as mensagens publicadas na rede social Twitter, gerando informações a respeito do serviço prestado por uma empresa/organização. Com a classificação concluída, foram apresentados gráficos que mostram informações sobre a atual reputação do serviço prestado na rede social mais utilizada. O software obteve 83% de acertos ao processar doze opiniões criadas pelo próprio autor, resultado que revela resultado satisfatório diante do trabalho proposto e indica que pode ser um aliado das organizações.



**Palavras-chave:** Análise de Sentimentos. Inteligência Artificial. Mineração de Dados. Vantagem de Mercado. Mineração de Opinião.

## ABSTRACT

Given the current scenario of great competitiveness among companies, the possession of strategic information has become a valuable advantage for emplacing products, brands, and even providing services. Nowadays, the internet is the way that consumers most express their opinions about experiences with organizations, exposing positive and negative feelings, contained in every post on websites, social media, forums, chats, blogs and others. This new behavior has created a new niche to be explored by companies and computer professionals in order to extract information that allows to know, for example, how a company's services are positioned in the market, how it can improve based on it's clients opinions as well as which decision to take before the panorama in which it's inserted. To do so, given the precariousness of the tools currently available, it is necessary to develop a software to classify the polarity of opinions about services, to provide strategical and tactical information to reduce the uncertainty associated with a decision and directly expose the attitude and the thoughts of the costumers. The analysis of feelings has become an important topic on the Web, especially in social medias, with the development of applications to monitor products and brands as well as the analysis of the repercution of important events. Several methods and techniques have been independently proposed in the literature. This project aimed to develop a software for the analysis of users feelings using the Naive Bayes classification algorithm to classify the messages published on Twitter, giving information about the service provided by a company/organization. With the classification concluded, were presented graphics that shows informations about the current reputation of the service provided on the most used social media. The software obtained 83% of correct answers when processing twelve opinions created by the author, a result that shows a satisfactory result and indicates that it can be an ally of the organizations.



**Key-words:** Analysis of feelings; artificial intelligence; data mining; market advantage; opinion mining.

## LISTA DE ILUSTRAÇÕES

**SUMÁRIO**

# 1 INTRODUÇÃO

Com o grande crescimento e desenvolvimento constante da internet, dos microcomputadores, *smartphones*, *tablets* e outras tecnologias que facilitam na inclusão digital da população, é cada vez maior o número de pessoas que expõem suas opiniões em blogs, sites e principalmente em redes sociais, por sua vez, essas opiniões podem ser sobre os mais variados assuntos, como acontecimentos políticos, filmes, series, catástrofes naturais, criticas sociais e também opiniões relacionadas ao contato com a prestação de serviços por uma empresa, que de maneira muito rápida pode ser propagada a uma grande quantidade de usuários (CARVALHO FILHO et al., 2014).

Com essa popularização da internet, o ritmo de crescimento dessas informações torna-se difícil de ser acompanhado pelas corporações, que por sua vez, tem grande interesse nas mesmas. A extração e análise da opinião dos usuários pode gerar a empresa vantagem competitiva de mercado, pois através dela é possível medir a aceitação de um produto ou serviço no mercado, e com isso tomar as medidas cabíveis para desenvolvimento de novos produtos e melhoria dos já existentes (SILVA et al., 2015).

O acompanhamento manual dessas informações torna-se praticamente inviável devido ao grande crescimento das mesmas, com isso, faz-se necessário o uso da tecnologia para auxiliar esse processamento e, por isso, a necessidade de um *software* para extrair esses dados e processá-los, a fim de definir a polaridade de cada opinião, e posteriormente gerar meios para auxiliar as organizações a melhorarem sua participação no mercado (CARVALHO FILHO et al., 2014).

A análise de sentimentos ou mineração de opinião é uma subárea da Inteligência Artificial, que tem como objetivo extrair textos de linguagem natural, escritos por pessoas, o ponto de vista, a opinião e a emoção que o mesmo representa, fornecendo a organização informações sobre sua aceitação e qualidade no mercado; com isso a empresa pode tomar decisões importantes como investimentos em marketing, melhoria da qualidade dos produtos e serviços, entre outras melhorias que lhe forneceram uma posição de destaque (CARVALHO FILHO et al., 2014).

Dessa forma, com o crescimento desenfreado de informações na internet, esse trabalho teve como objetivo construir um *software* que faça a extração das

informações de prestação de serviços de uma organização, e processe as mesmas utilizando técnicas de Inteligência Artificial e análise de sentimentos, gerando ao final desse processo relatórios, gráficos e uma visão geral que auxiliará a empresa. A importância desse trabalho pauta-se na avaliação do fluxo de informações nas redes sociais, o que pode incrementar o negócio das empresas, ou até corrigir problemas do lançamento de um produto, por exemplo. Hoje, colocar uma campanha de publicidade na TV e não monitorar as redes sociais é muito perigoso. Antigamente, se havia alguma coisa estranha, se alguém não estava gostando do produto, isso virava uma opinião individual. Atualmente, um comentário ruim pode deflagrar uma reação negativa em outras pessoas, e o efeito ser contrário ao esperado. Há risco de os efeitos serem completamente inversos, porque o consumidor não é mais passivo.

# 2 OBJETIVOS

Apresenta-se a seguir o objetivo geral e específicos desta pesquisa.

## 2.1 OBJETIVO GERAL

Desenvolver um *software* para a análise de sentimentos de usuários em redes sociais, que classifique opiniões e gere informações a respeito do serviço prestado pela organização.

## 2.2 OBJETIVOS ESPECÍFICOS

a) pesquisar sobre a análise de sentimentos, seus algoritmos, metodologias, técnicas e limitações;
b) selecionar uma mídia social para extração de comentários a serem classificados;
c) construir um Corpus de comentários com opiniões reais a serem analisadas;
d) utilizar métodos e técnicas da engenharia de *software* para modelar a proposta;
e) desenvolver o módulo do *software* que vasculhará a mídia social escolhida em busca de comentários;
f) aplicar a análise de sentimentos sobre o corpus utilizando uma implementação do algoritmo de classificação Naive Bayes para gerar um modelo de classificação de opiniões;
g) desenvolver rotina que identificará a polaridade dessas opiniões a respeito da organização;
h) desenvolver gráficos para exibição das informações obtidas, possibilitando a tomada de decisão.

# 3 SOFTWARE

Um *software* consiste em um programa de computador, um aplicativo para celular ou qualquer outro tipo de sistema que sirva para manipular um hardware em forma de linhas de código, podendo ter uma interface para interação com o usuário, ou não (PRESSMAN et al., 2011).

> [...] software consiste em: (1) instrução (programa de computador) que, quando executadas, fornecem características, funções e desempenho desejado; (2) estrutura de dados que possibilitam aos programas manipular informações adequadamente; e (3) informação descritiva, tanto na forma impressa quando na virtual, descrevendo a operação e o uso dos programas (PRESSMAN, 2009, p. 32).

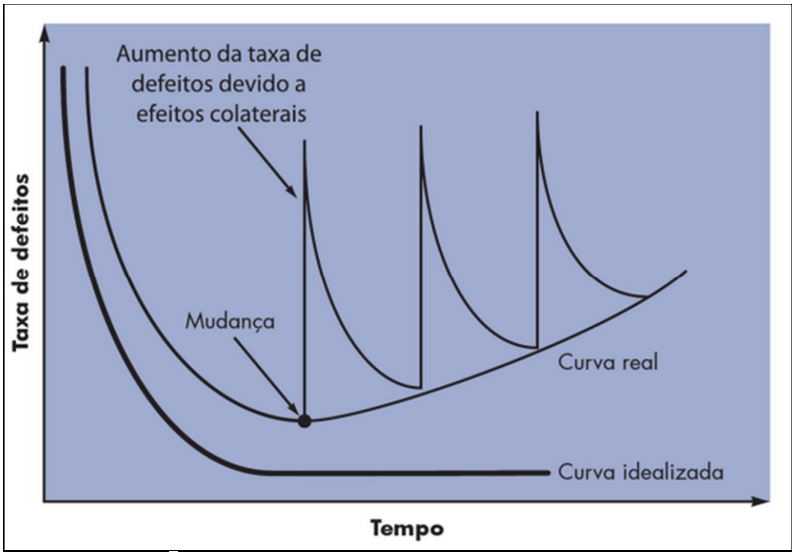
Para entender melhor a definição de *software* é preciso diferenciar sua construção de outras coisas que construímos. O *software* é um elemento de sistema logico, dessa forma, podemos dizer que suas características são diferentes do *hardware* (PRESSMAN et al., 2011).

Segundo Pressman (2011), as principais características do software se resumem em:

a) **Processo de engenharia**: o *software*, assim como o *hardware*, depende de um bom projeto para ter uma alta qualidade. Ambos têm seu foco na construção de um produto, porém são feitos de formas diferentes. O *software* tem seu custo concentrado no projeto, ou processo de engenharia, este processo não pode ser gerido como um simples projeto de fabricação.
b) **Desgaste**: o *software* é um sistema logico, ou seja, ele não é palpável, o que indica que não está suscetível aos males do ambiente, como chuva, sol e vento. Como pode ser observado na Figura 1, o *software* passa por uma alta taxa de defeitos logo no inicio do seu ciclo de vida. Essa taxa tende a diminuir, porém, toda a vez que o software passar por alterações, o mesmo tende a apresentar novos problemas, como pode ser observado na curva real ilustrada na Figura 1. Com isso podemos dizer que o *software* não se desgasta e se deteriora, não podendo assim ser substituído de maneira tão simples, tendo de passar por uma manutenção para sua correção.

c) **Software sob encomenda**: os componentes de um *software* devem ser projetados para ser utilizados em diversos projetos, mas essa ainda é uma prática que está começando a se difundir atualmente. Normalmente os *softwares* são projetados sob encomenda, e feitos apenas para um cliente, ou um nicho de mercado muito restrito.

Figura 1 - Gráfico de falhas em software




Fonte: Pressman (2011, p. 33).

## 3.1 TIPOS DE SOFTWARE

Para Pressman (2011), os *softwares* se encontram presentes nos mais diversos meios, isso inclui comércios, indústrias, pesquisa cientifica, nas engenharias, entre outros. De acordo com o autor, os tipos de software mais comuns existentes no mercado são:

a) **Software de sistema**: são programas construídos para atender outros programas, como por exemplo: compiladores, editores, componentes de sistemas operacionais, *drivers*, *softwares* de telecomunicações, e outros.
b) **Software de aplicação**: são programas construídos sob medida para atender as necessidades de um negocio especifico.
c) **Software científico/de engenharia**: são caracterizados por “number crunching” (para “processamento numérico pesado”). As aplicações

vão de astronomia, análise de tensões na indústria, dinâmica orbital e ônibus espacial e outros.

d) **Software embutido**: são *softwares* encontrados em aparelhos para sua manipulação, e que realizam a interação entre o usuário e a maquina para executar funções, como por exemplo, o controle do painel do forno micro-ondas.
e) **Software de inteligência artificial**: são sistemas que fazem a utilização de algoritmos numéricos, utilizados para solução de problemas complexos que visam imitar a inteligência humana em suas mais diversas formas, como exemplo, sistemas especialistas, tradutores automáticos, reconhecimento de padrões e etc.

# 4 TOMADA DE DECISÃO

A tomada de decisão é um fator muito importante diante do mercado competitivo da atualidade, dessa forma é a cada dia empresários de empresas de pequeno, médio e grande porte são obrigados a tomar decisões que serão cruciais para o desenvolvimento e crescimento da empresa, mas essas decisões também podem levar uma empresa a ruina.

Para auxiliar nessas decisões e torna-las mais eficientes as empresas vem dando cada vez mais valor aos sistemas de gestões, pois os mesmos vão fornecer informações que ajudam a tomar as melhores decisões em cada situação.

Segundo Simon (1960), citado por Laudon (2011), a tomada de decisão é composta de quatro etapas distintas, a inteligência, concepção, seleção e implementação.

A inteligência é o processo de identificação do problema, sua causa e suas consequências. A concepção é o processo de planejamento de soluções para o problema. A seleção consiste na escolha de um dos métodos planejados na concepção. Por fim a implementação é o momento em que a solução do problema será implantada na empresa e será usada de fato.

Diante disso os sistemas de análise de sentimentos têm o papel importante de fornecer informações sobre as opiniões de clientes de uma marca ou empresa, essas informações serão muito importantes para que eventuais falhas ou reclamações possam ser corrigidas ou melhoradas futuramente, assim como aquilo que é elogiado pode ser utilizado em forma de marketing para a promoção da empresa. Ao saber o que os clientes de um produto ou serviço acham do mesmo pode ser o ponto crucial para obter a vantagem competitiva e prosperar no mercado atual.

# 5 ENGENHARIA DE SOFTWARE

Conforme citado anteriormente, os *softwares* se fazem presentes nas mais diversas áreas de atuação da humanidade, e a busca por esses programas é a cada dia maior, e por sua vez, os usuários desses *softwares* buscam funcionalidades especificas e sugerem diversas modificações, de forma que ao criá-lo ou modificá-lo, uma vasta gama de pedidos dos mais variados tipos é feita, e nem sempre eles são compatíveis entre si, e em geral não costumam retratar a verdadeira raiz do problema a ser resolvido, dessa forma é de grande importância estudar, analisar e compreender os problemas e pedidos antes de desenvolver uma solução de *software* (PRESSMAN et al., 2011).

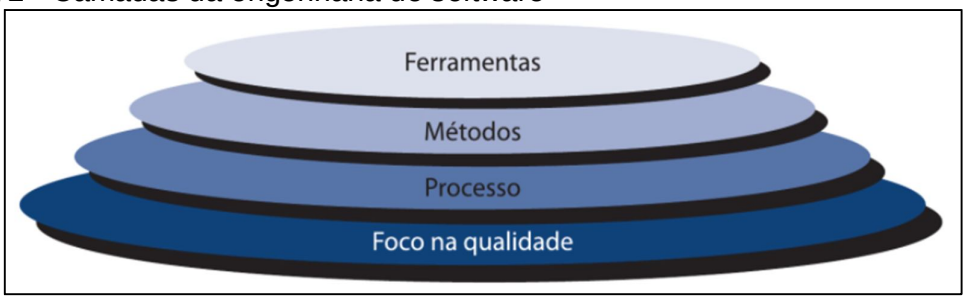
Os requisitos para a construção de um *software* tornam-se cada vez mais complexos, enquanto que as empresas, órgãos e organizações precisam a cada dia mais dele para se manterem “vivos” no mercado se considerarmos a grande quantidade de informações necessárias para manter a competitividade, e ao mesmo tempo tomar decisões mais sabias nos devidos momentos. Com isso a necessidade de modificações no *software* para se adequar ao cenário atual faz-se necessária, e, portanto, além do *software* ter de atender a todas as necessidades dos usuários fornecendo todas as informações necessárias, ele precisa ser passível de manutenção (PRESSMAN et al., 2011).

Com todas essas exigências para construir e manter um *software* surge à engenharia de *software*, a qual se torna muito importante para o sucesso dos projetos. Pressman (2011) define a engenharia de *software* como sendo

> [...] o estabelecimento e o emprego de sólidos princípios de engenharia de modo a obter software de maneira econômica, que seja confiável e funcione de forma eficiente em máquinas reais (PRESSMAN, 2009, p.39).

Apesar da definição apresentada por Fritz Bauer (PRESSMAN, 2009) na conferencia sobre engenharia de *software*, muitos aspectos não são citados, como a qualidade, satisfação do cliente e tempo de entrega. Assim como qualquer engenharia, o principal fundamento da engenharia de *software* deve ser a qualidade, e essa qualidade é obtida a partir das camadas que podem ser visualizadas na Figura 2, conforme destacam Pressman et al. (2011).

Figura 2 - Camadas da engenharia de software




Fonte: Pressman (2011, p. 39).

A seguir serão apresentadas as definições de cada camada ilustrada na Figura 2 em conformidade com Pressman et al. (2011).

A base da engenharia de *software* é o foco na qualidade, o aprimoramento contínuo dos processos leva a uma cultura que preza pela qualidade do software.

A camada de processos é de extrema importância para a engenharia de *software*, pois é ela que possibilita desenvolver *softwares* de forma racional e dentro do prazo. Por sua vez, também define a metodologia que será utilizada para entrega efetiva do *software*, controle do projeto e produção de produtos derivados como modelos, dados, manuais, relatórios e etc.

Os métodos da engenharia constituem a parte técnica do desenvolvimento de *software*, como o levantamento de requisitos, modelagem de projeto, construção do *software*, testes e etc.

As ferramentas auxiliam os métodos e processos, facilitando sua execução e melhorando a qualidade de sua execução.

## 5.1 LEVANTAMENTO DE REQUISITOS

Ainda, segundo o autor citado anteriormente, o levantamento de requisitos é uma parte do processo de desenvolvimento de *software* que consiste em conversar com o cliente, a fim de definir quais serão as funções que o sistema deve executar, e como elas devem ser executadas, compreendendo o objetivo do projeto.

A etapa do levantamento de requisitos é de extrema importância devido a ter grande participação no sucesso do projeto. É preciso saber tudo o que o cliente precisará para atendê-lo da maneira mais adequada, portanto, quando essa etapa é realizada com qualidade muitos problemas futuros são evitados, como, por exemplo,

a necessidade de reconstrução de um módulo do *software* porque o problema do cliente não foi entendido em sua totalidade, ou foi entendido de forma incorreta.

A grande complexidade dessa etapa fica por conta do cliente não possuir conhecimento técnico, dessa forma o profissional que executa esta etapa tem que atentar-se muito, e levantar a maior quantidade de informações possíveis, para assim “traduzir” as reais necessidades do cliente/usuário, e como elas devem ser projetadas para atenderem em sua totalidade, e consequentemente garantir a satisfação do usuário final.

## 5.2 MODELAGEM

A modelagem é uma técnica amplamente utilizada em diversas áreas do conhecimento humano. Para a construção de uma casa, prédio, avião, carro, levantamento/elucidação de requisitos, e muitas outras coisas são criados modelos, esse modelo nada mais é que uma simplificação da realidade (BOOCH; RUMBAUGH; JACOBSON et al., 2005).

Os modelos podem ser das mais variadas formas, ou seja, apresentando o nível de detalhamento desejado para o momento, e isso possibilita ao profissional que fará a construção do software ter uma visão geral do mesmo, possibilitando a previsão de falhas, reaproveitamento de código, simplificação de código e varias outras vantagens (BOOCH; RUMBAUGH; JACOBSON et al., 2005).

Para os autores supracitados, os modelos são construídos para compreendermos melhor o sistema que será projetado/desenvolvido. Os autores ainda destacam que, a partir da modelagem é possível se alcançar os seguintes objetivos:

a) Os modelos ajudam a visualizar o sistema como ele é ou como desejamos que ele seja;
b) os modelos permitem especificar a estrutura ou o comportamento de um sistema;
c) os modelos proporcionam um guia para a construção do sistema;
d) os modelos documentam as decisões tomadas.

Como pode ser observado, os modelos podem ser criados para as mais diversas funções, dentre as quais: construção da interface do *software*, camada de

dados, sequência de ações e outros. Essas medidas levam um projeto a ter maiores chances de sucesso, e por isso serão adotadas para o desenvolvimento desta proposta.

## 5.3 UML

A UML (*Unified Modeling Linguage*) ou Linguagem de Modelagem Unificada é uma linguagem destinada a visualizar, especificar, construir e documentar sistemas complexos. Podemos dizer que o UML é uma linguagem visual, pois a partir dos modelos gerados é feita a construção do código do programa (BOOCH; RUMBAUGH; JACOBSON et al., 2005).

A UML consiste em uma parte do desenvolvimento de *software*, a qual faz parte na modelagem, pois é através dela que podemos criar diversos modelos para as mais diversas funções de um *software*, como por exemplo, a interação do ser humano com a *interface*, o modelo de dados, as funções que o *software* deve desempenhar, as telas que interligam os módulos e etc. (BOOCH; RUMBAUGH; JACOBSON et al., 2005).

A UML é utilizada tanto na construção de sistemas simples até os mais complexos, dentre os quais podemos citar: telecomunicações, serviços bancários, venda e varejo, científicos, entre outros. Entretanto a UML não fica restrita só a modelagem de *software*, ela permite modelar sistemas como fluxos de trabalho, a estrutura e o comportamento de um sistema de saúde e seu projeto de hardware, etc. (BOOCH; RUMBAUGH; JACOBSON et al., 2005).

## 5.4 DIAGRAMAS UML

A UML é formada por diagramas, onde um diagrama UML é a representação gráfica que mostra os itens da linguagem e os relacionamentos entre eles. Esse diagrama consiste na representação de uma parte do sistema com os elementos que a compõem. Atualmente a UML dispõe de 13 tipos de diagramas com finalidades diferentes (BOOCH; RUMBAUGH; JACOBSON ET AL., 2005), os quais serão listados e explicados brevemente a seguir.

a) **Diagrama de Classe**: esse tipo de diagrama apresenta as classes, *interfaces* e colaborações de um sistema, e também a forma como elas

estão interligadas. São frequentes em sistemas que abordam o Paradigma de Orientação a Objeto (POO).

b) **Diagrama de Objeto**: mostra os objetos e relacionamentos entre eles. Representa a instância de itens encontrados no Diagrama de Classe, funcionando como um complemento para o mesmo.
c) **Diagrama de Componente**: abrange a visão da implementação de um projeto estático. Esse digrama é constituído por uma classe encapsulada e suas *interfaces*, portas e suas estruturas internas, conectada a componentes protegidos e conectores.
d) **Diagrama de Estrutura Composta**: possui as mesmas características de um Digrama de Componentes, porém, aplicado a qualquer classe.
e) **Diagrama de Caso de Uso**: exibe um conjunto de Casos de Uso, atores e seus relacionamentos. Esses diagramas são muito importantes para a modelagem de componentes do sistema e também para a organização do sistema.
f) **Diagrama de Sequência**: este diagrama enfatiza a ordenação temporal das mensagens, de acordo com a interação entre os conjuntos de objetos.
g) **Diagrama de Comunicação**: apresenta a organização estrutural dos objetos que enviam e recebem mensagens.
h) **Diagrama de Gráfico de Estado**: mostra uma maquina de estados, com estados, eventos, transições e atividades.
i) **Diagrama de Atividades**: Exibe o fluxo de controle, os dados de cada etapa de uma computação e a estrutura de um processo. É um diagrama muito importante para a modelagem das funções do sistema, dando ênfase ao fluxo de controle entre os objetos.
j) **Diagrama de Implantação**: mostra a configuração dos nós de processamento em tempo de execução, bem como, dos componentes que existem neles.
k) **Diagrama de Pacote**: ilustra a decomposição do próprio modelo em unidades organizacionais e suas dependências.
l) **Diagrama de Temporização**: mostra a interação entre os objetos em tempo real, e não a sequencia de mensagens entre eles.

m) **Diagrama de Visão Geral de Interação**: hibrido de um diagrama de atividade e um diagrama de sequência.

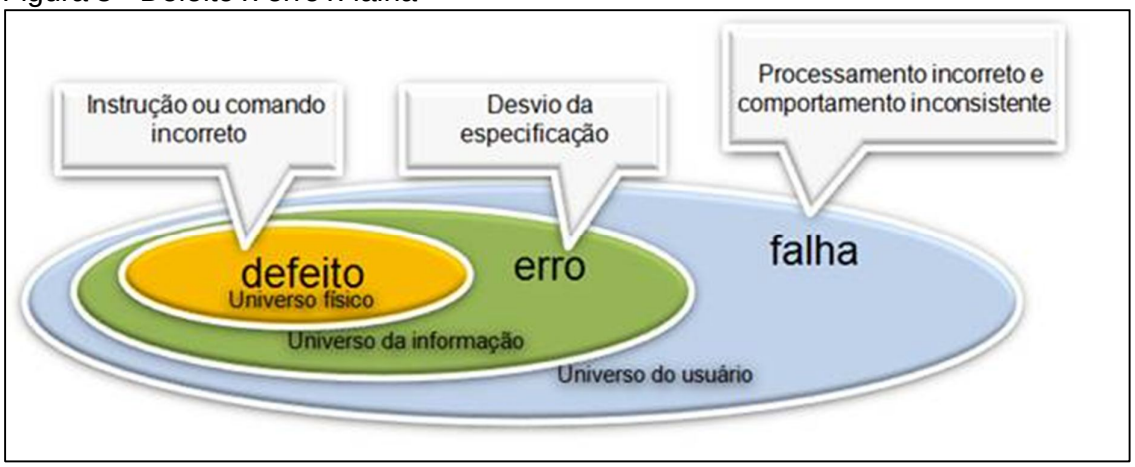
# 6 TESTE DE SOFTWARE

Antes de iniciar uma discussão sobre teste de *software* é preciso esclarecer alguns conceitos relacionados a essa atividade. Inicialmente, é preciso conhecer a diferença entre **Defeitos**, **Erros** e **Falhas**. As definições que serão usadas aqui seguem a terminologia padrão para Engenharia de *Software* do IEEE – *Institute of Electrical and Electronics Engineers* – (IEEE 610, 1990 citado por Claudio ([201-]).

a) **Defeito** é um ato inconsistente cometido por um indivíduo ao tentar entender uma determinada informação, resolver um problema ou utilizar um método ou uma ferramenta. Por exemplo, uma instrução ou comando incorreto.
b) **Erro** é uma manifestação concreta de um defeito num artefato de *software*. Diferença entre o valor obtido e o valor esperado, ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução de um programa constitui um erro.
c) **Falha** é o comportamento operacional do *software* diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos erros e alguns erros podem nunca causar uma falha.

A Figura 3 expressa a diferença entre esses conceitos.

Figura 3 - Defeito x erro x falha




Fonte: Claudio ([201-]).

Como pode ser observado na Figura 3, defeitos fazem parte do universo físico (a aplicação propriamente dita) e são causados por pessoas, por exemplo, através

do mal uso de uma tecnologia, podendo ocasionar a manifestação de erros em um produto, ou seja, a construção de um software de forma diferente ao que foi especificado (universo de informação). Por fim, os erros geram falhas, que são comportamentos inesperados em um software, que afetam diretamente o usuário final da aplicação (universo do usuário), e podem inviabilizar a utilização de um software.

Segundo Claudio ([201-]), teste de *software* é o processo de execução de um produto para determinar se ele atingiu suas especificações e funcionou corretamente no ambiente para o qual foi projetado, ao mesmo tempo em que faz parte de todo o processo de engenharia de *software*. O seu objetivo é revelar falhas em um produto, para que as causas dessas falhas sejam identificadas e possam ser corrigidas pela equipe de desenvolvimento antes da entrega final. Por conta dessa característica das atividades de teste, dizemos que sua natureza é “destrutiva”, e não “construtiva”, pois visa ao aumento da confiança de um produto através da exposição de seus problemas, porém antes de sua entrega ao usuário final.

Ainda segundo o autor supracitado, o conceito de teste de *software* pode ser compreendido através de uma visão intuitiva ou mesmo de uma maneira formal. Existem atualmente várias definições para esse conceito. De uma forma simples, testar um software significa verificar através de uma execução controlada se o seu comportamento ocorre de acordo com o especificado. O objetivo principal desta tarefa é revelar o número máximo de falhas dispondo do mínimo de esforço, ou seja, mostrar aos que desenvolvem se os resultados estão ou não de acordo com os padrões estabelecidos.

Para Pressman et al. (2011), o processo de testes visa encontrar falhas no sistema para corrigi-las antes de distribuir o sistema, ou de atualizar o *software* com novos recursos, portanto o sistema deve ser projetado e implementado pensando na facilidade da realização dos testes. Por sua vez os testes devem ser feitos levando em consideração certas características para que nada seja deixado para trás.

O conceito de testabilidade consiste em se medir o quão simples é o ato de testar um software. As características apresentadas seguir caracterizam um software a ser testável (PRESSMAN et al., 2011):

a) **Operabilidade:** um sistema projetado e implementado tendo em mente a qualidade, terá poucas falhas quando os testes forem realizados.

b) **Observabilidade:** quando é possível ver com clareza as entradas, saídas e variáveis do sistema fica mais fácil de detectar possíveis falhas.
c) **Controlabilidade:** entradas geram saídas especificas, e para cada tipo de saída existirá um tipo de entrada especifica. Se o engenheiro puder controlar essas entradas ficará mais fácil realizar os testes.
d) **Decomponibilidade:** o sistema é construído a partir de módulos e pode ser testado em partes.
e) **Simplicidade:** Quanto mais simples um sistema for, atingindo o objetivo, mais simples serão os testes.
f) **Estabilidade:** Quanto menos alterações o *software* tiver, menos testes precisarão ser feitos.
g) **Compreensibilidade:** Quanto mais informações estiverem disponíveis para o entendimento do *software*, mais eficazes serão os testes, isso inclui manuais organizados, detalhados e especificados.

A atividade de teste é composta por alguns elementos essenciais que auxiliam na formalização desta atividade, os quais serão apresentados a seguir.

a) **Caso de Teste:** descreve uma condição particular a ser testada e é composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado (CRAIG; JASKIEL, 2002).
b) **Procedimento de Teste:** é uma descrição dos passos necessários para executar um caso (ou um grupo de casos) de teste (CRAIG; JASKIEL, 2002).
c) **Critério de Teste:** serve para selecionar e avaliar casos de teste de forma a aumentar as possibilidades de provocar falhas ou, quando isso não ocorre, estabelecer um nível elevado de confiança na correção do produto (ROCHA et al., 2001). Os critérios de teste podem ser utilizados como:
    i. **Critério de Cobertura dos Testes:** permite a identificação de partes do programa que devem ser executadas para garantir a qualidade do software e indicar quando o mesmo foi suficientemente testado (RAPPS; WEYUKER, 1982). Ou seja, determinar o percentual de elementos necessários por um

critério de teste que foram executados pelo conjunto de casos de teste.

ii. **Critério de Adequação de Casos de Teste:** Quando, a partir de um conjunto de casos de teste T qualquer, ele é utilizado para verificar se T satisfaz os requisitos de teste estabelecidos pelo critério. Ou seja, este critério avalia se os casos de teste definidos são suficientes ou não para avaliação de um produto ou uma função (ROCHA et al., 2001).

iii. **Critério de Geração de Casos de Teste:** quando o critério é utilizado para gerar um conjunto de casos de teste T adequado para um produto ou função, ou seja, este critério define as regras e diretrizes para geração dos casos de teste de um produto que esteja de acordo com o critério de adequação definido anteriormente (ROCHA et al., 2001).

## 6.1 NÍVEIS DE TESTE DE SOFTWARE

O planejamento dos testes deve ocorrer em diferentes níveis e em paralelo ao desenvolvimento do *software* (Figura 4). Segundo Rocha et al. (2001) definimos que os principais níveis de teste de software são:

**a) Teste de Unidade:** também conhecido como teste unitário. Tem por objetivo explorar a menor unidade do projeto, procurando provocar falhas ocasionadas por defeitos de lógica e de implementação em cada módulo, separadamente. O universo alvo desse tipo de teste são os métodos dos objetos ou mesmo pequenos trechos de código.

**b) Teste de Integração:** visa provocar falhas associadas às interfaces entre os módulos quando esses são integrados para construir a estrutura do software que foi estabelecida na fase de projeto.

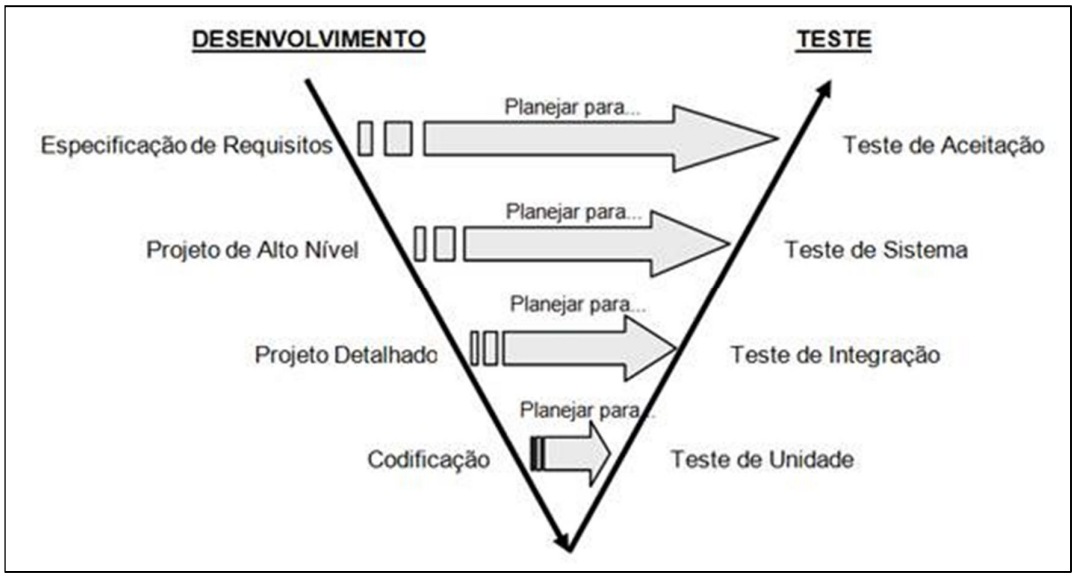
**c) Teste de Sistema:** avalia o *software* em busca de falhas por meio da utilização do mesmo, como se fosse um usuário final. Dessa maneira, os testes são executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia de manipulação do software. Verifica se o produto satisfaz seus requisitos.

d) **Teste de Aceitação:** são realizados geralmente por um restrito grupo de usuários finais do sistema. Esses simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado.
e) **Teste de Regressão:** Teste de regressão não corresponde a um nível de teste, mas é uma estratégia importante para redução de "efeitos colaterais". Consiste em se aplicar, a cada nova versão do software ou a cada ciclo, todos os testes que já foram aplicados nas versões ou ciclos de teste anteriores do sistema. Pode ser aplicado em qualquer nível de teste.

Dessa forma, em conformidade com a Figura 4, o planejamento e projeto dos testes devem ocorrer de cima para baixo, ou seja:

1. Inicialmente é planejado o teste de aceitação a partir do documento de requisitos;
2. após isso é planejado o teste de sistema a partir do projeto de alto nível do *software*;
3. em seguida ocorre o planejamento dos testes de integração a partir o projeto detalhado;
4. e por fim, o planejamento dos testes a partir da codificação.

Figura 4 - Modelo V descrevendo o paralelismo entre as atividades de desenvolvimento e teste de software.




Fonte: (Craig e Jaskiel, 2002).

## 6.2 TÉCNICAS DE TESTE DE SOFTWARE

Atualmente existem muitas maneiras de se testar um *software*. Mesmo assim, existem as técnicas que sempre foram muito utilizadas em sistemas desenvolvidos sobre linguagens estruturadas que ainda hoje tem grande valia para os sistemas orientados a objeto. Apesar dos paradigmas de desenvolvimento serem diferentes, o objetivo principal destas técnicas continua a ser o mesmo: encontrar falhas no software.

De acordo com Claudio ([201-]), as técnicas de teste são classificadas de acordo com a origem das informações utilizadas para estabelecer os requisitos de teste. Elas contemplam diferentes perspectivas do *software*, e impõe-se a necessidade de se estabelecer uma estratégia de teste que contemple as vantagens e os aspectos complementares dessas técnicas. As técnicas existentes são: técnica funcional e estrutural.

A técnica funcional, ou teste de caixa preta, trata o *software* como se fosse uma caixa com conteúdo desconhecido, sendo possível apenas a visualização do lado externo, ou seja, só é possível visualizar as entradas e as saídas que foram produzidas. Nesta técnica não há preocupação com os detalhes da implementação do projeto (MALDONADO; BARBOSA; VINCENZI, et al., 2004).

O teste funcional é constituído por dois passos principais, sendo o primeiro identificar as funções que software deve realizar e criar casos de teste capazes de fazer a verificação dessas funções. As funções de cada software são distintas e identificadas em suas especificais, dessa forma, para realizar esse tipo de teste é necessário que haja uma especificação bem elaborada e de acordo com os requisitos passados pelo usuário (MALDONADO; BARBOSA; VINCENZI; et al., 2004).

A técnica estrutural, também conhecida como teste de caixa branca, é um complemento à técnica de teste funcional, neste caso, os aspectos da implementação do projeto fazem parte dos testes. O teste estrutural se na baseia na própria estrutura do sistema para isso é feita uma representação do programa conhecida como grafo de fluxo de controle ou grafo de programa (MALDONADO, BARBOSA; VINCENZI; et al., 2004).

A técnica estrutural com a ajuda de seus grafos pode então realizar os testes do fluxo de controle do código de um programa, bem como o fluxo de dados e assim encontrar possíveis erros que serão gerados por falhas na estrutura do programa (MALDONADO; BARBOSA; VINCENZI, et al., 2004).

Outras técnicas de teste que podem e devem ser utilizadas de acordo com necessidades de negócio ou restrições tecnológicas são: teste de desempenho, teste de usabilidade, teste de carga, teste de stress, teste de confiabilidade e teste de recuperação.

Segundo Claudio ([201-]), alguns autores chegam a definir uma técnica de teste caixa cinza, que seria um mesclado do uso das técnicas de caixa preta e caixa branca, mas, como toda execução de trabalho relacionado à atividade de teste utilizará simultaneamente mais de uma técnica de teste, é recomendável que se fixe os conceitos primários de cada técnica.

# 7 BANCO DE DADOS

Um banco de dados se resume a um conjunto de informações armazenadas em determinado local. Uma lista telefônica ou armário de arquivos pode ser considerado um banco de dados, porém, esses bancos de dados manuais apresentam alguns problemas como, por exemplo, a dificuldade de busca, dificuldade de atualização dos dados, demora na busca para encontrar a informação e etc. (BEAULIEU et al., 2010).

Com base nos problemas citados anteriormente, os bancos de dados computacionais foram desenvolvidos para servirem como mecanismos de armazenamento e recuperação de dados computacionais. O armazenamento eletrônico de dados permite indexar informações de diversas formas, recupera-las e modificá-las com muito mais facilidade (BEAULIEU ET AL., 2010).

Ainda segundo o autor citado anteriormente, o modelo de banco de dados mais utilizado é o modelo relacional, o qual armazena as informações em tabelas, que são interligadas (BEAULIEU et al., 2010).

Os bancos de dados relacionais são manipulados com a utilização de uma linguagem chamada SQL, específica para recuperação e manipulação das informações contidas em bancos de dados eletrônicos. Existem vários tipos de sistemas de gerenciamento de banco de dados (SGBDs), estes por sua vez, são programas usam comandos SQL, os quais são utilizados para consultar, manipular e recuperar dados específicos de forma simplificada em um banco de dados (BEAULIEU et al., 2010).

## 7.1 FIREBIRD

O FireBird é um Sistema de Gestão de Banco de Dados (SGDB), multi-plataforma, *Open Source*, muito robusto, que conta com a grande maioria dos recursos mais utilizados, apresentando ótimo desempenho e sendo gratuito.

O FireBird surgiu em meados do ano 2000, quando a Borland também, também detentora e produtora da linguagem de programação Delphi, liberou o código fonte do InterBase 6.0 - o SGDB que era distribuído juntamente com o Delphi.

Com o lançamento do InterBase 6.0 Open, a Borland Software Corporation também se pronunciou informando ao publico que não daria continuidade ao projeto, diante disso algumas das principais figuras por traz do desenvolvimento do InterBase uniram-se aos atuais usuários do SGDB para dar continuidade a evolução do software, só que com o nome de FireBird. Das pessoas que estavam envolvidas nesse projeto, podemos citar Jim Starkey, Fundador do Interbase, Ann Harrison Ex-Presidente da InterBase Corp., Paul Beach, Ex-Gerente de vendas da InterBase Corp.

# 8 DELPHI

Em 1970, herdando diversas características do ALGOL, é lançada a linguagem de programação procedural Pascal. Na década de 80 a empresa Borland lança o Turbo Pascal que continua a se desenvolver, surgindo assim na década de 90 o Object Pascal, que já possuía as características da Orientação a Objeto (DALEPIANE, 2014).

O arquiteto do projeto da linguagem de programação Delphi foi Anders Heilsberg (também criador da linguagem C#), um grande pioneiro da área de tecnologias de desenvolvimento de software, como conexões com banco de dados, programação Orientada a Objeto e os ambientes de desenvolvimento rápido (RAD - *Rapid Application Development*) (DALEPIANE, 2014).

Em 1995, a Borland coloca no mercado uma ferramenta de criação em ambiente visual para criar aplicações Windows utilizado à linguagem Object Pascal, essa versão foi batizada de Delphi (DALEPIANE, 2014).

De acordo com Dalepiane (2014), em busca de inovações e de espaço no mercado de *softwares* para desenvolvimento de dados, a Borland continua a melhorar o Delphi. Na versão 2 do software surge o desenvolvimento, para 32 bits, mantendo a compatibilidade com a versão 16 bits. Na versão 3 a novidade foi o suporte a desenvolvimento de aplicações multicamadas em um ambiente RAD.

O Delphi continua a avançar e implementar novidades como a manipulação e criação de XML, suporte total a *WebService*, *DataSnap*, integração com o .NET e Linux (DALEPIANE, 2014).

Em 2009 foi lançada a primeira versão do Delphi sob o controle da Embarcadero, empresa que tem até hoje os direitos pelo Delphi e suas outras ferramentas. A Embarcadero por sua vez busca melhorar todas as ferramentas já presentes no Delphi, a fim de firmar o software como uma ferramenta consistente já que o Delphi havia perdido publico com o passar do tempo (RIZZATO, 2014).

De 2009 até 2016 a Embarcadero também criou um novo *framework* chamada FireMonkey que tem suporte a Windows, MAC, OS X, iOS, Android, Windows Phone e outros sistemas operacionais, isso permite que o desenvolvedor, crie uma versão para seu *software* utilizando um único código, independente do sistema em que será utilizado, faz se necessário apenas a modificação do Layout do

*software* devido a grande quantidade de tamanho de telas existentes entre notebooks, celulares e *tablets* (DALEPIANE, 2014).

Como ferramentas de grande sucesso que foram desenvolvidas em Delphi, podemos citar o Skype, Audio Grabber, Kindle Writer, WinRAR, entre outros.

# 9 INTELIGÊNCIA ARTIFICIAL

A definição de inteligência artificial não é tão simples quanto parece. Artificial é algo que pode ser definido com facilidade. Segundo Rosa et al. (2011) artificial é tudo aquilo que é feito pelo homem, a grande questão vem quando tentamos definir a inteligência.

Segundo Fernandes (2008), a inteligência sempre foi alvo de estudo durante nossa história. O Estudo da inteligência começa no campo da Filosofia que busca entender o pensamento do homem, porém a inteligência passou a ser estudada de forma científica por várias outras áreas do conhecimento humano, entre elas podemos citar, a psicologia, neurologia, engenharia, a própria computação, entre outros.

A inteligência é aquilo que permite ao ser humano tomar decisões, realizar escolhas e executar de forma eficiente diversas tarefas (FERNANDES, 2008).

A Inteligência Artificial (IA), então pode ser definida como a ciência que busca reproduzir o comportamento humano, em seus mais diversos tipos de sistemas, como pode ser visto na Figura 5, e isso por sua vez pode-se resumir em tarefas simples do dia-a-dia como, por exemplo, reconhecer as expressões faciais de uma pessoa, entender a linguagem natural (humana), gerar linguagem natural, imitar a visão humana, a fala e varias outras tarefas que podem ser executas até mesmo por uma criança de três anos com relativa facilidade, mas que se tornam extremamente complicadas de serem reproduzidas por um computador (ROSA, 2011).

Figura 5 - Tipos de sistemas de IA

| SISTEMAS QUE PENSAM COMO HUMANOS | SISTEMAS QUE PENSAM RACIONALMENTE |
|---|---|
| O novo e excitante esforço para fazer computadores pensarem... máquinas com mentes, no sentido literal e completo. (HAUGELAND, 1985).<br><br>A automação de atividades que nós associamos com o pensamento humano, atividades como tomada de decisões, solução de problemas, aprendizado. (BELLMAN, 1978). | O estudo de faculdades mentais através do uso de modelos computacionais. (CHARNIAK; MCDERMOTT, 1985).<br><br>O estudo de computação que tornem possível perceber, racionar e agir. (WINSTON, 1992). |
| **SISTEMAS QUE AGEM COMO HUMANOS** | **SISTEMAS QUE AGEM RACIONALMENTE** |
| A arte de criar máquinas que realizam funções que requerem inteligência quando realizadas por pessoas. (KURZWEIL, 1990).<br><br>O estudo de como fazer os computadores realizarem tarefas as quais, até o momento, as pessoas fazem melhor. (RICH; KNIGHT, 1994). | Um campo de estudo que busca explicar e emular comportamento inteligente em termos de processos computacionais. (SCHALKOFF, 1990).<br><br>O ramo da ciência da computação que se preocupa com a automação do comportamento inteligente. (LUGER; STUBBEFIELD, 1993). |

Fonte: Rosa (2011, p. 3).
Nota: Adaptada pelo autor.

Segundo Rosa (2011), a IA é uma área em constante expansão, e com a grande quantidade de estudos pelo futuro promissor que a tecnologia pode ter com os avanços. Em suas principais aplicações no meio cientifico podemos citar sua utilização em jogos como xadrez, em descoberta de falhas em projetos de engenharia, diagnósticos médicos, análises e projeções financeiras, e também na criação de sistemas cognitivos, permitindo assim, a coleta de informações a partir dos mais diversos meios de expressão humana, como a fala, a escrita e os sentimentos, e a partir dessas informações ser capaz de gerar conhecimento.

Na área da IA ao que se refere ao meio cognitivo podendo destacar o processamento de linguagem natural (PLN), esse assunto será tratado com mais detalhes posteriormente, mas é através dele que se torna possível a comunicação com a máquina utilizando a linguagem natural (humana) (SILVA, 2015).

## 9.1 PROCESSAMENTO DE LINGUAGEM NATURAL (PLN)

O Processamento de Linguagem Natural é uma subárea da IA que busca fazer com que as máquinas compreendam a linguagem natural dos seres humanos, ou seja, conseguir interpretar a escrita e a fala humana e, a partir disso, gerar conhecimento (ROSA, 2011).

O teste de Alan Turing (1950) foi criado com o intuito de definir se uma máquina é ou não inteligente através de um teste operacional. O teste baseia-se na premissa de que a maquina deve enganar um ser humano com louvor, para isso o computador será interrogado pelo usuário de um terminal, e partir disso ele deverá indicar se está conversando com o um humano ou com um computador. Se o computador conseguir convencer o usuário de que é outro ser humano, então, ele é considerado inteligente (ROSA, 2011).

Em conformidade com o teste de Alan Turing o computador precisaria possuir as seguintes capacidades (RUSSEL; NORVIG, 1995 citado por ROSA, 2011):

a) **Processamento de linguagem natural (PLN):** para que pudesse comunicar-se com sucesso em inglês ou qualquer outra língua humana.
b) **Representação de conhecimento:** para armazenar informações que fossem importantes antes e durante o interrogatório.
c) **Raciocínio automático:** para utilizar as informações que havia armazenado e com isso responder a questões e formular novas conclusões.
d) **Aprendizado de máquina:** para adaptar-se a novas situações e ir além dos padrões.

### 9.1.1 História

As primeiras pesquisas na área do PLN surgiram no inicio da década de 50, com um projeto de tradução automática, ideia de Warren Weaver, vice-presidente da fundação Rockfeller e grande conhecedor de criptografia computacional. Durante os dois primeiros anos várias universidades envolveram-se nesse projeto, dentre elas podemos citar a Universidade da Califórnia, Universidade de Harvard e Universidade de Georgetown e também o Instituto de Tecnologia de Massachusetts (MIT), e os

principais temas debatidos por eles eram a análise morfológica e sintática, necessidade de pré e pós-edição, a homografia (palavras de mesma escrita com sentidos diferentes dependendo da forma como é colocada em uma oração) entre outros (DIAS DA SILVA et al., 2007).

De acordo com Dias da Silva et al. (2007), os primeiros sistemas de tradução listavam, pura e simplesmente, várias possibilidades de tradução literal, não se preocupando com a análise sintática da oração, isso exigia que todos os textos fossem revisados por tradutores humanos para que apresentassem um resultado positivo. Com as criticas realizadas na época as pesquisas em PLN acabam se enfraquecendo e levando ao fim de vários projetos.

Por volta dos anos 70, as pesquisas em PLN foram retomadas de forma mais cautelosa e realista diante de toda a complexidade apresentada por cada linguagem segundo Dias da Silva et al. (2007). Vários outros projetos acadêmicos e comerciais não apenas voltados à tradução automática começaram a ganhar força.

Em 1970, uma tese de doutorado do MIT foi defendida por Winograd, este estudioso citou em sua tese o sistema SHRDLU, também conhecido como “mundo dos blocos”, que consistia em um sistema gráfico com blocos e um braço mecânico, de forma que o braço reproduzia o que era digitado, em inglês, no teclado, processando palavras para gerar as ações da maquina, e com isso Winograd conseguiu mostrar à comunidade que as máquinas podem ser programadas para interagir com seus usuários por meio da linguagem natural, a partir desse momento o PLN representou um objeto de grande valor a ser pesquisado, gerando várias possibilidades de pesquisas acadêmicas e comerciais sobre o assunto (DIAS DA SILVA et al., 2007).

### 9.1.2 Aplicações

Podemos citar algumas áreas de atuação dos sistemas de PLN, sendo elas sofisticadas ou mais simples, tais como:

a) **Manipulação de base de dados**: o sistema de manipulação de base de dados serve como uma forma de comunicação simples entre o usuário e a máquina, seu principal objetivo é transformar textos de linguagem natural em instruções para o banco de dados retornar os resultados esperados pelo usuário (DIAS DA SILVA et al., 2007).

b) **Sistemas tutores**: os sistemas tutores buscam simular diversas situações como: reproduzir o processo de aprendizagem de um aluno, métodos de ensino de conteúdos, métodos de analisar, corrigir e comentar erros, de avaliar o aprendizado, sugerir melhorias na forma de ensino buscando melhorar a interação com o aluno. Esses sistemas são um grande foco de pesquisas, visto que, utilizam muito o PLN e contém vários tópicos dentro de seu segmento o que gera uma grande riqueza de pesquisas (DIAS DA SILVA et al., 2007).
c) **Sistemas de automação de tarefas administrativas**: tem como objetivo auxiliar nas tarefas administrativas de uma empresa ou instituição, dentre as tarefas que eles realizam, podemos destacar a gerência de agendas de reuniões, selecionar e manipular, por meio de comandos orais, objetos no monitor de um computador, detectar erros ortográficos, gramaticais e frases que possam prejudicar a leitura do arquivo, entre outras funcionalidades (DIAS DA SILVA et al., 2007).
d) **Programação automática**: são sistemas bastante complexos que tem como principal objetivo receber e organizar as informações passadas por um programador por meio de uma entrevista, e a partir disso gerar um programa aceitável na linguagem de programação escolhida, a estrutura de dados mais adequada a situação, e corrigir erros caso seja necessário (DIAS DA SILVA et al., 2007).
e) **Sistemas específicos**: os sistemas específicos tem o objetivo de apresentar o conhecimento necessário para resolver uma situação, podemos exemplificar um sistema especifico como um sistema que recebe os sintomas de um paciente, e retorna o diagnostico, seu tratamento, os cuidados necessários e etc. de acordo com as pesquisas realizadas na sua base de dados (DIAS DA SILVA et al., 2007).
f) **Tradução automática**: os sistemas de tradução automática, como citado anteriormente, foram os pioneiros no ramo do PLN, e tem como objetivo traduzir textos de uma linguagem para outra, sem a necessidade de um tradutor humano para auxiliá-lo (DIAS DA SILVA et al., 2007).

g) **Sistemas de analise de sentimentos**: Os sistemas de análise de sentimentos ou mineração de textos, também foco deste trabalho, têm como principal objetivo analisar os textos escritos em linguagem natural, e a partir desse realizar um processamento para mostrar se aquela opinião sobre determinado assunto é positiva ou negativa. Esse processo aplicado a grandes massas de dados gera a informação geral da opinião de um determinado publico sobre um assunto, objeto, empresa entre outros.

### 9.1.3 Processo de Análise

Devido à grande quantidade de informações geradas, uma das principais aplicações do PLN é a extração de dados contidos em textos. Com isso surgem às dificuldades de colocar esse processo em prática, e o maior dos casos é ambiguidade. Existem dois tipos de ambiguidade: a léxica e a sintática (SILVA, 2014).

A ambiguidade léxica acontece quando a categoria e o sentido de uma palavra são ambíguos, por exemplo, “A rede foi cortada ontem à noite”, se não houver um contexto evolvido não é possível saber que tipo de rede é essa, pode ser uma rede elétrica, uma rede de computadores ou ainda uma rede de deitar (SILVA, 2014).

A ambiguidade sintática é aquela que pode conter uma sentença ambígua, por exemplo, “Eu li a notícia sobre a greve na faculdade”, no caso dessa sentença pode-se interpretar que a notícia foi lida na faculdade ou que a greve acontece na faculdade (SILVA, 2014).

Para a composição de um programa analisador de linguagem natural é necessário que seja implementado quatro módulos distintos, de forma que cada um deles realiza um tipo de análise, sendo elas, Análise Léxico-Morfológica, Sintática, Semântica e Pragmática, que serão abordadas a seguir (SILVA, 2014).

#### *9.1.3.1 Análise Léxico-Morfológica*

Para a análise léxico-morfológica a sentença é divida em itens lexicais (palavras) pelo analisador. O analisador realiza uma varredura, item a item,

decompondo-os em sues morfemas e faz a verificação das informações e características associadas a cada item lexical, ou seja, esta etapa faz a avaliação lexical e morfológica da sentença (SILVA, 2014).

O analisador léxico pode ser definido como “[...] uma lista de palavras e suas respectivas categorias gramaticais, classes sintáticas e informações semânticas.” (SILVA, 2014, p. 29), desta forma é associado a cada palavra as informações mais relevantes sobre ela.

No caso do analisador morfológico as palavras serão classificadas de acordo com sua categoria gramatical e para fazer a separação das palavras de uma sentença, o analisador utiliza delimitadores, como pontos e espaços (SILVA, 2014).

A grande dificuldade desta etapa fica por conta da ambiguidade, como já foi citado anteriormente, uma palavra pode conter mais de um sentido, assim como uma sentença também pode ser ambígua (SILVA, 2014).

#### *9.1.3.2 Análise Sintática*

A análise sintática consiste em analisar uma sentença item a item para verificar se se trata realmente de uma sentença válida, ou seja, ela relaciona a forma como as palavras são ligadas entre si, para verificar se não é apenas um conjunto de palavras em sequencia (SILVA, 2014).

Para a realização dessa análise, também conhecida como *Parser*, é necessário conhecer a estrutura que define uma linguagem, a partir disso as sequencias de palavras são quebradas em estruturas que estabelecem o relacionamento das palavras entre si (SILVA, 2014).

#### *9.1.3.3 Análise Semântica*

O analisador semântico, a partir da estrutura gerada pelo analisador sintático, vai designar um sentido a cada item. Com os objetos do domínio de atividade é feito um mapeamento na estrutura sintática, rejeitando as estruturas em que não é possível realizar o mapeamento (SILVA, 2014).

#### *9.1.3.4 Análise Pragmática*

A análise pragmática é a última e mais complicada etapa. O objetivo do analisador pragmático é responsável por verificar o sentido da frase e a parte gramatical e também se a frase possui nexo. Frases como “O número de ursos panda no Brasil esta diminuindo” pode atender os requisitos de todas as etapas anteriores, porém a frase não é coesa, já que no Brasil não existem ursos pandas (SILVA, 2014).

De acordo com Romeiro (2009, citado por SILVA, 2014), o sentido da frase, normalmente, é obtido nas etapas anteriores, deixando por conta do analisador pragmático conseguir o real sentido da mensagem.

O significado da frase é obtido na análise semântica, a partir da análise de suas partes, mas para o entendimento de um texto é preciso analisar o contexto e interpretá-lo, frases que expressam ironia tem significado diferente do literal e essa analise complexa é feita pelo analisador pragmático (SILVA, 2014).

## 9.2 MINERAÇÃO DE DADOS

A descoberta de conhecimento em volumes gigantescos de dados é conhecida como mineração de dados. O principal objetivo desse processo é encontrar padrões, fatos, associações não percebidas e, com isso, gerar conhecimento para auxiliar em tomadas de decisões. A realização desse tipo de processo por seres humanos seria extremamente custoso e demorado, para tanto surgiu a área da IA conhecida como *Data Mining* ou Mineração de Dados (CARRILHO JUNIOR et al., 2007).

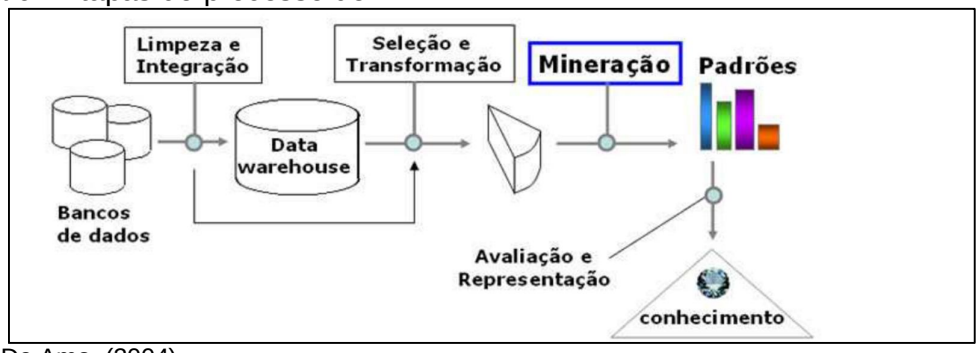
Ainda em conformidade com o autor citado, a Mineração de Dados é um processo composto por várias etapas, e geralmente realizado com dados estruturados, como por exemplo, o banco de dados de uma organização, utilizando informações como, por exemplo, pedidos realizados, compras de fornecedores, vendas e etc.

### 9.2.1 KDD

O termo KDD vem do inglês, *Knowledge Discovery in Database*, ou seja, descoberta de conhecimento em banco de dados, este é um amplo processo, no qual a mineração de dados está inserida. De acordo com De Amo (2004), e conforme ilustra a Figura 6, o KDD consiste nas seguintes etapas:

a) **Limpeza de dados:** nesta etapa os dados inconsistentes são retirados, assim como ruídos.
b) **Integração de dados:** junção de diferentes fontes de dados para criação de um repositório único.
c) **Seleção:** nesta etapa são definidos os atributos que interessam ao usuário, por exemplo, o usuário pode definir salario e numero de dependentes como informações relevantes para indicar se um cliente é bom pagador ou não, e descartar informações como o telefone ou número do documento, pois estas não serão relevantes.
d) **Transformação dos dados:** os dados são convertidos para um formato apropriado para que o algoritmo possa executar a mineração.
e) **Mineração:** nesta etapa são aplicadas técnicas para a extração os padrões de interesse do usuário.
f) **Avaliação ou Pós-processamento:** etapa onde os padrões de interesse do usuário são identificados de acordo com seus critérios.
g) **Visualização dos resultados:** etapa onde os resultados obtidos são mostrados ao usuário em alguma forma de representação, como gráfico ou relatório.

Figura 6 - Etapas do processo de KDD




Fonte: De Amo. (2004).

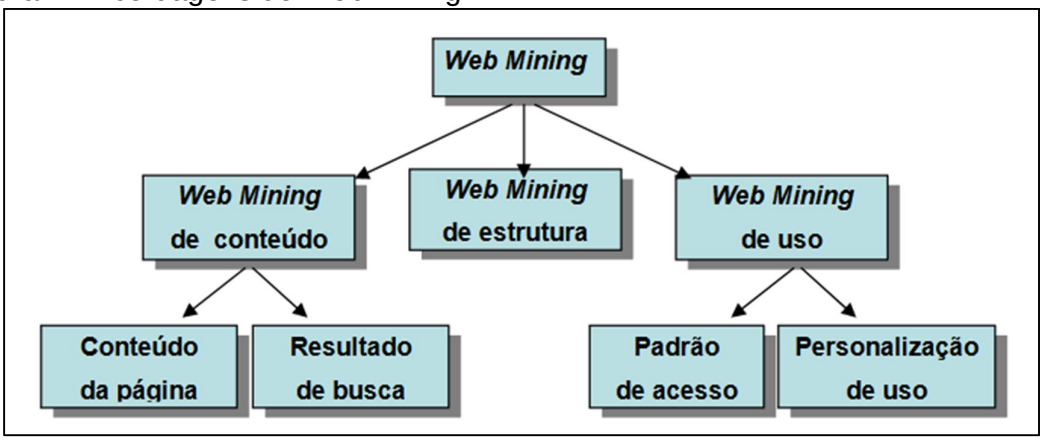
### 9.2.2 Web Mining

O *Web Mining* é uma variação de Data *Mining*, que tem como foco principal os dados que estão armazenados na *World Wide Web*, e que indiscutivelmente é uma grande fonte de informações dos mais variados tipos. Essas informações estão espalhadas em blogs, redes sociais, sites de pesquisas, sites de notícias, entre outros (CARRILHO JUNIOR et al., 2007).

O grande problema, é que cerca de 80% desses dados estão armazenados de forma não estruturada, ou seja, gráficos, imagens, vídeos, textos ou áudios. Por outro lado a internet também possui características únicas de estruturação que são muito úteis e interessantes para a extração de padrões, conhecimentos e até estudos de *interface*, a partir das tomadas de decisões dos usuários, dentre elas podemos citar os *hiperlinks* que formam a ligação entre as páginas e a estrutura de *tags*, bastantes utilizadas em documentos HTML (CARRILHO JUNIOR et al., 2007).

A *Web Mining* é constituída por três abordagens que tem como objetivo a descoberta de informação, sendo elas, *Web Mining* de conteúdo, *Web Mining* de estrutura e *Web Mining* de uso, conforme pode ser visualizado na Figura 7.

Figura 7 - Abordagens do Web Mining




Fonte: Carrilho Junior et al. (2007).

O *Web Mining* de conteúdo é utilizado para facilitar o acesso a conteúdo desestruturado, ou seja, analisar textos, imagens, vídeos e outros componentes existentes em páginas HTML. O *Web Mining* de estrutura estuda a forma como uma página está ligada a outra através de seus *hiperlinks*, dessa forma torna-se mais simples buscar um conteúdo específico da área de interesse. Já o *Web Mining* de

uso busca obter conhecimento através dos acessos, visitas e buscas dos usuários nos mais diversos *sites* da internet, com isso é possível reconhecer bons padrões para criação de *interfaces* para a internet, buscas, e também a personalização do uso da internet (CARRILHO JUNIOR et al., 2007).

Uma das principais áreas de aplicação do *Web Mining* é a análise de sentimentos ou Mineração de Opiniões, que conforme já citado, é o principal foco deste trabalho. Com a ajuda deste recurso é possível obter informação e conhecimento de maneira mais rápida sobre a opinião de pessoas sobre determinado assunto, produto ou serviço.

### 9.2.3 Mineração de Opinião

A mineração de opinião, mineração de sentimentos ou análise de sentimentos é a área da mineração de dados focada na extração de informação de textos específicos, no caso extração de informações de textos opinativos.

Pessoas costumam expressar sua opinião através de textos publicados nos mais diversos meios de comunicação da internet, e com o auxílio da mineração de opinião é possível extrair atributos e componentes que ajudam a identificar se um comentário sobre determinado assunto é positivo, negativo ou neutro.

Para a realização da maioria das compras é muito comum o usuário procurar a opinião de outras pessoas que já compraram aquele produto ou serviço (SANTOS, 2013).

Opinião e sentimento são conceitos distintos se o sentido literal da palavra for analisado, porém, de acordo com Santos (2013), quando se trata da mineração de opinião esses são conceitos equivalentes.

Com a grande quantidade de textos produzidos no ambiente da internet atualmente, o ramo da mineração de opinião vem crescendo, buscando ajudar pessoas e empresas, a saber, a qualidade de um determinado produto ou serviço de forma simples, o que contribui para que as empresas possam melhorar seus produtos e desta forma as pessoas possam comprar produtos mais confiáveis que atendam melhor suas necessidades.

### 9.2.4 Etapas da Coleta de Dados

A etapa da coleta dos dados tem como objetivo buscar opiniões em redes sociais utilizando mecanismos avançados de busca visando minimizar o máximo possível o esforço humano. Segundo Becker e Tumitan (2005), aqueles que são fatos puros e simplesmente devem ser descartados, porém, opiniões baseadas em fatos devem ser mantidas para análise.

#### *9.2.4.1 Crawler*

A utilização de *Crawlers* é um método de coleta muito eficiente, pois eles são robôs que percorrem páginas *web* de maneira específica, salvando URLs de páginas, arquivos, imagens e textos de acordo com a necessidade do usuário (SANTOS, 2010).

De acordo com Santos (2010), os *Crawlers* funcionam da seguinte forma: é fornecido um ou mais *sites* como base, por sua vez, esse *site* recebe o nome de semente ou seed, e a partir desse *sit*e é iniciada a navegação. O *Crawler* então começa a realizar uma "varredura" na página e armazena os dados que vai encontrando, inclusive URLs, formando uma lista de *sites* que ele pode visitar posteriormente para adquirir mais dados.

Ainda em conformidade com o autor supracitado, alguns tipos de *Crawler* são focados em operações específicas, como extração de imagens, arquivos, textos contidos em *tags* especificas ou com palavras específicas. Esses *Crawlers* recebem o nome de *Crawler* Focado.

#### *9.2.4.2 Coleta*

O processo de coleta consiste na busca e recuperação de dados com o objetivo de formar uma base de dados textual, da qual se pretende extrair o conhecimento (CARRILHO JUNIOR et al., 2007).

As três principais fontes para a coleta de dados são: pastas e arquivos, bancos de dados e a internet. A seguir será relatado cada um dos ambientes.

A busca de pastas e arquivos consiste em encontrar arquivos de texto no disco rígido de um computador. A grande dificuldade dessa busca está em identificar

os arquivos que são gerados por pessoas, e os arquivos binários e de configurações que são gerados pelo sistema operacional para guardar as preferencias do usuário, e as configurações do sistema (CARRILHO JUNIOR et al., 2007).

A obtenção de dados por meio de tabelas de um banco de dados ocorre com a extração de textos armazenados em campos do tipo string, que são campos de livre digitação onde a única restrição é a quantidade de caracteres suportados pelo campo (CARRILHO JUNIOR et al., 2007).

A terceira fonte de coleta, e também foco deste trabalho é a internet, que de acordo com Carrilho Junior et al. (2007), nesse ambiente as dificuldades são muitas devido a grande diversidade de tipo de páginas encontradas, como por exemplo, blogs, revistas, anúncios, redes sociais, entre outros, por isso é muito comum a utilização de ferramentas de busca avançadas como os *Crawlers*, já explicados anteriormente, para encontrar e armazenar esses dados de forma específica.

Os dados armazenados constituem uma coleção de comentários que são conhecidos na mineração de dados como corpus, de forma geral são comentários previamente coletados de alguma forma definida pelo usuário (CARRILHO JUNIOR et al., 2007).

#### *9.2.4.3 Pré-processamento*

De acordo com Carrilho Junior et al. (2007), a etapa de pré-processamento consiste em retirar dos dados coletados tudo que seja inútil, para assim melhorar o desempenho do software nas etapas seguintes. O processo consiste na retirada de Links, caracteres não alfabéticos e etc.

Ainda nesta etapa, conforme o autor citado existe a aplicação da remoção de stopwords, que segundo Dias da Silva (2007) são as palavras de classe fechada como pronomes, conjunções, artigos e preposições, e também palavras de significa desprezível para o contexto, podendo variar com o projeto, essas palavras aparecem com grande frequência e correspondem a artigos, preposições, pontuação, conjunções e pronomes. A remoção dessas palavras, como dito acima, contribui para a melhoria no desempenho do software, como exemplos dessas palavras pode-se citar: “A”, “O”, “pelo”, “por”, “em”, “!”, “?”, “são”, “seu”, “como”, “lá”, entre outras.

Após a retirada das stopwords é criado um vetor de palavras com as palavras que ainda restaram, essa representação é conhecida como bag of words, ou saco de palavras, visto que, o comentário é visto como um contêiner de tokens onde não há ligação nem uma entre eles (CARRILHO JUNIOR et al., 2007).

#### *9.2.4.4 Classificação*

Nessa etapa é feita a classificação dos dados coletados, determinando se uma opinião é positiva, negativa ou neutra. Durante esta etapa de grande importância dentro do processo de análise de sentimentos, as palavras mais importantes são aquelas que expressam a opinião do autor, como exemplo dessas palavras pode citar: “ótimo”, “bom”, “ruim”, “péssimo” e etc. (BECKER; TUMITAN, 2005 citado por SILVA, 2015).

#### *9.2.4.5 Apresentação dos resultados*

Após classificados os dados, a próxima etapa consiste na apresentação dos resultados obtidos ao usuário. Isso pode ser feito de forma gráfica ou em forma de texto, a apresentação de gráficos sempre é uma forma melhor de visualizar esse tipo de informação, pois facilita o entendimento do usuário (BECKER; TUMITAN, 2005, citado por SILVA, 2015).

## 9.3 NAIVE BAYES

A classificação de dados é uma técnica amplamente utilizada em mineração de dados, esse processe consiste em encontrar um modelo ou função que descreva diferentes classes de dados, dessa forma, novas instâncias devem ser classificadas automaticamente aplicando o modelo ou função aprendida (LUCCA; PEREIRA; PRISCO et al., 2013).

No processo de classificação de dados diversos algoritmos utilizam os princípios e teorias da estatística. Um dos métodos utilizados foi formulado por Thomas Bayes, ministro presbiteriano britânico que viveu no século XVIII, e formulou o Teorema de Bayes que é utilizado no Classificador Bayesiano Ingênuo, que está

relacionado ao cálculo de probabilidades condicionais (GOLDSCHMITD; BEZERRA; PASSOS, 2015)

O algoritmo de Naive Bayes é um algoritmo baseado em cálculo probabilístico. Esse algoritmo baseia-se na ocorrência de um fato X e um fato Y que depende diretamente das mais diversas formas de ocorrência de um fato X, dessa forma é calculado o número de casos de ocorrência de X e Y, dividido pela ocorrência apenas do X (FACELI; LORENA; GAMA et al., 2011), conforme ilustra a equação a seguir.

$$P\,(X|Y) = \frac{P(X1|Y)*P(X2|Y)\ldots P(Xn|Y)*P(Y)}{P(X1)\,*P(X2)\ldots P(Xn)} \quad (1)$$

A fórmula pode ser aplicada em ocasiões que existam várias classes, ou seja, o fato Y pode pertencer a mais de uma classe, nesse caso, o algoritmo de Bayes é aplicado a todas as diversas classes, e a cada uma delas é atribuído o valor da probabilidade do fato Y pertencer a classe. Com todas as probabilidades de cada classe atribuídas é selecionada a classe com a maior estimativa de probabilidade como saída do algoritmo (BAEZA-YATES; RIBEIRO-NETO et al., 2013).

Para garantir melhores resultados as probabilidades é realizado o processo de normalização. O processo de normalização consiste em dividir por um valor fixo a probabilidade a posteriori de um par de variáveis, para garantir que a soma de seus valores seja 1(GOLDSCHMITD; BEZERRA; PASSOS, 2015).

Para exemplificar a utilização do algoritmo bayesiano será utilizado um exercício bastante comum no aprendizado dessa técnica, que consiste em indicar se uma pessoa deve ou não jogar Tênis, em um dia com determinadas condições climáticas.

Considere o banco de dados mostrado na Figura 8 para execução do exemplo.

Figura 8 - Banco de dados exemplo

| Aparência | Temperatura | Umidade | Vento | Jogar Tênis |
|---|---|---|---|---|
| Ensolarado | Quente | Alta | Fraco | Não |
| Ensolarado | Quente | Alta | Forte | Não |
| Nublado | Quente | Alta | Fraco | Sim |
| Chuvoso | Moderado | Alta | Fraco | Sim |
| Chuvoso | Fresco | Normal | Fraco | Sim |
| Chuvoso | Fresco | Normal | Forte | Não |
| Nublado | Fresco | Normal | Forte | Sim |
| Ensolarado | Moderado | Alta | Fraco | Não |
| Ensolarado | Fresco | Normal | Fraco | Sim |
| Chuvoso | Moderado | Normal | Fraco | Sim |
| Ensolarado | Moderado | Normal | Forte | Sim |
| Nublado | Moderado | Alta | Forte | Sim |
| Nublado | Quente | Normal | Fraco | Sim |
| Chuvoso | Moderado | Alta | Forte | Não |

Fonte: Goldschmidt, Passos et al. (2005).

O atributo “Jogar Tênis” é o objetivo da classificação, então, podemos observar na Figura 8 que existem duas classes: Jogar = sim e Jogar = não.

Os atributos Xi são “Aparência”, “Temperatura”, “Umidade” e “Vento”. Para um dia ensolarado, quente, de alta umidade e vento fraco uma pessoa deve ou não jogar Tênis?

Com a aplicação do Classificador Bayesiano Ingênuo para a possibilidade Sim de jogar, tem-se:

1. P(Jogar=Sim| ensolarado, quente, alta umidade, vento fraco) = P(Jogar=Sim) * P(ensolarado|Jogar=Sim) * P(quente|Jogar=Sim) * P(alta umidade|Jogar=Sim) * P(vento fraco|Jogar=Sim) = ?.
2. P(Jogar=Sim| ensolarado, quente, alta umidade, vento fraco) = (9/14) * (2/9) * (2/9) * (3/9) * (6/9) = ?.
3. P(Jogar=Sim| ensolarado, quente, alta umidade, vento fraco) = 0,6429 * 0,2222 * 0,2222 * 0,3333 * 0,6667 = 0,0071.

Com a aplicação do Classificador Bayesiano Ingênuo para a possibilidade Não de jogar, tem-se:

1. P(Jogar=Não| ensolarado, quente, alta umidade, vento fraco) = P(Jogar=Não) * P(ensolarado|Jogar=Não) * P(quente|Jogar=Não) * P(alta umidade|Jogar=Não) * P(vento fraco|Jogar=Não) = ?.
2. P(Jogar=Não| ensolarado, quente, alta umidade, vento fraco) = (5/14) * (3/5) * (2/5) * (4/5) * (2/5) = ?
3. P(Jogar=Não| ensolarado, quente, alta umidade, vento fraco) = 0,3571 * 0,6000 * 0,4000 * 0,8000 * 0,4000 = 0,0274.

Aplicando a normalização tem-se:

**a)** P(Jogar=Sim| ensolarado, quente, alta umidade, vento fraco) = 0,0071/ (0,0071 + 0,0274) = 0,20.

**b)** P(Jogar=Não| ensolarado, quente, alta umidade, vento fraco) = 0,0274/ (0,0071 + 0,0274) = 0,80.

Com base nos resultados do cálculo podemos dizer que o dia não é apropriado para jogar Tênis, já que a porcentagem da Classe “Não” é maior que a da Classe “Sim”.

O algoritmo de Naive Bayes é bastante simples por ser fácil de ser implementado, e o processo de construção de modelos iniciais para testes é bastante eficiente (FACELI; LORENA; GAMA et al., 2011). Domingos e Pazzini (1997 citados por FACELI et al., 2011) destacam que o algoritmo apresenta um bom desempenho em uma grande variedade de domínios, enquanto Kononenko (1991 citado por FACELI et al., 2011) complementa que o algoritmo é robusto, pois tem bom desempenho diante da presença de classes irrelevantes e ruído.

O algoritmo de Naive Bayes será utilizado para realizar a classificação dos comentários neste trabalho, de forma que ele será responsável por calcular a probabilidade de um comentário pertencer à determinada classe.

# 10 TRABALHOS CORRELATOS

As áreas da IA, inclusive a de processamento de linguagem natural são áreas relativamente novas e em constante expansão, por isso estão em destaque no ramo da computação, o que justifica a grande quantidade de informação e pesquisas feitas sobre o assunto.

Toda a metodologia deste trabalho foi construída com base em experiências anteriores citadas por autores em seus trabalhos, a fim de não cometer os mesmos erros e salientar os pontos mais relevantes e críticos do projeto.

Como trabalhos correlatos podemos citar a pesquisa intitulada “Mineração de Textos: Análise de Sentimentos Utilizando Tweets Referentes à Copa do Mundo 2014”, de José Adail Carvalho Filho, que mostra a implementação de um sistema voltado à mineração de opiniões em um ambiente especifico, que tinha como principal objetivo mostrar a implementação desse tipo de sistema e os resultados obtidos.

Outra pesquisa de grande importância intitulada “Mineração de Opiniões Aplicada a Mídias Sociais para as Organizações”, da autora Jéssica Rodrigues da Silva, cujo objetivo era retratar o desenvolvimento de um sistema de análise de sentimentos, suas dificuldades e sua importância. Este trabalho acabou por ofertar uma visão geral da importância desta temática para o mercado, podendo proporcionar grandes ganhos e vantagens competitivas as empresas se que utilizam a mesma, bem como informações de implementações, dificuldades e também os resultados sobre a aplicação desse sistema.

Dessa forma, o principal objetivo deste trabalho é contribuir nas pesquisas relacionadas a área de PLN, fornecendo informações, análises e resultados sobre as relevâncias da área, visando auxiliar organizações prestadoras de serviços a agradar seus clientes e manter-se de forma competitiva no mercado.

## 11 METODOLOGIA

O trabalho em questão retrata o desenvolvimento de um software na área de análise de sentimentos, para isso, duas etapas principais foram adotadas, a primeira etapa consistiu no levantamento e escrita do referencial teórico, enquanto a segunda consistiu na modelagem e desenvolvimento do software se valendo de técnicas de engenharia de software para a construção de um software robusto.

O referencial teórico consistiu em uma investigação sobre conteúdos pertinentes aos assuntos e tecnologias que foram utilizadas. Dentre elas pode-se salientar a Inteligência Artificial e suas subáreas, como por exemplo, o processamento de linguagem natural (PLN), e a mineração de dados que são à base desse estudo.

Os softwares Delphi Seattle em conjunto com o gerenciador de banco de dados FireBird 2.5 foram estudados, com o propósito de se familiarizar com os mesmos, e ao mesmo tempo, investigar se atenderiam as necessidades.

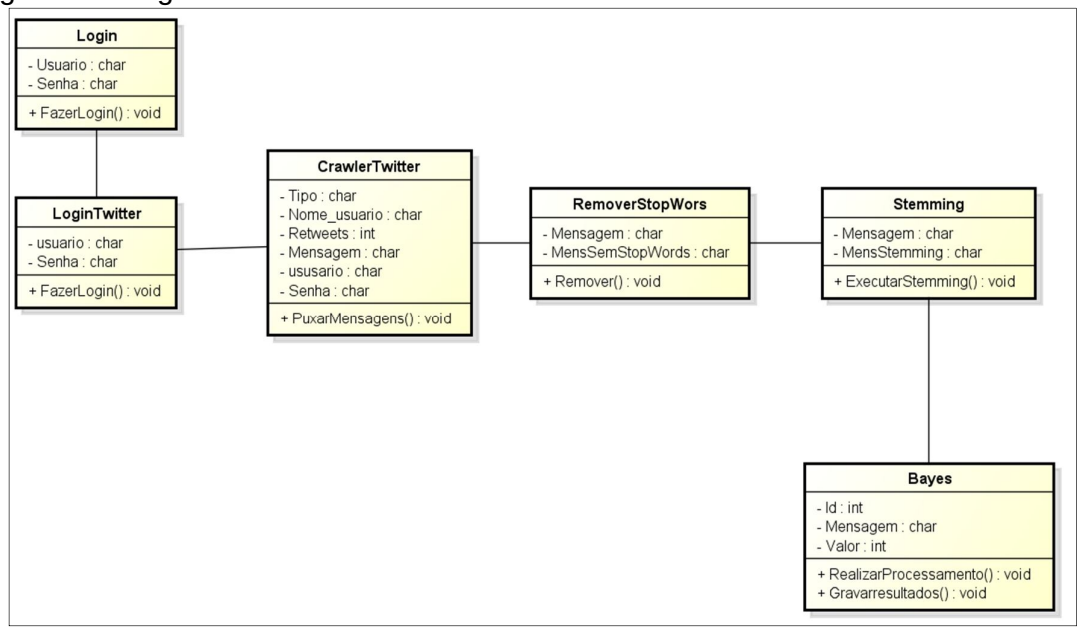
O Delphi Seattle foi escolhido por ser uma ferramenta de desenvolvimento robusta, que apresenta aspectos como a orientação a objeto, possibilidade de manter dados em memória enquanto eles são manipulados, o que acaba aumentando o desempenho da aplicação, além de ser um ambiente de desenvolvimento rápido, com a possibilidade de criação de protótipos utilizando massas de dados fictícias.

O sistema de gerenciamento de banco de dados FireBird 2.5 foi escolhido por também ser gratuito, de fácil instalação e configuração, e ainda apresentar ótimo desempenho para aplicações com uma massa intermediaria de dados.

Para as etapas seguintes foram feitas as modelagens do projeto utilizando a Linguagem de Modelagem Unificada (UML), linguagem esta utilizada para especificação, visualização, construção e documentação de artefatos de desenvolvimento de sistemas.

Dentre os diagramas, foi desenvolvido o Diagrama de Classes para fazer a representação das classes, interfaces e outros componentes do sistema, conforme ilustra a Figura 9.

Figura 9 - Diagrama de Classe do sistema



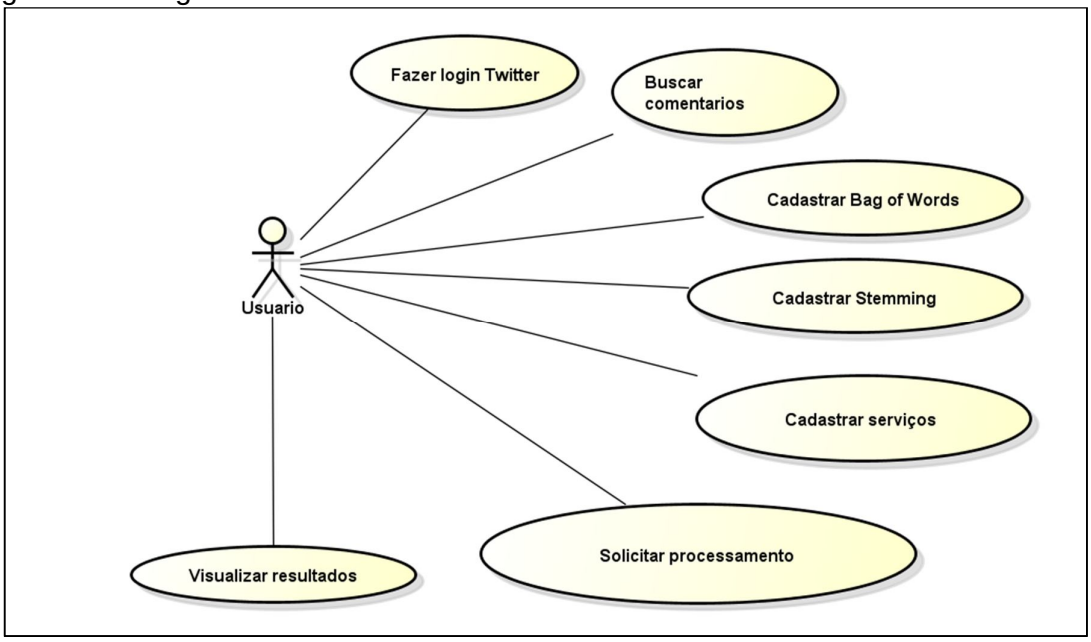

Fonte: Elaborada pelo autor.

No Diagrama de Classe foi criada a classe LoginTwitter que armazena as informações do login do usuário e realiza o processo de login, logo em seguida a classe CrawlerTwitter, puxa os Tweets de um determinado usuário e armazena o comentário, data, o usuário e a quantidade de Retweets, essa classe também é responsável por realizar o filtro dos comentários que não são relacionados a serviço.

A classe RemoverStopWords remove as palavras irrelevantes do comentário, gravando a Bag of Words, em seguida a classe Stemming realiza a aplicação do Stemming na Bag of Words melhorando o desempenho com a remoção de grau e gênero das palavras. Por fim, a classe Bayes realiza a classificação Bayseana e registra a polaridade encontrada para os comentários que estão sendo analisados.

Em seguida foi feito o Diagrama de Caso de Uso mostrando os atores envolvidos e seus relacionamentos, com o objetivo de visualizar de forma ampla a interação dos usuários com o sistema, conforme mostrado na Figura 10.

Figura 10 - Diagrama de Caso de Uso



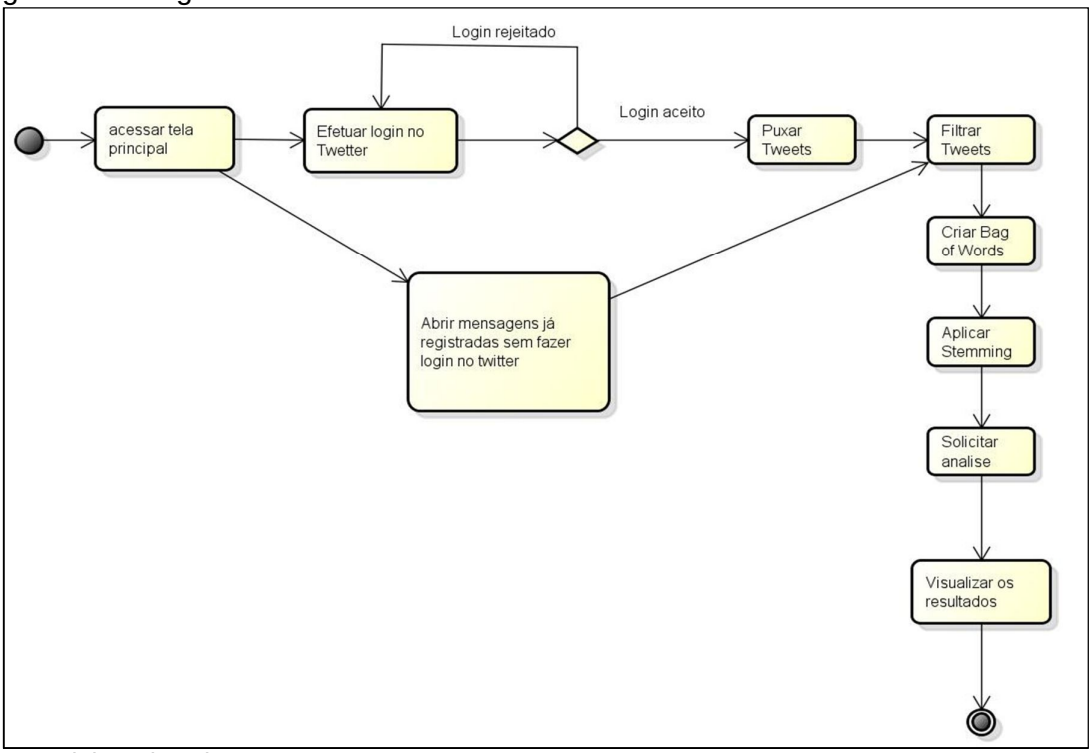

Fonte: Elaborada pelo autor.

Das ações disponíveis no Diagrama de Caso de uso é possível ver que o usuário pode fazer o login no Twitter com isso ele terá acesso a buscar novos comentários. Para a realização dos treinamentos é cadastrar a Bag of Words, Stemming e as palavras que remetem aos serviços prestados.

Com os treinamentos feitos, ao solicitar o processamento todos os cálculos serão feitos possibilitando ao usuário visualizar os resultados obtidos pelo sistema.

Por fim, foi feito um Diagrama de Atividades responsável por mostrar os fluxos de controle e fluxos de dados de cada etapa do software, conforme retrata a Figura 11.

Figura 11 - Diagrama de Atividades



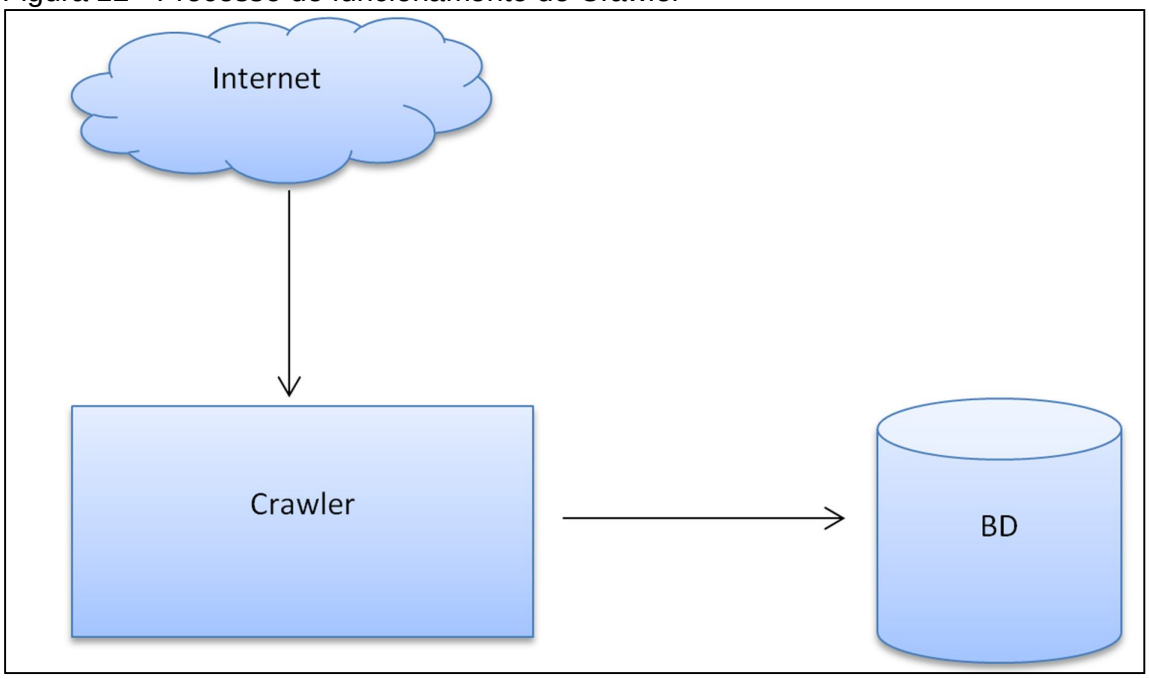

Fonte: Elaborada pelo autor.

No Diagrama de Atividades é possível ver a sequência de atividades realizadas pelo usuário, sendo assim, ao iniciar o sistema ele estará na tela inicial, onde pode fazer o login no Twitter para puxar novos comentários ou simplesmente acessar os comentários já registrados. A partir disso é possível filtrar os relatórios com base nos serviços cadastrados e posteriormente solicitar a criação da Bag of Words, aplicação do Stemming e análise utilizando o algoritmo Bayseano, ao fim dessa sequencia é possível ver os resultados obtidos.

Para construir esses diagramas foi utilizada a ferramenta Astah Community, pois é uma ferramenta simples de ser utilizada, que permite a construção de diagramas UML a partir de uma interface simples, além de ser uma ferramenta gratuita.

Dentre os módulos que foram modelados/desenvolvidos, foi criado um Crawler, com a finalidade de acessar a rede social Twitter, através da API, que é uma interface de comunicação disponibilizada por eles para ter acesso mais fácil aos Tweets feitos pelos usuários. O Crawler vai reunir as opiniões que serão retiradas de forma automática e fazer seu armazenamento em um banco de dados para posterior análise e processamento, de acordo com a Figura 12.

Figura 12 - Processo de funcionamento do Crawler




Fonte: Elaborada pelo autor.

As opiniões foram extraídas da rede social Twitter, rede social bastante utilizada nos tempos atuais, que conta com uma grande variedade de informações já definidas, e uma estruturação que permite agrupar comentários com marcações do nome da empresa, dessa forma a obtenção de opiniões sobre um usuário do Twitter é mais precisa.

Na sequência, foi desenvolvido o módulo responsável pela análise dessas opiniões capturadas. Inicialmente foi feito uma Bag of Words que consiste na extração das palavras mais importantes. Esse processo retira do texto palavras como “O”, “A”, “ELE”, “ELA”, e outras palavras que são necessárias para dar o sentido à frase, mas que não são necessárias para realizar o cálculo posteriormente, com esse processo o desempenho tende a melhorar, porém os resultados de melhoria não são garantidos.

Após a extração da Bag of Words, foram aplicadas técnicas de Stemming, essas técnicas consistem na eliminação de variações morfológicas de uma palavra, para isso são extraídos e utilizados apenas o radical da palavra, como exemplo vamos utilizar a palavra “Aluno” e suas variações. Neste caso as palavras “Aluno”, “Aluna”, “Alunos”, “Alunas” possuem o mesmo sentido no momento de realizar os cálculos, assim utilizamos apenas o radical Alun.

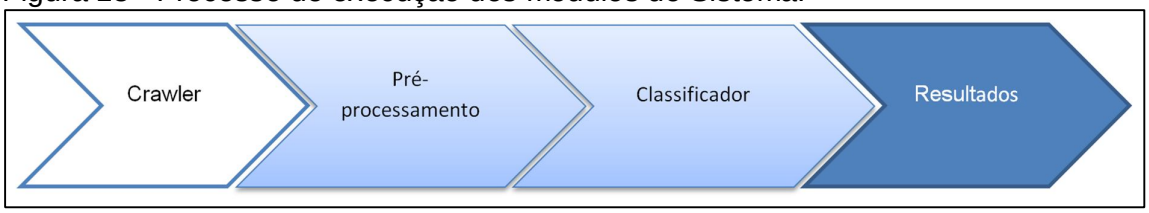
Após a aplicação do Stemming, com os dados já pré-formatados, estes foram submetidos ao algoritmo de Naive Bayes, que através de cálculos probabilísticos, com base na ocorrência de determinadas palavras no texto, que indicou a probabilidade de cada comentário pertencer à determinada classe, e assim possibilitando definir se um comentário é positivo, negativo ou neutro.

O algoritmo de Naive Bayes foi escolhido, pois é simples de ser implementado, apresenta bom desempenho em uma grande variedade de domínios, e também não é tão suscetível a ruído e classes irrelevantes. Todas essas características foram apresentadas no referencial teórico desta pesquisa, acompanhadas de seus respectivos autores.

Os testes do sistema foram realizados pelo próprio desenvolvedor, definindo uma quantidade de 50 comentários extraídos de redes sócias. Com base neles o sistema foi treinado, definindo as palavras que fariam parte da Bag of Words, aplicado o processo de stemming e definindo a polaridade de cada comentário de forma manual. Após o processo, doze comentários foram criados pelo autor para que todas as classes pudessem ser testadas, visto que, é difícil encontrar comentários positivos e neutros, então, os comentários foram submetidos ao processamento do software para averiguar os resultados.

Durante todo o processo além de acompanhar visualmente os resultados, também foi acompanhado todos os passos que estavam sendo feitos pelo programa em suas linhas de código, analisando assim, todos os valores que estavam sendo guardados em cada variável.

Figura 13 - Processo de execução dos módulos do Sistema.



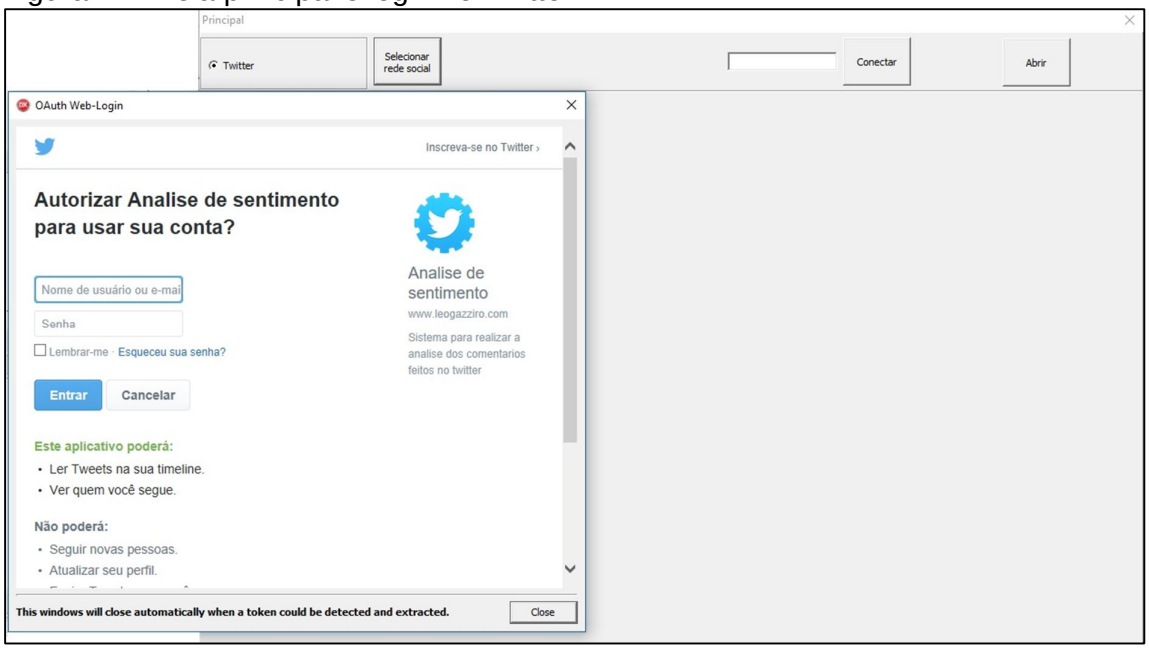

Fonte: Elaborada pelo autor.

Para a finalização, conforme o processo indicado na Figura 13, os resultados da análise foram apresentados de forma gráfica ao usuário, para que a partir dessas informações, decisões importantes pudessem ser tomadas.

## 11.1 DESENVOLVIMENTO DO SOFTWARE

Inicialmente ao abrir o sistema o usuário poderá ter acesso aos dados que já estão armazenados, ou poderá fazer o login na rede social Twitter para realizar suas pesquisas, conforme pode ser visualizado na Figura 14.

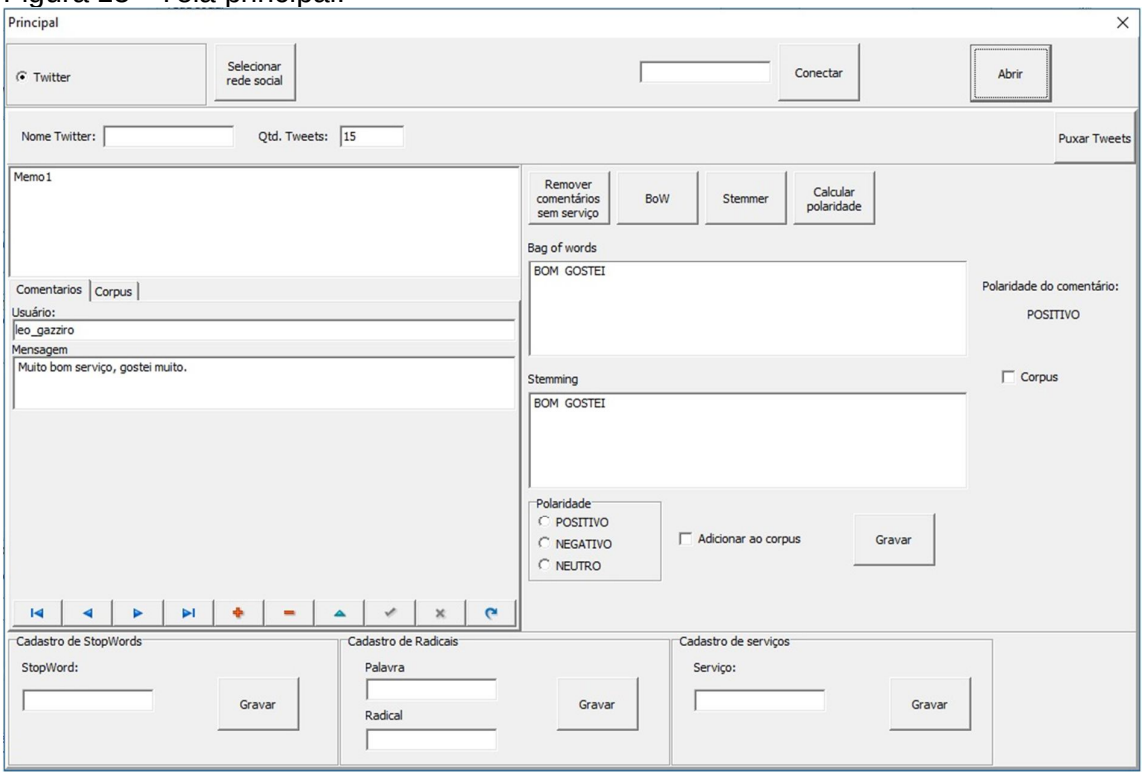
Figura 14 - Tela principal e login no Twitter.




Fonte: Elaborada pelo autor.

Com o login realizado o usuário tem acesso à tela principal do sistema, retratada na Figura 15, que busca os comentários feitos sobre determinada conta do Twitter.

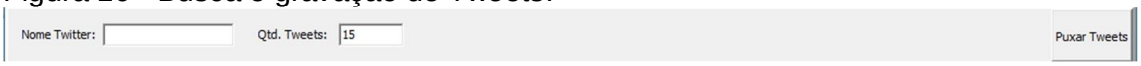
Figura 15 - Tela principal.




Fonte: Elaborada pelo autor.

É possível definir a quantidade de Tweets que serão buscados, sendo o máximo 100, restrição da própria API do Twitter, sendo eles os mais recentes, como pode ser visto na Figura 16. Com as mensagens gravadas é possível realizar a exclusão daquelas que são indesejadas, e que por sua vez, não fazem parte do grupo dos serviços definidos anteriormente.

Figura 16 - Busca e gravação de Tweets.




Fonte: Elaborada pelo autor.

É muito importante que as StopWords, ou seja, as palavras que são irrelevantes para a realização do cálculo de polaridade do comentário sejam cadastradas, mostrado na Figura 17, e que a Bag of Words seja construída de uma forma eficiente, tendo em vista que apenas as palavras realmente irrelevantes para a classificação devem ser cadastradas.

A função de cadastramento de Stemming realiza o cadastramento das palavras e seus respectivos radicais, como pode ser visto na figura 17, esse

processo também é importante para melhorar o desempenho do cálculo, dessa forma, como citado anteriormente, variações de gênero e grau das palavras são desconsideradas e os cálculos utilizam apenas os radicais das mesmas.

Figura 17 - Cadastros de StopWords, Stemming e palavras referentes ao serviço da empresa.


Fonte: Elaborada pelo autor.

Após definir os comentários que serão utilizados, criar o Bag of Words e aplicar o Stemming é muito importante que uma quantidade, que o usuário julgue razoável seja definida como Corpus, retratado na Figura 18. O Corpus são comentários que serão utilizados pelo sistema para a realização do cálculo probabilístico de Naive Bayes. São comentários definidos de forma manual dentre as três classes existentes, positivo, negativo e neutro, de forma que o autor puxou comentários reais da rede social Twitter e realizou a leitura de todos os eles, classificando-os de acordo com a polaridade apresentada.

Figura 18 - Busca e gravação de Tweets.


Fonte: Elaborada pelo autor.

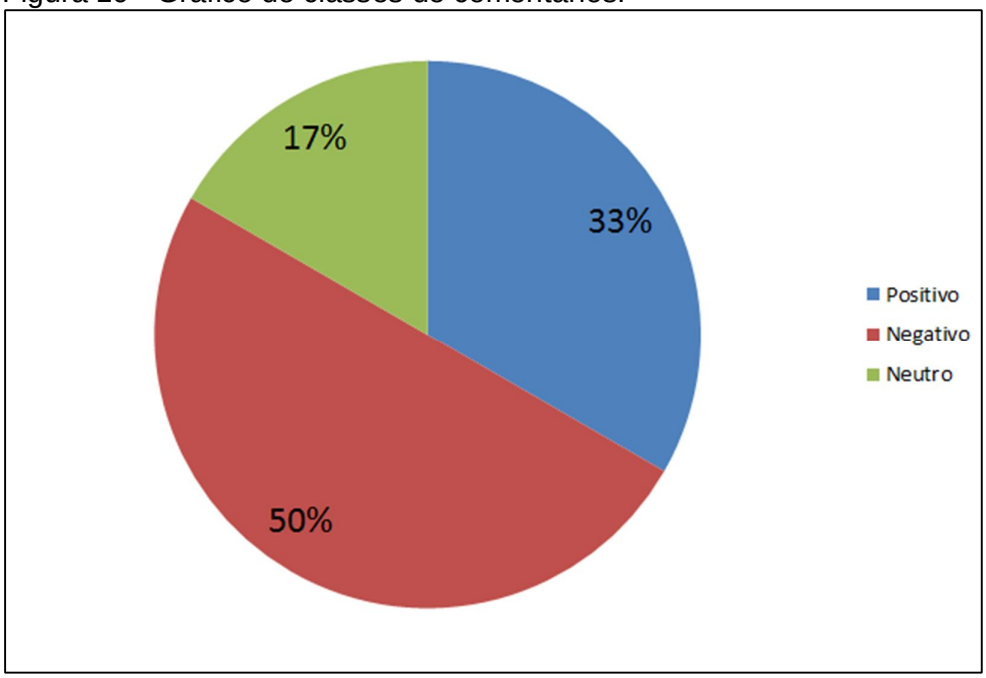
Com os pré-requisitos citados anteriormente então definidos, o sistema pode ser utilizado, dessa forma ele indicará, utilizando o algoritmo de probabilidade Bayseano a qual classe um comentário pertence, para isso é realizado o calculo palavra a palavra, não considerando expressões. Para que a precisão do sistema seja melhorada, após realizar a conferência dos resultados apresentados em cada comentário, é possível fazer a alteração da polaridade comentário, e posteriormente inclui-lo no Corpus, melhorando a classificação dos próximos comentários.

## 12 RESULTADOS

Para avaliar a eficácia do sistema proposto, ele foi submetido a testes, para isso foi utilizado 50 comentários extraídos da rede social Twitter, com esses comentários foram cadastradas as StopWords para criar a Bag of Words, também foram cadastradas as palavras e seus respectivos radicais para a aplicação do Stemming, a partir disso cada comentário foi classificado manualmente lendo cada um deles e indicando a qual classe o comentário pertence e o mesmo foi adicionado ao Corpus.

Com o sistema treinado, foram criados 12 comentários pelo próprio autor, sendo eles 4 positivos, 6 negativos e 2 neutros, como pode ser visto na Figura 19, utilizados para medir a precisão dos cálculos. Essa medição foi feita considerando o número de acertos do sistema diante dos cálculos realizados.

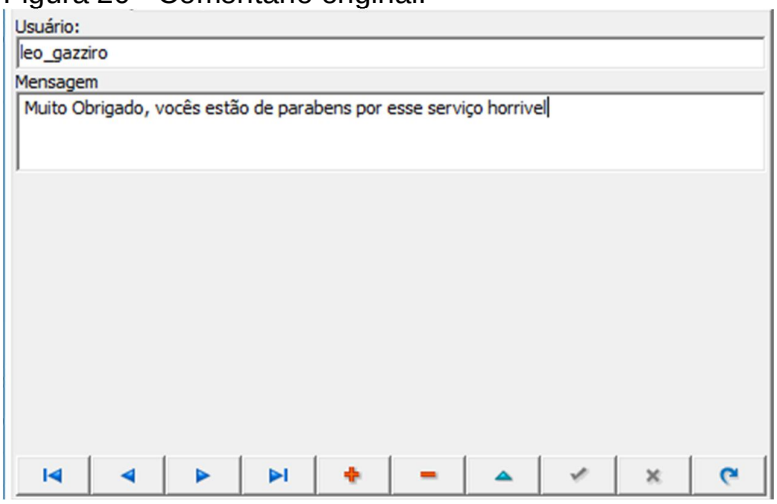
Figura 19 - Gráfico de classes de comentários.




Fonte: Elaborada pelo autor.

O comentário que pode ser visto na Figura 20, foi um dos 12 comentários registrados de forma manual pelo próprio autor.

Figura 20 - Comentário original.




Fonte: Elaborada pelo autor.

Como o sistema já estava treinado, o comentário foi submetido ao processo de remoção de StopWords, que retirou do texto as palavras que foram consideradas como irrelevantes durante o treinamento, com isso foi criada a Bag of Words (BoW), que pode ser visto na Figura 21, de forma que apenas as palavras “OBRIGADO”, “PARABENS” e “HORRIVEL” não foram retiradas do comentário.

Figura 21 - Bag of Words.


Fonte: Elaborada pelo autor.

O comentário então foi submetido ao processo de Stemming que utilizou as palavras contidas no Bag of Words e deixou apenas os radicais das palavras que foram cadastradas, o resultado do Stemming pode ser visto na Figura 22.

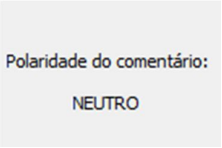
Figura 22 - Aplicação do Stemming.


Fonte: Elaborada pelo autor.

Das três palavras que restaram na Bag of Words, apenas a palavra “OBRIGADO” tinha radical registrado, visto que apenas ele havia sido cadastrada, dessa forma o processo de Stemming só foi registrada para ela, gerando a palavra “OBRIGAD”.

Por fim o comentário geral gerado pelo Stemming foi submetido ao cálculo probabilístico de Naive Bayes, realizando o calculo palavra a palavra contida na bag of words para assim mostra a probabilidade de um comentário pertencer a cada classe e dessa forma o comentário é classificando dentro da polaridade com maior porcentagem, como pode ser visto na Figura 23.

Figura 23 - Aplicação do Stemming.




Fonte: Elaborada pelo autor.

Nesta ocasião o comentário foi classificado de forma incorreta, como o comentário apresenta ironia em seu conteúdo, isso fez com que o classificador identificasse o comentário como neutro, quando na realidade deveria ser classificado como negativo.

Na Tabela 1 estão registrados todos os comentários utilizados para realização dos testes, com as classificações feitas pelo autor e a classificação feita pelo sistema.

Tabela 1 – Comentários usados para teste.

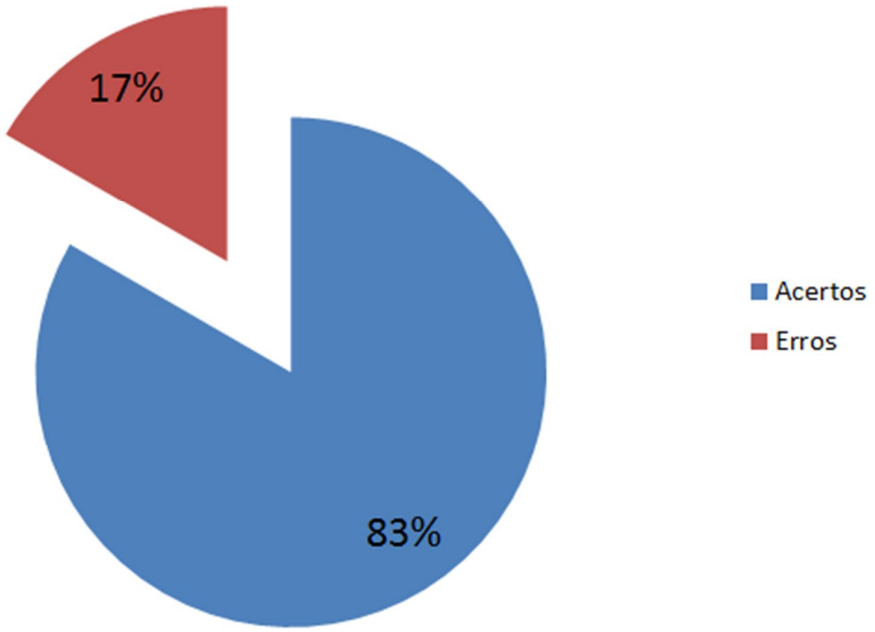
| Comentário | Avaliação do autor | Avaliação do Software |
|---|---|---|
| Obrigadu, o serviço é otimuh | Positivo | Neutro |

| | | |
|---|---|---|
| Muito Obrigado, vocês estão de parabens por esse serviço horrivel | Negativo | Neutro |
| Muito Obrigado, esse serviço é uma porcaria. | Negativo | Negativo |
| Falta de respeito, propaganda enganosa. | Negativo | Negativo |
| Não funciona, uma porcaria. | Negativo | Negativo |
| Funcionou perfeitamente, estou muito satisfeito, obrigado. | Positivo | Positivo |
| Ainda estou avaliando o serviço, nenhum problema até agora. | Neutro | Neutro |
| Muito obrigado pelo ótimo serviço | Positivo | Positivo |
| Gostaria de dar os parabéns pelo ótimo serviço prestado | Positivo | Positivo |
| O serviço não esta funcionando, uma porcaria. | Negativo | Negativo |
| Falta de respeito ao consumidor, péssimo atendimento. | Negativo | Negativo |
| Tudo em ordem até agora | Neutro | Neutro |

Fonte: Elaborada pelo autor.

Após realizar os cálculos o sistema obteve uma taxa de acerto de 83%, errando apenas 17% dos comentários registrados, conforme pode ser observar na Figura 24. É muito importante salientar que a taxa de acertos do sistema tende a crescer de acordo com o seu treinamento, quanto mais comentários bem definidos houver no Corpus e palavras bem filtradas na Bag of Words, melhor tende a ser o processo.

Figura 24 - Gráfico de acertos.




Fonte: Elaborada pelo autor.

Dentre os comentários que foram classificados de forma incorreta, foi detectada a presença de ironia, erros de digitação e também de abreviações, porém esses assuntos que não foram previstos e podem dar origem a novos trabalhos.

# 13 CONSIDERAÇÕES FINAIS

O presente trabalho teve o objetivo de desenvolver um sistema de análise de sentimentos utilizando linguagem de programação Delphi e FireBird como um gerenciador de banco de dados, o qual obteve uma precisão de 83% de acerto.

Sobre as tecnologias utilizadas, ou seja, o Delphi e o FireBird 2.5 elas atendem perfeitamente a todos os requisitos necessários, o Delphi conta com vários recursos para manipulação de texto que já são nativos do programa e também existem classes disponibilizadas por usuários da comunidade para esse tipo de atividade.

O algoritmo de Naive Bayes teve uma taxa de acertos satisfatória para o que foi proposto, mostrando-se um algoritmo simples de ser implementado e com um desempenho aceitável.

Um de seus pontos fracos é a necessidade de um Corpus de comentário muito bem cadastrado para que as classes do algoritmo Bayesiano tenham uma boa base para a realização dos cálculos.

O algoritmo de Naive Bayes é um algoritmo Booleano, ou seja, ele só pode dizer se um comentário pertence a determinada classe ou não, com isso não é possível medir o grau de satisfação ou de insatisfação dos comentários de uma forma mais profunda para saber o quanto feliz ou descontente um usuário está com seu serviço.

Como trabalhos futuros propõe-se a melhoria do software aplicando a ele corretores para corrigir os possíveis erros existentes nos comentários, bem como, um dicionário de abreviações, para que as respectivas palavras sejam utilizadas durante o processo e por fim aprofundar as pesquisas para encontrar formas de tratar a ironia nas frases e classificá-las corretamente, buscando melhorar o processamento do Software.

Com base neste trabalho, podemos concluir que o mesmo contribui para os estudos no campo da ciência da computação, em especial a área de inteligência artificial, voltado para processamento de linguagem natural e análise de sentimentos, mostrando a implementação do algoritmo Bayesiano, destacando as principais dificuldades na área, e a precisão, que consiste na taxa de acertos do software, apresentada pelo mesmo em sua execução, outros parâmetros como o F-measure e o Recall não foram utilizados para medir o desempenho do software.

O software também pode ser utilizado em empresas prestadoras de serviço que buscam vantagens competitivas no mercado e preocupam-se com a opinião de seus usuários, fornecendo assim algumas informações que seriam de grande valia.

## REFERÊNCIAS

BAEZA-YATES, R., RIBEIRO-NETO, B,. **Recuperação de Informações** – Conceitos e Tecnologia das Máquinas de Busca – 2 ed. São Paulo: Techbook, 2011.

BEAULIEU, A. **Apendendo SQL**. São Paulo: Novatec Editora, 2010

BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML:** Guia do Usuário. Rio de Janeiro: ELSEVIER, 2005.

CLAUDIO, A. Introdução a teste de sofware. **DEVMEDIA**, Rio de Janeiro, 201-. Disponível em: <http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035>. Acesso em: 26 maio. 2016.

CRAIG, R. D., JASKIEL, S. P. **Systematic Software Testing**. Boston: Artech House Publishers, 2002.

CARRILHO JUNIOR, J. R. **DESENVOLVIMENTO DE UMA METODOLOGIA PARA MINERAÇÃO DE TEXTOS.** 2008. 96f. Pontifícia Universidade Católica do Rio de Janeiro – PUC-RIO, Rio de Janeiro, 2007. Disponível em: <http://www.maxwell.vrac.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=11675@1>. Acesso em: 20 maio. 2016.

CARVALHO FILHO, J. A. **MINERAÇÃO DE TEXTOS: ANÁLISE DE SENTIMENTO UTILIZANDO TWEETS REFERENTES À COPA DO MUNDO 2014**. 2014. 44 f. Universidade Federal do Ceará, Quixadá, 2014. Disponível em: <http://www.repositoriobib.ufc.br/000017/0000179f.pdf>. Acesso em: 20 abr. 2016.

DALEPIANE, F. Entenda a Delphi Language. **DEVMEDIA**, Rio de Janeiro, 2014. Disponível em: <http://www.devmedia.com.br/entenda-a-delphi-language/31353>. Acesso em: 26 maio. 2016.

DE AMO, S. **TÉCNICAS DE MINERAÇÃO DE DADOS**. 2004. 43 f. Universidade Federal de Uberlândia, Uberlândia, 2004, Disponível em < http://www.deamo.prof.ufu.br/arquivos/JAI-cap5.pdf>. Acessado em 6 de ago. de 2016.

DIAS DA SILVA, B. C. et al. **Introdução ao Processamento das Línguas Naturais e Algumas Aplicações**. 2007. 119 f. Série de Relatórios do Núcleo Interinstitucional de Linguística Computacional, São Carlos, 2007. Disponível em: <http://docplayer.com.br/332720-Introducao-ao-processamento-das-linguas-naturais-e-algumas-aplicacoes.html>. Acesso em: 17 abr. 2016.

FACELLI, K., LORENA, A. C., GAMA, J., CARVALHO, A. C. P. de L. F. de. **Inteligência Artificial:** uma abordagem de aprendizado de máquina**.** São Paulo: Gen, 2011.

FERNANDES, A. M. da R. **Inteligência Artificial:** noções gerais. São Paulo: VisualBook, 2008.

GOLDSCHMIDT, R., PASSOS, E., BEZZERA, E,. **Data Mining** – Conceitos, técnicas, algoritmos, orientações e aplicações – 2. ed. Rio de Janeiro: Elsevier Editora, 2015.

HISTORICAL References. Firebird. 20--. Disponível em: <http://www.firebirdsql.org/en/historical-reference/>. Acesso em: 26 maio. 2016.

LAUDON, K., LAUDON J. **Sistemas de Informação Gerenciais**. São Paulo: Pearson, 2011.

LUCCA, G., PEREIRA, I. A., PRISCO, A., BORGES, E. N. **UMA IMPLEMENTAÇÃO DO ALGORITMO DE NAIVE BAYES PARA CLASSIFICAÇÃO DE TEXTO**. 2013. 4 f. Centro de Ciências Computacionais – Universidade Federal do Rio Grande, Rio Grande, 2013. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/erbd/2013/0019.pdf>. Acesso em: 20 ago. 2016.

MALDONADO, J. C., BARBOSA, E. F., VINCENZI, A. M. R., DELAMARO, M. E., SOUZA, S. do R. S de, JINO, M. **INTRODUÇÃO AO TESTE DE SOFTWARE**. 2004. 56 f. Instituto de Ciências Matemáticas e de Computação, São Carlos, 2004. Disponível em: <http://pbjug-grupo-de-usuarios-java-da-paraiba.1393240.n2.nabble.com/attachment/2545944/0/Introducao%20Ao%20Teste%20De%20Software%20-%20Maldonado,Jose.pdf>. Acesso em: 15 maio. 2016.

PRESSMAN, R. S. et al. **Engenharia de software** – Uma Abordagem Profissional – 7. ed. São Paulo: ARTMED, 2006.

RAD Studio Application Showcase. **Embarcadero**. 20--. Disponível em: <https://www.embarcadero.com/br/products/rad-studio/application-showcase>. Acesso em: 26 maio. 2016.

RAPPS, S.; WEYUKER, E.J. **Data Flow analysis techniques for test data selection** In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, p. 272-278, Tokio, Set. 1982.

RIZZATO, F. **Embarcadero**: Um pouco da história para começar nossa conversa. GIZMODO, São Paulo, 2014. Disponível em: <http://gizmodo.uol.com.br/canais/embarcadero/embarcadero-um-pouco-de-historia-para-comecar-nossa-conversa/>. Acesso em: 26 maio. 2016.

ROCHA, A. R. C.et al. **Qualidade de software**: teoria e prática. São Paulo: Prentice Hall, 2001.

ROSA, J. L. G. **Fundamentos da Inteligência Artificial**. Rio de Janeiro: Gen-LTC, 2011.

SANTOS, F.**MINERAÇÃO DE OPINIÃO EM TEXTOS OPINATIVOS UTILIZANDO ALGORITMOS DE CLASSIFICAÇÃO**. 2013. 71 f. Universidade de Brasília, Brasília, 2013, Disponível em: < http://bdm.unb.br/bitstream/10483/7711/1/2013_FernandoLeandrodosSantos.pdf >. Acesso em: 7 ago. 2016.

SANTOS, L. M. **PROTÓTIPO PARA MINERAÇÃO DE OPINIÃO EM REDES SOCIAIS: ESTUDO DE CASOS SELECIONADOS USANDO O TWITTER**. 2007. 103 f. Universidade Federal de Lavras, Lavras, 2010, Disponível em: <http://repositorio.ufla.br/bitstream/1/5190/1/MONOGRAFIA_Prototipo_para_minerac ao_de_opiniao_em_redes_sociais_estudo_de_casos_selecionados_usando_o_twitt er.pdfl>. Acesso em: 17 abr. 2016.

SILVA, H. da. **TeO: um chatterbot para telessaúde**. 2014. 103 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração, Bauru, 2014.

SILVA, J. R. da. **MINERAÇÃO DE OPINIÕES APLICADAS A MÍDIAS SOCIAIS PARA AS ORGANIZAÇÕES**. 2015. 51 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração, Bauru, 2015.