Dionnys Santos Marinho

# DESENVOLVIMENTO DA PLATAFORMA DE INTEGRAÇÃO, ATUALIZAÇÃO E VISUALIZAÇÃO DE DADOS DA SENTIMENTALL

Palmas – TO
2020

Dionnys Santos Marinho

# DESENVOLVIMENTO DA PLATAFORMA DE INTEGRAÇÃO, ATUALIZAÇÃO E VISUALIZAÇÃO DE DADOS DA SENTIMENTALL

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientadora: Profa. Dra. Parcilene Fernandes de Brito.

Palmas – TO

2020

Dionnys Santos Marinho

# DESENVOLVIMENTO DA PLATAFORMA DE INTEGRAÇÃO, ATUALIZAÇÃO E VISUALIZAÇÃO DE DADOS DA SENTIMENTALL

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientadora: Profa. Dra. Parcilene Fernandes de Brito.

Aprovado em: _____/_____/_____

BANCA EXAMINADORA

_____

Profa. D.ra. Parcilene Fernandes de Brito

Orientadora

Centro Universitário Luterano de Palmas – CEULP

_____

Prof. M.e Jackson Gomes de Souza

Centro Universitário Luterano de Palmas – CEULP

_____

Prof. M,e Fabiano Fagundes

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2020

## RESUMO

MARINHO, Dionnys Santos. **Desenvolvimento da plataforma de integração, atualização e visualização de dados da SentimentALL**. 2019. 10 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Ciência da Computação, Centro Universitário Luterano de Palmas, Palmas/TO, 2019.

Este trabalho apresenta o desenvolvimento de uma plataforma Web para o gerenciamento de processos e a visualização da informação da ferramenta SentimentALL. Essa ferramenta utiliza técnicas de processamento de linguagem natural e análise de sentimentos para definir a polaridade (Positivo ou Negativo) da opinião de autores sobre diversos aspectos no contexto do turismo nacional. Os dados processados e analisados pela ferramenta são de comentários textuais oriundos do site TripAdvisor direcionados a hotéis, restaurantes e atrações. Dois módulos foram criados para a plataforma. O primeiro módulo trata do gerenciamento do processo de obtenção contínua de dados brutos do site TripAdvisor. Este módulo possui uma interface para execução, agendamento e gerenciamento de Crawlers usados na extração, possibilitando que o usuário administrador escolha o intervalo em que as execuções de atualização da base de dados deverão acontecer. O segundo módulo desenvolvido apresenta graficamente dados obtidos a partir do processo de Análise de Sentimentos da ferramenta. Para que a visualização de informação seja eficiente, a estrutura de dados foi remodelada e otimizada para consultas que acontecem em tempo real, garantindo sempre uma visão atualizada dos dados da SentimentALL. Testes comparando os bancos de dados da versão dois e da nova versão proposta também foram executados, assim foi possível garantir que as alterações feitas no Banco de Dados da aplicação tiveram o efeito esperado na redução do tempo de consultas. A apresentação desses dados, oriundos dos processos de análise de sentimentos da ferramenta, é feita através de métodos de visualização da informação com o uso de gráficos e tabelas que proporcionam um melhor entendimento das informações analisadas sobre o turismo no Brasil.

## LISTA DE FIGURAS

**LISTA DE TABELAS**

## LISTA DE ABREVIATURAS E SIGLAS

AS – Análise de Sentimentos

BD – Banco de Dados

HTML - *HyperText Markup Language*

HTTP - *Hypertext Transfer Protocol*

PLN – Processamento de Linguagem Natural

PoS – *Part of Speech*

SQL - *Structured Query Language*

URL - *Uniform Resource Locator*

**SUMÁRIO**

# 1. INTRODUÇÃO

O uso da internet como meio para expressar opiniões sobre produtos, empresas e lugares faz com que um grande volume de dados seja gerado diariamente. Assim, o desenvolvimento de soluções computacionais para o tratamento e análise desses dados é essencial para diversas indústrias. A ferramenta SentimentALL, objeto de estudo deste trabalho, surgiu dessa necessidade. A SentimentALL é uma ferramenta de análise de dados que recebe um grande volume de dados brutos, e não estruturados, na forma de avaliações textuais e define de forma automática se o autor da avaliação expressou sentimentos positivos ou negativos sobre aspectos avaliados. Segundo Brito (2018) e Araújo (2017), a ferramenta SentimentALL tem como objetivo realizar a Análise de Sentimentos (AS) de opiniões de sites da internet escritos na língua portuguesa.

A versão atual da SentimentALL, apresentada por Araújo (2017), utiliza comentários do site TripAdvisor (https://www.tripadvisor.com.br/), e oferece uma análise de sentimentos no contexto do turismo nacional. No terceiro trimestre de 2019, o site somava um total de 830 milhões de avaliações sobre uma grande listagem de atrações, hotéis e restaurantes no mundo todo (TRIPADVISOR, 2019). Esses tipos de dados são valiosos, segundo Pang e Lee (2008), pois opiniões de indivíduos sobre algo sempre foi uma informação importante no processo de tomada de decisões.

O gerenciamento dos processos de extração da segunda versão da ferramenta desenvolvida por Araújo (2017) oferece poucas opções de controle de execução, o que dificulta a atualização contínua da base de dados brutos a serem analisados. Quanto a visualização de dados, a ferramenta é complementada por um dashboard desenvolvido em Sousa (2017), que permite a exploração de dados oriundos do processo de análise, em um formato quantitativo e direto, mas com poucas opções para exploração de dados do processo que trazem informações dos aspectos avaliados.

A complexidade dos sistemas de extração de dados, análise de sentimentos e visualização de dados da ferramenta deu origem ao problema de pesquisa que é: Como desenvolver uma plataforma web que ofereça uma interface para gerenciamento do processo de extração, consulta em tempo real e visualização da informação obtida no processo de análise de sentimentos da SentimentALL? A partir disso, o trabalho apresentado tem como objetivo apresentar a construção da plataforma web para a SentimentALL, em que suas

etapas de extração de dados, análise de sentimentos e visualização da informação sejam interligadas e estruturadas.

Para a construção dessa plataforma, o sistema foi estruturado em dois módulos. O primeiro tem como objetivo o gerenciamento da execução do processo de obtenção de dados brutos feito com o uso de Crawlers. Crawlers ou Spiders são programas que visitam e analisam a estrutura de páginas web obtidas a partir de uma lista de URLs iniciais e funciona de forma recursiva a partir de novas URLS encontrados no momento da extração de dados (DIKAIAKOS; STASSOPOULOU; PAPAGEORGIOU, 2005). O trabalho apresenta a definição do módulo para extração e a adição de uma interface que será usada no gerenciamento de execuções programáticas do processo de extração e manutenção do projeto onde Crawlers da SentimentALL estão organizados. Este módulo também integra os processos de análise de sentimentos. Isso garante que, ao final do processo de extração de novos comentários brutos do site, o sistema inicie o processamento e análise desses dados.

O processo de análise de sentimentos da SentimentALL resulta em um grande volume de dados. O banco de dados (BD) e o ajuste de consultas são particularmente críticos em aplicativos que lidam com grandes quantidades de dados (NEVEDROV, 2006). A desnormalização pode ser descrita como um processo para reduzir o grau de normalização e complexidade do modelo de dados com o objetivo de melhorar o desempenho do processamento de consultas (SANDERS; SHIN, 2001). A desnormalização foi usada como método para preparar o Banco de dados da plataforma, apresentando como resultado um novo modelo de dados para a ferramenta SentimentALL.

O segundo módulo da plataforma é responsável pela visualização dos dados analisados e armazenados no novo modelo proposto. O uso de gráfico oferece uma forma simplificada para o entendimento de uma grande quantidade de informações. A visualização de informação envolve o aparato sensorial humano primário, a visão, e em conjunto com o poder de processamento da mente humana, é um meio simples e eficaz para comunicar informações complexas e/ou volumosas (SCHROEDER; LORENSEN; MARTIN, 2004, pág. 1). O foco desse módulo da plataforma é a apresentação de dados que expõe características avaliadas e contextualizam para visitantes da plataforma o sentimento de turistas ao avaliar aspectos específicos de um destino.

A criação de índices mais genéricos para um conjunto de palavras-chave pode melhorar o processo de quantificação da informação obtida no processo de AS, a técnica de Lematização foi utilizada neste trabalho para a criação desses índices. Segundo Ingason et. al (2008), Lematização é uma técnica de normalização que tem o propósito de criar uma conexão entre palavras ou formas de palavras relacionadas, é uma etapa importante em

processos de classificação de texto e extração de informação. De forma sucinta, esse processo permite que palavras que são diretamente derivadas e que tenham um significado semelhante sejam consideradas iguais a partir da geração de um novo termo único. Um índice mais genérico para aspectos avaliados e palavras opinativas permite que ao consultar e apresentar esses dados, o sistema considere um maior número de dados como iguais, sendo assim, mais significativo ao representar essa informação.

Seções subsequentes do trabalho apresentam de forma estruturada o desenvolvimento da plataforma. A seção 2 apresenta um embasamento teórico sobre o projeto e dá uma visão geral de processos e tecnologias envolvidas. A seção 3 detalha etapas que formam a metodologia usada para o desenvolvimento da plataforma. A seção 4 expõe os resultados obtidos, apresentando o desenvolvimento dos módulos que compõe a plataforma e melhorias feitas no modelo de dados e inclusão de algoritmos para lematização de palavras opinativas e aspectos avaliados. Ao final serão apresentadas considerações sobre o trabalho e a possibilidade de trabalhos futuros.

# 2. REFERENCIAL TEÓRICO

Esta seção apresenta conceitos importantes para o entendimento do trabalho realizado. A seção 2.1 apresenta Crawlers como um método para obtenção de dados da Web, em seguida, conceitos e técnicas da área de processamento de linguagem natural (subseção 2.2) usados nos processos de análise são citados. A seção 2.3 apresenta a desnormalização como um método para otimização de bancos de dados (BD). A seção 2.6 apresenta técnicas de visualização da informação.

## 2.1 EXTRAÇÃO DE DADOS DA WEB

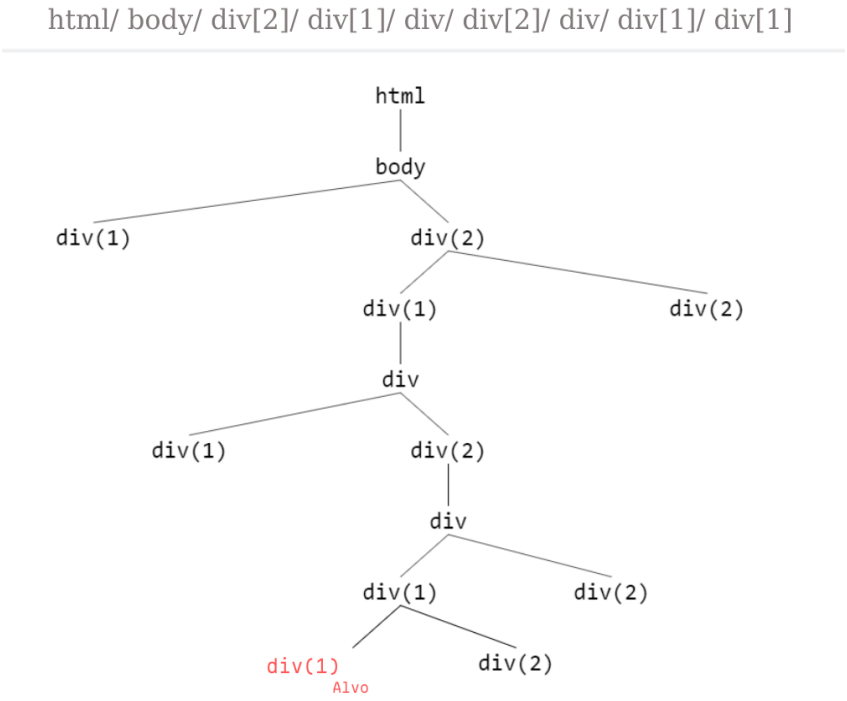
A extração automática de dados da web é uma tarefa complexa. Acessar e obter dados pouco estruturados de fontes externas dificulta o trabalho de ferramentas que executam essa tarefa. Frameworks e métodos automatizados para obtenção de dados da Web começaram a ser desenvolvidos devido a essa demanda.

Grande parte das informações disponíveis na web está em páginas HTML de sites, blogs e redes sociais. A obtenção automática desses dados pode ser feita a partir do uso de técnicas, como, por exemplo, os Crawlers, também conhecidos como Robôs ou Spiders. Crawlers são programas que visitam e analisam a estrutura de páginas web obtidas a partir de uma lista inicial de links e funciona de forma recursiva a partir de novos links encontrados (DIKAIAKOS; STASSOPOULOU; PAPAGEORGIOU, 2005). Esses programas são capazes de seguir links e extrair uma grande quantidade de informação, nem sempre completamente estruturadas, como é o caso de comentários escritos.

Na pesquisa de Ferrara et al. (2014) diversas técnicas e aplicações para a extração de dados da Web são abordadas, demonstrando a grande variedade de recursos para utilização dessas informações que envolvem sistemas complexos e técnicas de obtenção e processamento de dados. Dentre os domínios apresentados, a mineração de opinião se posiciona entre as principais aplicações que consomem dados extraídos da Web. Esse campo de estudo será abordado em detalhes em seções subsequentes do trabalho.

Web scraping é uma das tarefas executadas por Crawlers. Uma variante mais recente de rastreadores da Web é a Web raspadores, que visa procurar certos tipos de informações como preços, descrições e comentários (VARGIU; URRU, 2013). Crawlers que executam o processo de raspagem possibilitam a criação de bases de dados com informações estruturadas a partir de sites e contextos específicos. Para isso, seletores CSS ou Xpath podem ser utilizados. A Figura 1 exemplifica o formato de árvore do HTML e o uso de seletores Xpath para acesso direto a nós dessa estrutura.

**Figura 1 - Seletor Xpath em Árvore de Estrutura HTML**

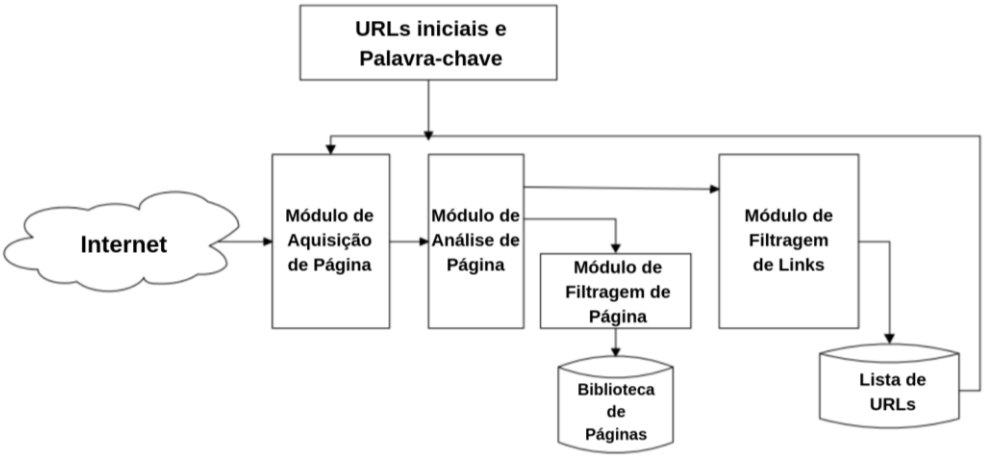



Esta seção apresenta conceitos importantes para o entendimento do trabalho que será realizado. A seção 2.1 apresenta Crawlers como um método para obtenção de dados da Web, em seguida, conceitos e técnicas da área de processamento de linguagem natural (subseção 2.2) usados nos processos de análise e mineração de dados textuais são citados. A seção 2.3 apresenta a desnormalização como um método para otimização de bancos de dados (BD). A seção 2.6 apresenta técnicas de visualização da informação.

### 2.1.1 Crawlers Focados em Contexto

Crawlers focados em contexto são programas que têm a tarefa de raspar páginas relevantes através de extrações, análises e filtros para obter informações úteis (XIE; XIA, 2014). Diferente de Crawlers gerais que, segundo Menczer (2001), geralmente são utilizados por mecanismos de pesquisa e procuram obter informação do maior número de páginas possíveis independente do contexto. Crawlers focados procuram por páginas com informações específicas, das quais possa ser obtido um conjunto de informações pré-definido com base no contexto em que a informação será utilizada. A Figura 2 ilustra a estrutura desse tipo de Crawler.

**Figura 2 - Estrutura de Crawler Focado em Contexto**




Fonte: Xie e Xia, 2014, p. 488

A Figura 2 ilustra a arquitetura de um Crawler Focado, que primeiro recebe URLs iniciais e palavras-chave. Essas informações são usadas como ponto de partida para a busca de páginas que pertencem ao contexto explorado. As URLs iniciais e as palavras-chave são passadas para o módulo de aquisição de páginas Web, que utiliza as URLs iniciais e executa requisições HTTP para obter todo o conteúdo da página HTML. As páginas retornadas são avaliadas no Módulo de Análise de páginas que utiliza as palavras-chave como parâmetro de busca para páginas do contexto, em seguida esse módulo é responsável por filtrar páginas que não estão dentro do contexto buscado. Após esse filtro, as páginas são salvas na biblioteca de páginas que pertencem ao contexto. Além de buscar por páginas específicas, o Web Crawler focado também procura por novas URLs (links), e estes são enviados para o módulo de filtragem de links que define se as URLs encontradas são válidas para o contexto. Os links válidos são salvos e então usados como entrada para que o processo se repita até que todas as páginas e URLs do contexto tenham sido visitadas.

A Ferramenta SentimentALL utiliza Crawlers focado em contexto desenvolvido com o Scrapy, sendo este o principal componente do processo de extração. Scrapy é um framework que possui uma arquitetura clara, flexível e conveniente, que oferece ótimos resultados em processos de raspagem (XIE; XIA, 2014). O contexto dos Crawlers apresentados e implementados pela ferramenta SentimentALL são páginas HTML do site TripAdvisor que possuem informações de avaliações direcionada a hotéis, restaurantes e atrações. Seções subsequentes do trabalho apresentam técnicas aplicadas sobre os dados textuais brutos obtido após o processo de extração.

## 2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

Segundo Russell e Norvig (2013), Processamento de Linguagem Natural (PLN) é um conjunto de técnicas utilizadas para que computadores possam adquirir informação com base na linguagem escrita. Processamento de Linguagem Natural refere-se a técnicas computacionais envolvendo linguagem (GRUS, 2016). Técnicas de PLN são aplicadas com o intuito de estruturar e obter informações de textos também conhecidos como corpus. As seções subsequentes explicitam algumas técnicas que são parte de processos de PLN.

Para analisar avaliações obtidas na etapa de extração feito pela ferramenta SentimentaLL, um conjunto de técnicas de PLN é usado com o objetivo de estruturar o corpus, permitindo que o computador possa analisar esses dados. Algumas das principais técnicas usadas no processamento de dados brutos são: Tokenização, correção ortográfica e PoS Tagging, essas técnicas são essenciais para o processamento de textos obtidos da web que serão usados por algoritmos de mineração de dados como o de Análise de Sentimentos. Nas seções subsequentes do trabalho essas técnicas serão abordadas com mais detalhes sobre seu funcionamento e utilidade como parte da ferramenta SentimentALL.

### 2.2.1 Tokenização

A Tokenização é um dos primeiros passos para a obtenção de informação a partir de dados textuais. Essa tarefa está entre as técnicas primárias da área de PLN. Encontrar os limites das palavras e símbolos, também conhecidos como tokens, define o processo de Tokenização. A etapa de Tokenização permite organizar um texto em pedaços menores e linguisticamente significativos (PENTHEROUDAKIS; BRADLEE; KNOLL, 2001). Essa técnica é aplicada em processos que envolvem mineração de informações de textos, como, por exemplo, análise de sentimento. A Tokenização de um corpus pode ter dois níveis de detalhamento, por frases ou tokens.

A Tokenização do texto em sentenças busca identificar no corpus o limite das sentenças. Há um conjunto de caracteres de pontuação que normalmente introduzem os limites das sentenças em um corpus, os principais são ponto de interrogação “?”, ponto de exclamação “!”, reticências “...”, dois pontos “:”, ponto e vírgula “;” e o ponto final “.” (JURISH; WÜRZNER, 2013). O Exemplo 1 apresenta um texto com duas frases separadas por um ponto.

Exemplo 1: “*A cachoeira é de fácil acesso. Você pode ir de carro até a entrada.*”

Após o processo de Tokenização de sentenças, o resultado é (“A cachoeira é de fácil acesso.”, “Você pode ir de carro até a entrada.”). Assim, a divisão resulta em uma lista de

sentenças. Essa estruturação torna mais fácil a interpretação automática feita por algoritmos executados sobre essas listas de sentenças, sendo este processo importante pois resulta em um corpus mais estruturado, facilitando sua análise. O resultado do processo de Tokenização de sentenças também é essencial para a análise de sentimentos. No Exemplo 1, ao aplicar a Tokenização de sentenças, o resultado apresenta na primeira frase um texto opinativo, e na segunda um fato sobre o local, desta forma, ao avaliar o contexto dessas sentenças separadamente torna a identificação de sentenças e palavras opinativas menos complexa.

O segundo nível da Tokenização, sendo este ainda mais detalhado, é o de palavras e/ou símbolos, chamados de tokens. Estes são uma das menores partes significativas do texto. Segundo Jurish e Würzner (2013), o uso de espaços em branco como delimitadores dos limites de cada token é comum, porém a pontuação também deve ser considerada, o que geralmente gera um nível de complexidade maior para a tarefa. O Exemplo 2 apresenta um corpus antes de passar pelo processo de Tokenização por palavras.

Exemplo 2: *"A cachoeira fica ao final de uma pequena trilha... muito relaxante!".*

No exemplo 2, o resultado do processo de Tokenização é ("A" , "cachoeira", "fica", "ao", "final", "de", "uma, "pequena", "trilha", "...", "muito", "relaxante", "!"), a lista resultante possui as palavras apresentadas no corpus de entrada e também os símbolos de reticências e o ponto de exclamação. A correta aplicação desse processo é importante para sistemas que executam análises sobre dados textuais, visto que etapas de análises mais detalhadas, como a de AS, utilizam esses dados pré-processados para a definição da polaridade de aspectos avaliados no texto. A Tokenização viabiliza a análise de cada parte do texto de forma isolada e detalhada. As técnicas apresentadas a seguir tem como base um corpus que tenha passado pelo processo de Tokenização nesse nível de detalhamento.

### 2.2.2 Correção Ortográfica

A correção ortográfica é importante para o processo de mineração de informação em textos extraídos da internet. Segundo Asghar e Kundi (2014), a maioria dos comentários e avaliações da internet tem como característica a pouca preocupação com regras e padrões gramaticais, resultando em erros ortográficos. A ocorrência dessas palavras escritas de forma incorreta pode causar problemas em análises textuais. Técnicas automatizadas para a correção de termos gramaticalmente incorretos podem ser aplicadas e, com isso, aumentar a precisão de sistemas que executam o processo de AS como a SentimentALL.

Um conjunto de palavras conhecidas pode ser usado como parâmetro para encontrar e corrigir termos incorretos. De forma sucinta, palavras que não pertencem a essa base são

identificadas como possíveis erros gramaticais. Alguns desafios devem ser levados em conta nesse processo. Quando uma palavra incomum aparece em um texto, a possibilidade de que essa palavra seja apenas uma palavra rara é maior do que seja um erro gramatical (DAMERAU; MAYS, 1989). Para a ferramenta SentimentALL, palavras conhecidas foram obtidas a partir de sites de notícias. Artigos de notícias geralmente são mais longos e mais bem estruturados do que mensagens vistas nas redes sociais, bate-papo ou de e-mail (AGGARWAL; ZHAI, 2012).

Ao identificar uma palavra incomum que não está presente no corpus de palavras conhecidas, transformações serão feitas e com isso serão geradas palavras candidatas. Segundo Damerau e Mays (1989), as transformações que proporcionam um conjunto de palavras candidatas mais eficiente são:

- Adicionar uma letra.
- Deletar uma letra.
- Substituir uma letra por outra.
- Troca de posição letra adjacentes

Caso alguma das palavras candidatas obtidas a partir das transformações apareça entre as palavras conhecidas da base de dados, a correção é feita, caso contrário a palavra incomum é mantida.

### 2.2.3 Part-of-Speech Tagging

PoS Tagging é uma técnica de PLN que consiste em atribuir para cada palavra de um texto uma tag morfossintática correta para seu contexto (MARQUEZ; RODRIGUEZ, 1998). Essas tags, chamadas de Part-of-Speech (PoS), são compostas principalmente por classes gramaticais, como verbos, substantivos e adjetivos. PoS Tagging também é conhecido como o processo de etiquetagem morfológica de um corpus. Classes gramaticais são parte de uma estruturação morfológica da linguagem e representa o significado da palavra ou token no texto. Part-of-Speech tags são úteis, pois revelam muito sobre a palavra analisada e palavras vizinhas (JURAFSKY e MARTIN, 2000). PoS Tags também ajudam no processo de identificação de opiniões, conforme afirmação de Liu (2012, p. 32), “foi mostrado que os adjetivos são indicadores importantes de opiniões”.

Um dos desafios para a classificação e definição de PoS Tagging são as ambiguidades. “A ambiguidade ocorre quando uma palavra possui mais de um PoS tag possível, como por exemplo, “para”, que pode ser um verbo (parar) ou uma preposição” (ARAÚJO, 2017, pág. 19). Para melhorar o nível de acerto no caso de ambiguidades durante o processo de PoS Tagging, modelos estatísticos podem ser usados para classificar corretamente qual tag melhor representa a utilização da palavra em um dado contexto. A

abordagem baseada em regras para o processo de PoS Tagging é possível, porém devido ao grande esforço para definir essas regras, o uso de técnicas baseadas no aprendizado de máquinas é mais comum (FONSECA; ROSA, 2013). O uso de abordagens estatísticas que envolvem aprendizado de máquinas depende de grandes bases de dados previamente marcados. Para marcar uma base de dados que seja utilizável no processo de treino de teste de modelos estatísticos, são usados tagsets pré definidos. A Tabela 2 apresenta o modelo de notação do tagset da base de dados Mac-Morpho.

**Tabela 1 - Tagset base de dados Mac-Morpho**

| CLASSE GRAMATICAL | TAG |
|---|---|
| ADJETIVO | ADJ |
| ADVÉRBIO | ADV |
| ARTIGO | ART |
| CONJUNÇÃO COORDENATIVA | KC |
| CONJUNÇÃO SUBORDINATIVA | KS |
| INTERJEIÇÃO | IN |
| NOME | N |
| NOME PRÓPRIO | NPROP |
| NUMERAL | NUM |
| PARTICÍPIO | PCP |

Fonte: Aluísio et. al (2003)

Mac-Morpho é um corpus de textos em português Brasileiro de dados marcados com Part-of-speech tags (ALUÍSIO et. al, 2003). Este é um dos mais populares datasets usados no processo de PoS Tagging para o Português do Brasil. A base de dados é composta 1.1 milhões de palavras do jornal Folha de São Paulo, essas notícias foram marcadas com tags denotadas a partir de processos automáticos e complementados com correções e anotações manuais.

Um exemplo de técnica de aprendizado para classificação de Part-of-speech tags é a utilização de modelos estatísticos com base em um conjunto de palavras e tags chamado de Bigram, um dos modelos conhecidos para essa aplicação é o Modelo Oculto de Makrov (Hidden Markov Model - HMM). A marcação de PoS tags usando um modelo de Markov pode ser considerada como uma instância de inferência bayesiana (HUANG; EIDELMAN; HARPER, 2009). De forma sucinta, a abordagem determina que a probabilidade de uma palavra receber uma tag depende da palavra em si e da tag anterior, utilizando uma tupla com a estrutura (tag Anterior, Palavra Atual). Nesse processo, o modelo estatístico utiliza a

probabilidade de possíveis tags para palavra atual em conjunto com a tag da palavra anterior. O uso da probabilidade conjunta aumenta o acerto do processo, visto que utiliza o contexto do token na frase. Esse tipo de modelo estatístico é descrito em detalhes no trabalho de Huang, Eidelman e Harper (2009). Assim, o uso dessa técnica utiliza o potencial de algoritmos de aprendizado de máquina e um corpus robusto como o Mac-Morpho, oferecendo resultados precisos para o processo de PoS Tagging.

### 2.2.4 Lematização e Stemming

"Stemming é o processo de combinar as formas variantes de uma palavra em uma representação comum" (ORENGO; HUYCK, 2001. pág. 186). Essa técnica busca reduzir e representar uma palavra avaliada para uma forma mais comum, removendo prefixos e sufixos. No momento da identificação dessa raiz (stem) da palavra, o contexto em que a palavra original foi utilizada não é determinante para o termo resultado. Encontrar o stem de palavras se tornou uma tarefa importante dentre as técnicas de PLN, principalmente em sistemas de indexação, onde a busca por palavras deve encontrar valores que possuem sentidos iguais, porém formatos temporais e verbais distintos. Um exemplo do processo de Stemming pode ser obtido a partir das palavras local, localização, localizado, resultando no stem "local" como um termo equivalente. Como o stem de um termo representa um conceito mais amplo que o original, o processo aumenta o número de documentos recuperados em um sistema de recuperação de informação (WATZLAWIK; VALSINER, 2012).

A lematização lida com a obtenção do lema de uma palavra que envolve a redução à sua forma raiz de acordo com o contexto da palavra na sentença especificada (WATZLAWIK; VALSINER, 2012). As técnicas de Lematização e Stemming são usadas no processo de normalização da forma de palavras. Segundo Korenius et. al (2014), a aplicação de técnicas para normalização da forma de termos auxilia sistemas que executam a busca por informações, visto que reduz o número total de índices (entradas), causando a expansão de resultados, encontrando palavras variantes do mesmo termo.

Processos de Stemming e Lematização são semelhantes, de modo que o uso das técnicas possui objetivos similares, porém ao aplicar cada técnica o resultado é diferente. Enquanto o resultado do Stemming de palavras é definido principalmente pela raiz do termo que se mantém inalterada em versões semelhantes, gerando resultados que são parte de uma palavra (palavra truncada) e não uma palavra completa, a Lematização busca trocar prefixo e sufixo da palavra de forma que o resultado seja uma nova palavra equivalente, geralmente a palavra origem, porém no infinitivo. A Tabela comparativa das técnicas apresenta palavras semelhantes e o resultado de cada após a aplicação das técnicas de Stemming e Lematização.

**Tabela 2 - Comparativo de resultados dos processos de Stemming e Lematização**

| Palavras de Entradas | Stemming | Lematização |
|---|---|---|
| restaurante, restaurantes | restaurant | restaurante |
| hotéis, hotelaria, hoteleiros | hot | hotel |
| gostei, gostou, gostamos | gost | gostar |

A Tabela apresenta o resultado dos conceitos de uma forma comparativa. Os resultados de cada técnicas são semelhantes, porém a lematização oferece uma forma mais eficiente de representação dos termos de entrada. Os benefícios da lematização e Stemming são semelhantes, pois criam índices únicos para termos similares, porém, os termos resultantes ao usar a lematização são mais significativos, visto que, representam formas flexionadas de palavras e não termos truncados e ambíguos como no caso do processo de Stemming (KORENIUS et. al, 2004).

### 2.2.5 Análise de Sentimentos

A análise de sentimentos (AS), também conhecida como mineração de opinião, é um campo de estudo que analisa opiniões, sentimentos e avaliações de pessoas direcionados a entidades como produtos, serviços e organizações (LIU, 2012). Processos de AS são compostos por técnicas da área de Processamento de Linguagem Natural (PLN).

"As opiniões são centrais para quase todas as atividades humanas porque são fundamentais influenciadores dos nossos comportamentos" (LIU, 2012, pág. 8). Compreender como os produtos e serviços de uma empresa são avaliados é algo valioso para outros consumidores e para a própria empresa que pode tomar decisões estratégicas com base nessa informação. Opiniões e conceitos relacionados como sentimentos, avaliações, atitudes e emoções são elementos de estudo da área de análise de sentimentos e mineração de opinião (LIU, 2012). Visto isso, a análise de sentimentos pode ser dividida em duas tarefas básicas. O reconhecimento da emoção e a detecção de polaridade (CAMBRIA et al., 2017).

A primeira tarefa base de AS é o reconhecimento de emoção, que corresponde ao processamento que define a emoção expressada em um corpus através de um conjunto de emoções básicas como raiva, tristeza e alegria. A segunda tarefa corresponde a detecção de polaridade, geralmente binária de um aspecto ou corpus, produzindo resultados como "positivo" ou "negativo", comumente denotados por valores numéricos 1 e -1 respectivamente. A ferramenta SentimentALL utiliza a segunda forma de classificação.

Uma das formas comuns para a definição da polaridade de uma opinião é feita através do uso de léxicos de sentimentos. "Léxicos e conjuntos de dados de sentimentos representam

a base de conhecimento que constitui um sistema de AS" (JOSHI, BHATTACHARYYA e AHIRE, 2017, pág. 85). A Tabela seguinte apresenta uma amostra do SentLex-PT, um léxico de sentimento para português.

**Tabela 3 - Amostra de dados Léxico SentLex-PT**

| Palavra | PoS | Polaridade |
|---|---|---|
| amável | Adjetivo | 1 |
| ambicioso | Adjetivo | 0 |
| ambidestro | Adjetivo | 0 |
| ambiguidade | Substantivo | -1 |
| ameaça | Substantivo | -1 |

Fonte: (CARVALHO; SILVA, 2015)

A estrutura de um léxico de sentimentos busca representar a palavra e sua polaridade associada mais comum. No exemplo da Tabela 4, além da palavra e da polaridade, a base de dados do SentLex-PT também possui marcações de qual Part-of-Speech tag a palavra foi associada, essa informação serve como um segundo parâmetro comparativo no momento da definição da polaridade durante o processo de AS.

Para definir a polaridade de um aspecto é importante que o corpus analisado tenha passado pelas etapas de processamento, como as apresentadas anteriormente. Esse conjunto de técnicas estão presente nos processos realizados pela ferramenta SentimentALL.

## 2.3 DESNORMALIZAÇÃO

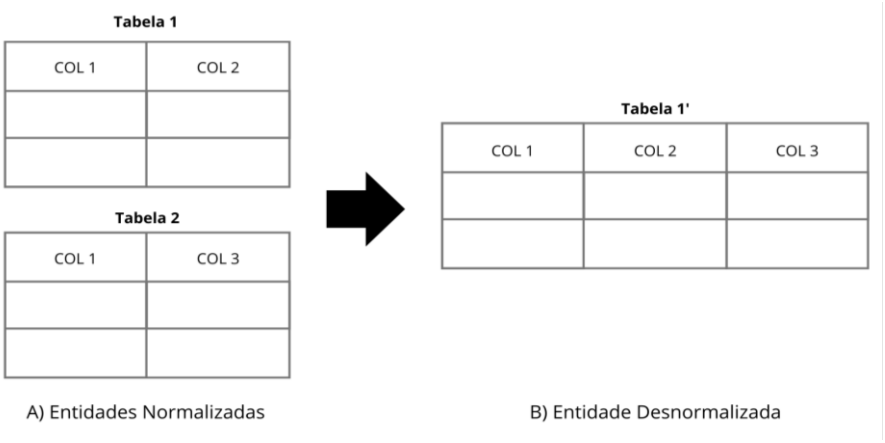
Bancos de dados tradicionais costumam aplicar técnicas de normalização no processo de definição do modelo lógico de um projeto. A normalização é utilizada com o objetivo de criar um conjunto de tabelas relacionais com quantidade mínima de dados redundantes que podem ser modificados de maneira consistente e correta (BAHMANI; NAGHIBZADEH; BAHMANI, 2008). O modelo normalizado de um BD busca, através da criação de tabelas e relacionamentos, diminuir a redundância de dados. A normalização pode causar ineficiências significativas em cenários onde são feitas poucas atualizações e muitas consultas, que poderão envolver um grande número de *joins* (Junções) (SHIN; SANDERS, 2006).

"A desnormalização é um esforço que busca otimizar desempenho, mantendo a integridade dos dados" (PINTO, 2009, pág. 44). De forma sucinta, a desnormalização é um conjunto de técnicas que se propõe a aplicar o processo inverso à normalização com o

objetivo único de melhorar a performance de um modelo de dados em relação ao tempo de execução de consultas. Segundo Shin e Sanders (2006), existem duas formas básicas para a desnormalização, a redução de tabelas intermediárias através da modificação inicial do diagrama entidade relacionamento e a criação de atributos redundantes.

Na primeira, o diagrama entidade relacionamento é usado para reduzir o número de tabelas diminuindo a quantidade de tabelas intermediárias que representam relacionamentos muitos para muitos, principalmente em tabelas que possuem poucos dados descritivos. No segundo, a desnormalização é feita criando entidades ou atributos para facilitar consultas especiais. A Figura 3 ilustra o processo de desnormalização responsável pela compactação de duas tabelas em uma chamada de Pré-junção de tabelas.

**Figura 3 - Desnormalização de Tabelas do Banco de Dados**




Fonte: Sanders e Shin (2001)

O modelo A de entidades normalizadas da Figura 3 apresenta a estrutura normalizada com duas tabelas. No exemplo, ambas possuem um relacionamento direto de um para um representado pela COL 1. No modelo B, que corresponde ao formato desnormalizados das mesmas tabelas do exemplo, a COL 1 que representa a chave estrangeira do relacionamento das tabelas do primeiro exemplo foi mantida e os dados relacionados COL 2 e COL 3 foram agregados à nova tabela desnormalizada, assim fazendo parte de uma única entidade (Tabela 1'). Com a aplicação dessa técnica, o modelo se torna mais simples e a complexidade computacional que envolvia consultar várias tabelas foi eliminado. Segundo Sanders e Shin (2001), a junção de duas tabelas no processo de normalização permite diminuir o tempo de acesso, visto que o modelo possui menos objetos físicos, o que reduz a sobrecarga em consultas. Assim, a simplificação do modelo de dados, além de diminuir a complexidade da criação de consultas, também diminui o tempo de recuperação desses dados.

No trabalho de Pinto (2009) apresenta um framework para a desnormalização de modelos de dados, a descrição do framework traz como objetivo uma criação de tabelas

desmoralizadas que estão posicionadas junto a um conjunto de tabelas normalizadas, dessa forma um modelo desmoralizado do modelo original fica disponível para tarefas que envolvem consultas de tabelas com muitos dados. Algumas das técnicas e descrições apresentadas no trabalho estão listadas na Tabela 4 apresentada abaixo.

**Tabela 4 - Técnicas de Desnormalização**

| Técnica | Descrição de Uso |
|---|---|
| Pré-junção de tabelas | Criação de uma nova tabela resultado da junção de duas outras tabelas que são consultadas em conjunto regularmente |
| Tabelas de relatório | Criação de uma nova tabela que mantém cálculos derivados de junções e manipulações de outras tabelas do banco. |
| Separação de tabelas | Criação de uma nova tabela resultado da fragmentação de uma tabela genérica em duas ou mais tabelas mais específicas. |
| Tabela de hierarquias | Criação de tabela que armazene estruturas hierárquicas presente no modelo original. |
| Redundância de colunas | Criação de colunas em uma tabela para armazenar dados de outra tabela relacionada. |
| Coluna de dados derivados | Criação de coluna para que os dados sejam pré-calculados e armazenados. |

Fonte: (PINTO, 2009)

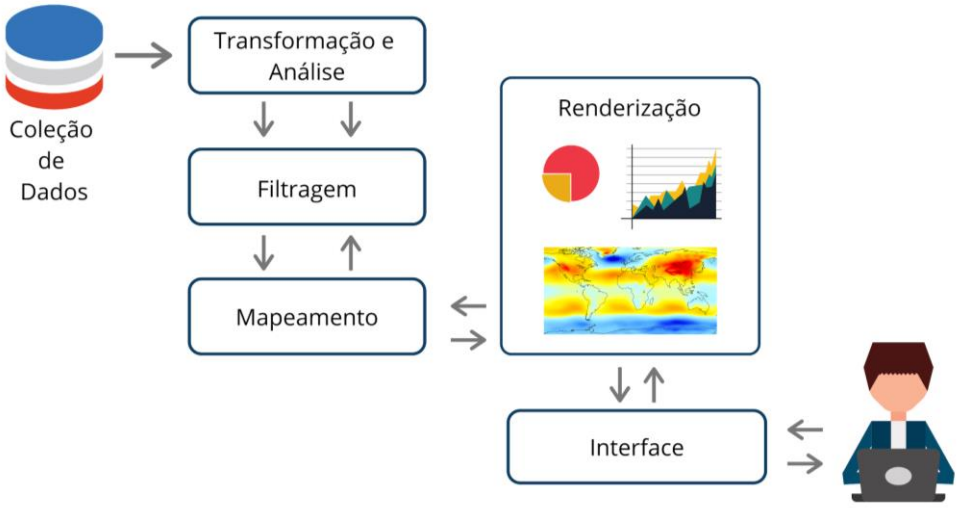
As técnicas apresentadas na Tabela 4 são exemplos de processos de desnormalização que causam um impacto positivo quando os devidos critérios são atendidos e percebidos em um modelo normalizado. Além das técnicas apresentas a criação de exibições conhecidas como *views,* é uma opção rápida de definição e utilização de modelos desnormalizados, porém nem sempre esse método se mostra eficaz. A maioria dos SGBDs (Sistemas de Gerenciamento de Bancos de Dados) processa exibições em tempo de execução, de modo que uma exibição não resolve o desempenho, mas aumenta a facilidade de uso (SANDERS; SHIN, 2001).

## 2.4 VISUALIZAÇÃO DA INFORMAÇÃO

A visualização da informação é o estudo da transformação de dados, informações e conhecimentos em representações visuais interativas que ajudam na formação de modelos mentais (LIU SHI et. al, 2014). Ao oferecer formas diferentes de interpretar dados, sistemas aumentam o entendimento que poderá ser obtido sobre o dataset fonte, assim, auxiliando na percepção de nuances.

Segundo Chen et. al (2008), para o desenvolvidos de sistemas de visualização da informação, são tomadas decisões sobre quais técnicas de exploração de dados, estilos, layout, mapas de cores, etc. serão usados, até que uma coleção de formas de visualização da informação satisfatória seja obtida. A Figura 4 exemplifica o modelo da arquitetura de sistemas para visualização da informação.

**Figura 4 - Arquitetura de Sistema de Visualização da Informação**




Fonte: (LIU SHI et. al, 2014)

A Figura 4 apresenta o fluxo de interações de um sistema que oferece a visualização da informação a usuários. Primeiro é necessário que exista uma base de dados sobre determinado contexto. Em seguida esses dados devem passar por processos de transformação e análise. Nessa etapa técnicas de PLN e mineração de informação, como a análise de sentimento, são aplicadas para que seja possível obter informações significativas de um grande volume de dados, assim, a partir de um conjunto semi estruturado ou não estruturado é possível obter informações importantes para o contexto.

As informações obtidas no processo de análise são então filtradas com base no contexto que cada parte representa. Nessa etapa, um dos objetivos, por exemplo, pode ser a obtenção de informações que estão organizadas em um intervalo de tempo específico, a partir desses dados filtrados e apresentados visualmente poderão oferecer uma visão mais ampla do significado da informação disponível. A etapa de filtragem também é importante em casos onde muitos dados estão disponíveis, auxiliando na performance da ferramenta.

A etapa de mapeamento consiste na adequação dos dados de forma que representem valores quantitativos que poderão ser comparados por representações de volume, posição, cores etc. A renderização corresponde a etapa seguinte e busca transformar dados mapeados em formas que serão mais facilmente entendidas pelos seres humanos. Entre o sistema e o

usuário final deve haver uma interface que permita a interação do usuário com os dados, oferecendo métodos para analisar o significado da informação apresentada e alterar parâmetros específicos, assim, quem interage com a informação pode visualizá-la de diferentes formas. Segundo Hu (2017) como apresentar dados complexos para um público é o principal desafio do campo de visualização da informação, principalmente considerando que os conjuntos de dados a serem analisados tendem a ser mais complexos e crescem com o tempo.

Informações quantitativas podem ser visualizadas através de gráficos estatísticos, esse tipo de visualização pode oferecer um entendimento maior sobre os dados quando usado de forma correta. Segundo Tufte (1983), alguns fatores são essenciais para a correta utilização de visualizações de informações quantitativas, esses fatores são:

- Mostrar os dados;
- Induzir o usuário a pensar sobre o significado da informação e não na metodologia e/ou design do gráfico;
- Não distorcer o  significado dos dados;
- Apresentar muito números em um espaço pequeno;
- Tornar coerentes grandes volumes de dados;
- Encorajar os olhos a comparar diferenças;
- Revelar níveis diferentes de detalhamento;
- Ter um propósito claro: descrições, exploração de dados, tabulação;
- Estar integrado com as estatísticas e descrições dos dados.

A partir dos estudos iniciais de Tufte (1983), é possível perceber que ao representar graficamente dados com o uso dos processos de transformação, filtragem e mapeamento, formas simples que evidenciam os dados e a informação obtida podem ser mais eficientes.

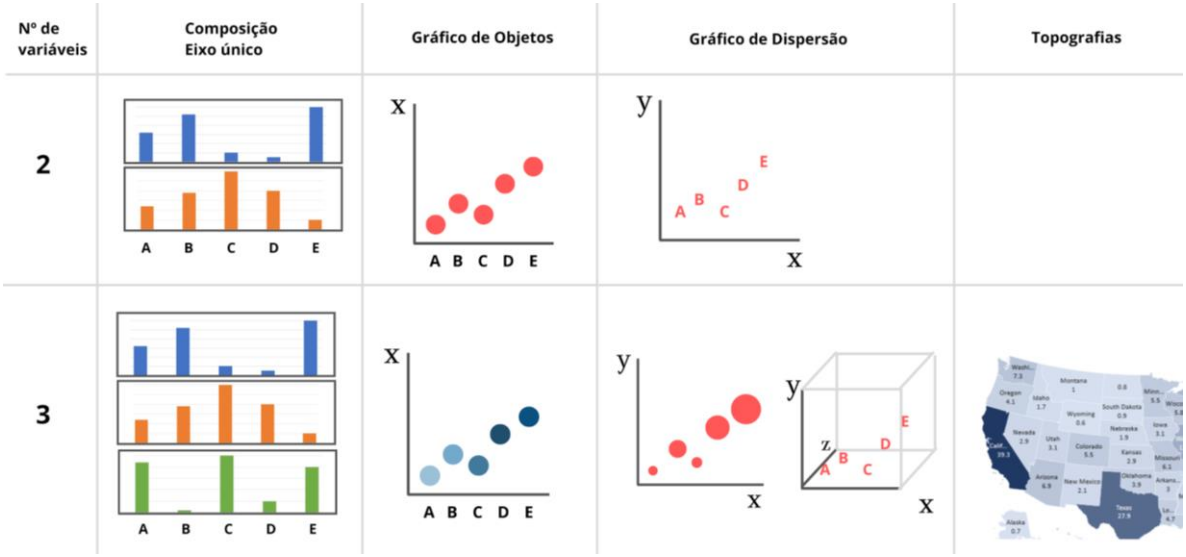
As formas e características estudadas em Bertin (2010) são propriedade de variáveis visuais importantes, que ajudam a transmitir informação visualmente. Os seguintes passos para desenvolvimento de representações de informação em gráficos são apresentados por Card (2008):

- Determinar quais variáveis da abstração devem ser mapeadas em posição espacial na estrutura visual.
- Combine esses mapeamentos para aumentar a dimensionalidade (por exemplo, através da agregação de dados).
- Usar variáveis da retina como forma de adicionar mais dimensões.
- Adicionar controles para interação

- Considere recursos relativos à atenção para expandir o espaço e gerenciar atenção

A Figura 5 apresenta exemplos de gráficos com dimensões diferentes usados para visualização de dados com duas e três variáveis.

**Figura 5 - Exemplos de gráficos com duas e três variáveis**




Fonte: CARD (2008)

A quantidade de várias afeta diretamente na forma como a informação deve ser apresentada visualmente. A composição de eixo único de gráficos apresenta a forma mais simplificada de representar múltiplas variáveis sem adicionar um número maior de eixos/dimensões. Gráficos de objetos representam uma dimensão e a informação de labels, possui um modelo mais simples no caso em que apenas duas variáveis devem ser representadas, mas pode ser expandido para três variáveis com o uso de uma escala de cores ou tamanhos. Gráficos de dispersão são comumente usados também na forma de barras, bolhas e pontos, esse tipo de gráfico representa de uma maneira mais direta informações de dados com duas e três variáveis. Esse tipo de gráfico também oferece o acréscimo na quantidade de eixos, no caso de dados com três variáveis podendo ser representado com três eixos que relacionam quantitativos diferentes a serem apresentados. Topografias são um tipo de gráficos mais complexo que deve funcionar melhor ao representar três ou mais variáveis, o uso de mapas  em gráficos facilita o entendimento principalmente de dados que possuem coordenadas geográficas e posicionamentos específicos que podem ser representados de uma forma mais direta.

A visualização da informação envolve mais que a representação de dados em uma forma gráfica, os gráficos devem ajudar o usuário em ver estrutura nos dados. (UNWIN, CHEN, HÄRDLE, 2007). A partir do entendimento do contexto e a correta utilização das

formas básicas e gráficos para representar a informação, leitores poderão ver além da informação apresentada graficamente, também inferir novos conhecimentos.

# 3. MATERIAIS E MÉTODOS

O trabalho desenvolvido teve características de uma pesquisa aplicada e com fins práticos. Para o desenvolvimento de uma plataforma web de gerenciamento e manutenção do processo de extração e visualização de dados da ferramenta SentimentALL, foi necessário entender o ambiente onde os dados estavam armazenados na sua forma bruta. Esse ambiente será apresentado na seção seguinte. As seções subsequentes apresentam materiais e métodos para o desenvolvimento da plataforma.

## 3.1 AMBIENTE DA COLETA DOS DADOS

A plataforma desenvolvida trata de comentários em português oriundos do site TripAdvisor (https://www.tripadvisor.com.br/). Os dados relacionados a comentários disponíveis no site estão organizados em três categorias principais, hotéis, restaurantes e atrações. As informações apresentadas foram obtidas a partir do site e equivalem a quantidade de hotéis, restaurantes e atrações cadastradas até o terceiro trimestre de 2019, a extração desses dados feita pela plataforma busca a obtenção de dados de destinos brasileiros. As avaliações disponíveis possuem um título, texto da avaliação, uma nota geral, mês e ano em que a visita foi feita, e a quantidade de votos recebidos por outros usuários que indicam o comentário como útil.

No site TripAdvisor, hotéis somam um total de 3.101 estabelecimentos, distribuídos em 993 cidades brasileiras. Esses estabelecimentos estão organizados por tipos de acomodações que são: hotéis, pousadas, albergue, cabana, resort, rancho, condomínio, chalé, acampamento, hotel especializado, tudo incluído, vila, hotel fazenda e hotel para motoristas. Cada avaliação possui a data da estadia. Os hotéis avaliados recebem uma pontuação com base no total geral de avaliações. Além da pontuação geral na escala *likert* de 0 a 5, os seguintes aspectos poderão ser avaliados: localização, limpeza, atendimento, custo-benefício.

Os restaurantes estão distribuídos por 5.253 cidades brasileiras. Os restaurantes cadastrados possuem informação sobre os tipos de refeições que servem dentre 56 tipos diferentes que vão desde Pizza e Fast-food até comidas típicas como brasileiras, sul-americana, chinesa etc. Para restaurantes também é possível informar o tipo de refeição, que são: Café da manhã, almoço e jantar. Os usuários poderão avaliar os tipos de preços do restaurante, que são classificados em: baratos, moderados, sofisticados. Além da pontuação geral na escala *likert* de 0 a 5, os restaurantes também possuem pontuações para os seguintes aspectos: comida, serviço, preço.

Cerca de 3.990 atrações brasileiras também são avaliadas por usuário do TripAdvisor. Além de esportes, excursões e passeios ao ar livre, são mais de 26 tipos de atrações. Ao avaliar, o usuário do site também pode informar os seguintes tipos de categorias de visitantes:

Romântica, Família (Crianças pequenas), família (Adolescentes), amigos, negócios e a sós. Atrações avaliadas recebem uma pontuação geral na escala *likert* de 0 a 5, e a duração sugerida para a visita.

O ambiente usado para a coleta dos dados é organizado e possui além dos textos e comentários, informações complementares que auxiliam na classificação de destinos turísticos do Brasil. O site também se mostra estável, o que é uma característica importante para o trabalho, visto que a obtenção desses dados abertos é feita através de acessos ao site. Assim, com base no conjunto de informações do TripAdvisor, foi possível obter uma grande quantidade de dados que pode auxiliar no entendimento de como é avaliado os destinos turísticos no Brasil.

## 3.2 MATERIAIS

A seguintes ferramentas, bibliotecas e tecnologias foram usadas no desenvolvimento da plataforma Web:

**SentimentALL** é uma ferramenta implementada em Python que objetiva a análise de sentimento de avaliações escritas na língua portuguesa. É uma das partes principais da plataforma, utilizada no processamento de dados brutos, mais precisamente oferecendo dados sobre a opinião de usuário do site TripAdvisor direcionadas a aspectos relacionados a hotéis, restaurantes e atrações de diversos destinos no Brasil.

**Spacy** é uma biblioteca de código aberto para processamento de linguagem natural, escrita nas linguagens de programação Python e Cython. Foi utilizada no processo de normalização de palavras opinativas e aspectos avaliados, através da aplicação de um modelo pré-treinado que executa o processo de lematização de palavras.

**JMeter** é um software de código aberto totalmente escrito em Java, projetado para carregar o comportamento funcional de teste e medir o desempenho de sistemas. Por oferecer uma interface simples, essa ferramenta de testes foi usada para simular vários acessos e avaliar a estrutura de dados usada pela plataforma. Os resultados dos testes garantem que o novo modelo proposto possui um desempenho superior a estrutura original.

**Celery** é uma biblioteca *open source* Python que realiza a execução de processos de uma fila de tarefas. Essa biblioteca é focada na operação em tempo real, mas também suporta a definição de execuções de processos de forma programáticas através de agendamentos. Essa biblioteca é usada no controle de execuções dos processos de análise de sentimentos e na criação de agendamentos de execuções dessas etapas.

**Scrapy** é um framework *open source* usado no desenvolvimento de Web Crawlers. O projeto Scrapy oferece uma base extensível onde foi possível adaptar e criar algoritmos capazes de extrair dados estruturados de páginas HTML do TripAdvisor.

**Scrapyd** é uma aplicação utilizada para o controle de projetos Scrapy. Possui um sistema que permite o envio de projetos e o gerenciamento de execuções e agendamentos do processo de extração através de uma API json. Esse sistema foi usado como parte do sistema de gerenciamento da execução do processo de extração.

**Django** é um *framework* desenvolvido na linguagem Python. Oferece uma estrutura para o desenvolvimento rápido de aplicações web, com suporte a estrutura MVC que garante uma estruturação para o projeto e possibilita o desenvolvimento rápido e iterativo. O *framework* é usado na consulta e acesso a dados da plataforma. Django é usado no acesso aos dados principalmente nas etapas de pré-processamento e análise de dados da Sentimentall, o framework também é utilizado para criação da API que consulta os dados exibidos no módulo de visualização da informação. Esse *framework* foi escolhido por utilizar a linguagem de programação Python, que também é usada nos processos da ferramenta SentimentALL, permitindo uma integração maior com o projeto.

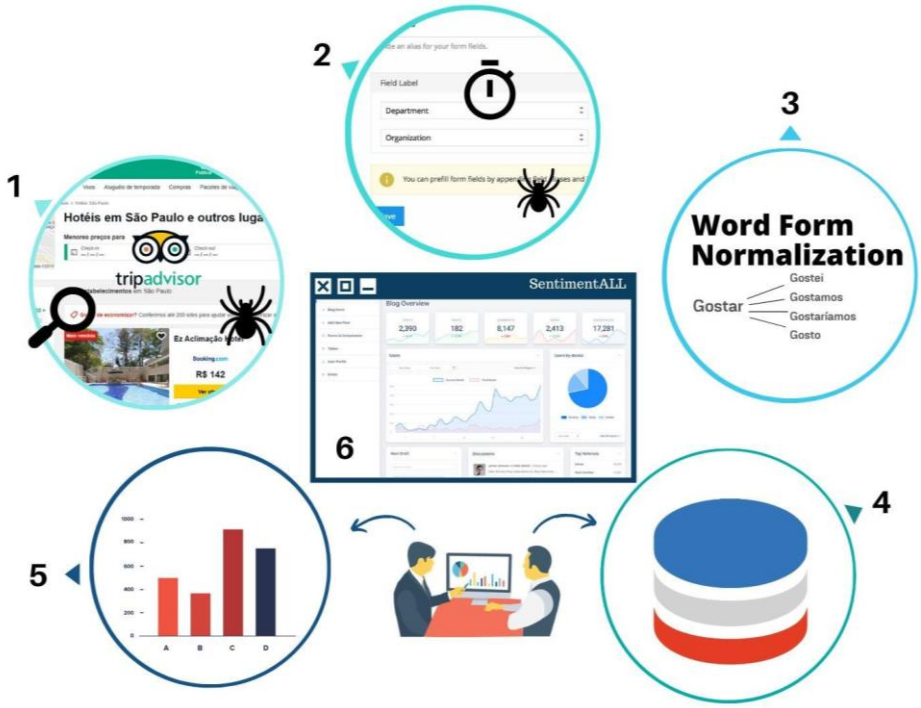
**Angular** é um *framework front-end* para o desenvolvimento de aplicações Web e *mobile*. Utiliza a linguagem TypeScript e fornece uma estrutura modularizada para construção de aplicações escaláveis. Será usado no desenvolvimento dos módulos e estruturação do *front-end* da plataforma.

**HighCharts** é uma biblioteca de código aberto que permite o uso de gráficos JavaScript simples e limpos, baseados em HTML5. É usada na apresentação de informações oriundas do processo de análise de sentimentos feitos pela SentimentALL.

### 3.3 MÉTODOS

Para o desenvolvimento dos módulos que compõem a plataforma SentimentALL um conjunto de etapas foram definidos. A Figura 2 ilustra cada etapa do trabalho e a ordem de execução.

**Figura 6 - Fluxograma de etapas do Trabalho**




A primeira etapa da Figura 6 representa o processo de análise feito nas páginas do *site* TripAdvisor. Essa etapa inicial foi usada para a definição de quais dados deveriam ser extraídos do site alvo e no planejamento de uma estrutura para a extração contínua e sistematizada desses dados. Após essa análise do padrão de páginas HTML do site alvo, um projeto Scrapy foi criado, incluindo 4 Crawlers que foram desenvolvidos para que respectivamente fossem usados na extração de dados de atrações, hotéis, restaurantes ou avaliações.

A segunda etapa ilustra o desenvolvimento do módulo de gerenciamento de Crawlers e do processo de extração usado na atualização da base de dados da plataforma da SentimentALL. Nessa etapa o projeto Scrapy que inclui os quatro Crawlers criado foi modificado. Dentre as modificações feitas, funções foram incluídas para permitir o controle de dados obtidos no processo, permitindo que seja feita uma atualização contínua da base de dados, sem a duplicação de dados existentes.

Ainda na segunda etapa, o módulo teve a inclusão do sistema Scrapyd, que permitiu um controle maior das execuções programáticas da extração de dados. A aplicação Scrapyd web também foi incluída como parte do módulo criado, essa aplicação serve como interface para comandos que iniciam e monitoram os processos de extração. Por fim, esta etapa teve como resultado o primeiro módulo da plataforma que permite o gerenciamento dos processos de extração.

A terceira etapa é referente a implementação de um algoritmo para o processo de Normalização da forma de palavras (em inglês, Word Form Normalization). Nessa etapa, o

algoritmo de lematização da biblioteca spaCy foi usado na definição de um termo comum para palavras que correspondem a aspectos avaliados e palavras opinativas. A aplicação dessa etapa permitiu o uso de termos comuns para palavras com significados idênticos ou muito próximos, principalmente para palavras flexionadas como, por exemplo, das palavras gosta, gostei, gostamos, gosto, que serão normalizadas e definidas pelo termo comum "gostar". A informação da palavra normalizada é armazenada em conjunto aos dados resultado do processo de AS da SentimentALL, dessa forma é possível recuperar, por exemplo, qual o aspecto avaliado na sua forma bruta ou a forma normalizada desse mesmo termo.

A quarta etapa corresponde ao processo de remodelagem do modelo de dados para o uso em consultas que envolvem a visualização de informação na plataforma. Uma avaliação inicial do modelo de dados foi feita, e serviu como métricas comparativas para os resultados do novo modelo proposto. A nova estrutura de dados tem como objetivo a consulta eficiente dos dados da SentimentALL, viabilizando a visualização das informações. Para avaliar as duas versões do Banco de Dados e oferecer dados comparativos, testes foram criados e executados utilizando a ferramenta JMeter. Um agente foi criado, e simulou 100 acessos diretos ao banco na forma de instruções SQL pré-definidas. Um relatório do desempenho do banco foi emitido no final de cada teste. Consultas de agregação comuns como, por exemplo, a quantidade de aspectos avaliados como positivos e negativos de uma determinada cidade, foram usadas como forma de avaliação do modelo original de dados e do modelo de dados proposto.

Técnicas de desnormalização e a criação de índices foram usados para melhorar o tempo de resposta do BD. O resultado dessa etapa são análises do processo e uma estrutura de dados menos complexa que permita que a consulta da informação tenha tempos menores quando comparadas aos testes do modelo de dados original. Visto que o objetivo das modificações no DB é a otimização do tempo de consultas aos dados, essa informação que é obtida nos relatórios de testes do JMeter são indicadores de melhorias do modelo.

As etapas 4 e 5 da metodologia de desenvolvimento da plataforma foram interligadas por reuniões com a especialista do domínio. Essas reuniões foram essenciais para a definição do novo modelo de dados e desenvolvimento do protótipo do módulo de visualização de dados da SentimentALL.

A etapa 5 ilustra o desenvolvimento de um protótipo do módulo de visualização da informação. O objetivo principal dessa etapa foi definir quais informações a serem apresentadas no módulo de visualização da informação. Essa etapa de definição foi importante para orientar etapas seguintes de implementação do módulo de visualização de informação da plataforma.

A sexta etapa ilustra a implementação do módulo de visualização de informação com base no protótipo definido na etapa anterior. O módulo de visualização desenvolvido oferece meios para que o usuário possa interagir com os dados, assim, métricas a serem visualização que foram definidas na etapa de prototipação foram implementadas de forma que parâmetros pudessem ser alterados e diferentes formas de visualização dessa informação fossem disponibilizadas.

# 4. RESULTADOS E DISCUSSÃO

Esta seção do trabalho apresenta como resultados o desenvolvimento dos módulos da plataforma SentimentALL, os resultados obtidos estão organizados em quatro etapas principais, o desenvolvimento do módulo de gerenciamento do processo de extração e atualização contínua, a remodelagem e testes de performance para o Banco de Dados da plataforma, a inclusão de um algoritmo para normalização de aspectos e palavras opinativas e a criação do protótipo e desenvolvimento do módulo de visualização da informação. Cada etapa do desenvolvimento será descrita em detalhes a seguir, acompanhando o fluxo de trabalho como definido na metodologia apresentada.

## 4.1 MÓDULO DE EXTRAÇÃO DE DADOS

Antes de inicial o desenvolvimento do módulo de extração de dados foi necessário uma análise inicial do site TripAdvisor. Por trabalhar com a estrutura HTML dessas páginas, antes da criação de Crawlers para extração e definição de processos é preciso conhecer a estrutura atual do site e como explorá-la de maneira eficiente. A seção subsequente apresenta os resultados dessa análise. Em seguida será apresentado a arquitetura completa do módulo, incluindo descrição para os Crawlers que executam a extração de dados do site alvo e mecanismos para o gerenciamento do processo.

### 4.1.1 análise de dados para extração

Os usuários do site TripAdvisor podem avaliar três tipos de estabelecimentos, hotéis, restaurantes e atrações. Na versão 2 da ferramenta SentimentALL apresentada em Araújo (2017), esses itens avaliados receberam o nome de Objetos. Na versão anterior do processo de extração, como definido em Araújo(2017), os seguintes dados eram obtidos e armazenados: código do objeto, nome, código da cidade onde o objeto está localizado e o tipo (hotel, restaurante ou atração). Nessa versão inicial, foi verificado que muitos dados presentes hoje no TripAdvisor não estavam sendo obtidos durante o processo de extração.

A análise das páginas de Hotéis, Restaurante e Atrações possibilitou a definição de um novo conjunto de dados para extração. Obter mais dados sobre o destino torna o alvo da avaliação uma entidade mais completa dentro da plataforma, agregando valor ao conjunto de dados. A Figura 7 ilustra no formato de tabelas quais dados foram identificados para os tipos de objetos avaliados no site TripAdvisor.

**Figura 7 - Dados de objetos avaliados**

| Restaurantes | Hoteis | Atrações |
|---|---|---|
| Cidade | Código | Código |
| Estado | Cidade | Cidade |
| Nome | Estado | Estado |
| Preço (Barato, Médio, Caro) | Nome | Nome |
| Código | Likert Geral | Tipo (Natureza, Parque, praias) |
| Localização (endereço) | Likert Localização | Escala Likert |
| Colocação (N° x de Y atrações) | Likert Limpeza | Quantidade avaliações em Português |
| Likert Comida | Likert Atendimento | Localização |
| Likert Serviço | Likert Custo-Benefício | Colocação (N° x de Y atrações) |
| Likert preço | Categoria (Estrelas) | url |
| Likert Ambiente | Tipo de Hotel (executivo, etc) | |
| Likert geral | Idiomas Falados | |
| Cozinhas (Brasileira, sul-americana,etc) | Colocação (N° x de Y hotéis) | |
| Quantidade avaliações em Português | Quantidade avaliações em Português | |
| url | Localização (endereço) | |
| | url | |

Com base nessa análise inicial, foi possível identificar que cada tipo de Objeto avaliado apresenta um conjunto distinto de informações, por esse motivo não faz sentido que os três tipos de objetos avaliados sejam tratados da mesma forma. Essas informações influenciam em processos seguintes de definição do modelo de dados e no desenvolvimento do módulo de visualização da informação.

A definição dos conjuntos de dados disponíveis para cada tipo apresentado na Figura 7 foi importante para as etapas seguintes que envolvem o desenvolvimento de Crawlers para acessar e extrair esses dados de forma automática. Parte das informações de objetos disponíveis para serem avaliados são cadastrados pelos donos do estabelecimento, essa é uma das funcionalidades do TripAdvisor que permite que comerciantes inscrevam e divulguem seus negócios. As informações disponibilizadas pelos donos do estabelecimento são, por exemplo, nome, média de preços de um restaurante, tipos de cozinhas, a categoria de estrelas de um hotel, entre outros.

Um segundo tipo de informação é obtido a partir de cálculos feitos pelo TripAdvisor. Esses dados são formas de condensar as avaliações de visitantes. Um exemplo desse tipo de informação são os dados de escala Likert do objeto que variam entre 1 e 5 estrelas, colocação do objeto em relação a objetos semelhantes na mesma cidade, e o total de avaliações disponíveis na língua portuguesa.

Além da análise de páginas de objetos avaliados, também foram definidos o conjunto de dados a serem extraídos de avaliações e autores do TripAdvisor. Os dados definidos para avaliações são: Texto do comentário, título, data da visita, o código identificador do hotel, restaurante ou atração avaliada, quem é o autor da avaliação e a nota do autor para o objeto. A extração de dados também deverá obter os seguintes dados sobre autores: nome, ano de chegada, cidade e o nível do autor no TripAdvisor. A seção seguinte apresenta o

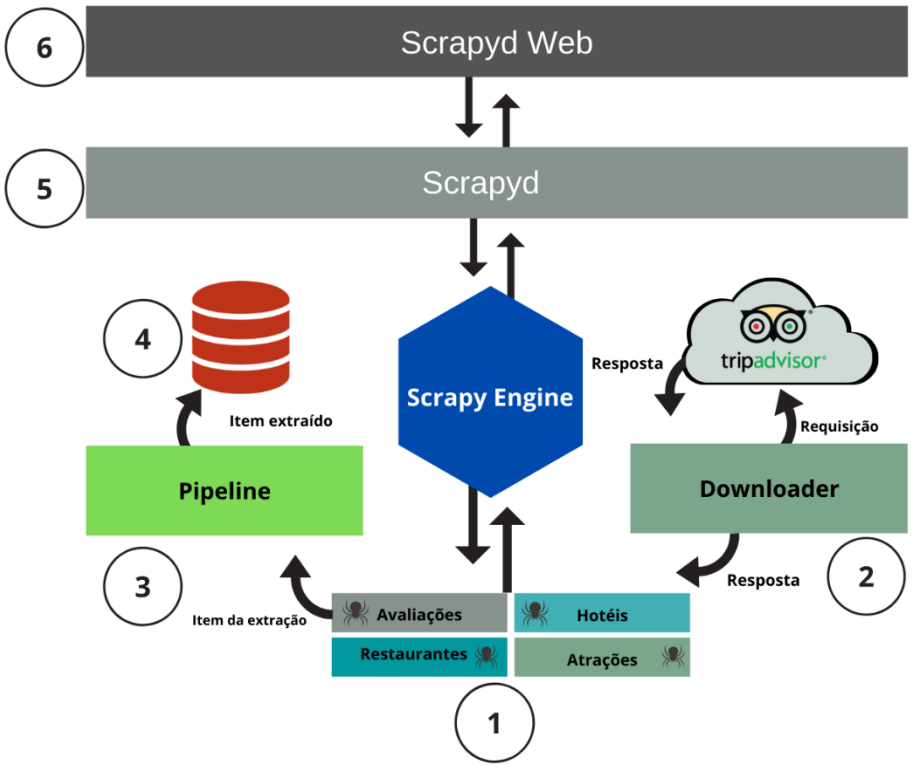
desenvolvimento de Crawlers criados para obter os dados de avaliações, autores e objetos avaliados, como definido nesta etapa de análise das páginas, e também descreve a reestruturação para que esses sistemas possam realizar esse processo de atualização da base de dados de forma contínua.

### 4.1.2 Crawlers e módulo de gerenciamento da extração

Para manter a base de dados da plataforma o mais atualizada possível, o processo para extração de dados foi reformulado utilizando recursos do *framework* para obtenção de dados da Web Scrapy. Todo o módulo de gerenciamento do processo de extração está organização sobre um único projeto Scrapy. O projeto desenvolvido armazena toda a lógica necessária para garantir a extração contínua de dados. Após definir o projeto base, quatro Crawlers foram criados, cada um é responsável por um tipo específico de extração de dados, o objetivo é obter dados sobre hotéis, atrações, restaurantes ou avaliações. Esses dados obtidos seguem o mesmo conjunto de dados como definido na seção anterior do trabalho.

Cada Crawler desenvolvido foi adaptado para navegar e obter dados das páginas alvo da extração. Por padrão essa tecnologia de extração de dados na Web utiliza elementos das páginas HTML do TripAdvisor para encontrar e obter informações. Assim, cada Crawler tem um propósito específico, de forma que, esses sistemas não funcionam corretamente ao serem aplicados a contextos diferentes, por exemplo, não é possível utilizar o Crawler de restaurantes para obter também dados de hotéis ou atrações. Para melhor entender a estrutura do projeto Scrapy e a reestruturação do processo, a Figura 8 ilustra o modelo completo da arquitetura do módulo de extração.

**Figura 8 - Arquitetura do módulo de extração de dados da plataforma SentimentALL**




O item 1 da Figura 8 ilustra os Crawlers criados para o processo de extração de dados da plataforma. Os Crawlers são os principais componentes de controle, responsáveis por armazenar a lógica de obtenção de dados, regras para a busca recursiva de novas páginas e tratamento de dados obtidos. Este item do módulo de extração também tem contato direto com o Scrapy Engine, esse componente centraliza as funções do framework Scrapy.

Cada Crawler desenvolvido utiliza um conjunto de regras para a obtenção recursiva de novas URLs, isso torna o procedimento de extração mais dinâmico, sendo desnecessário ter conhecimento prévio de toda a estrutura e organização do site. Assim, a partir de URLs iniciais mais genéricas, os Crawlers conseguem encontrar outras URLs que tem um padrão conhecido de páginas alvo. Essas páginas são então acessadas e uma nova busca por URLs é feita, proporcionando uma execução recursiva do processo de obtenção de novas páginas e extração de dados. As regras para obtenção de URLs foram introduzidas nesta versão da SentimentALL. A Figura 9 a seguir exibe as regras usadas para na obtenção de informações de hotéis como forma de exemplificar essa nova funcionalidade.

**Figura 9 - Regras para busca de URLs de objetos alvo**

```
rules = (
    1 Rule (
        LinkExtractor(allow=('Hotels-g.+',), deny=('RegistrationController', 'or\d+'))
    ),
    2 Rule(
        LinkExtractor(allow=('Hotel_Review-g.+',), deny=('RegistrationController', 'or\d+')),
        callback='parse_hoteis'
    )
)
```

As regras (*rules*) apresentadas na Figura 9 são definidas como atributos da classe HoteisCrawler, que define o Crawler para extração de dados de Hotéis. Essa estrutura de regras também é usada nos demais Crawlers da plataforma. As regras utilizam padrões utilizando a linguagem de expressões regulares Python, cada padrão é usado na busca de novas URLs.

O item 1 da Figura 9, ilustra uma regra que é definida pela classe Rule, essa primeira regra é utilizada para entrar URLs de páginas intermediárias que dão acesso a páginas de hotéis avaliados. Uma regra é composta por um componente chamado LinkExtractor, essa classe utiliza o atributo é *allow* para definir as expressões regulares usadas na busca de URLs desejadas, e o atributo *deny*, que contém padrões de URLs indesejadas e que por isso devem ser descartadas.

Durante o processo de análise das páginas do site foi possível identificar que as URLs que possuem o padrão “Hotels-g.+”, pertencem a páginas intermediárias que dão acesso às páginas que contém uma listagem de hotéis, por exemplo o padrão da regra 1 da Figura 9 é responsável por encontrar as URLs das páginas que listam todos os hotéis da cidade de Palmas.

A Segunda regra item 2 da Figura 9, utiliza a expressão regular “Hotel_Review-g.+” para encontrar e acessar URLs de um hotel específico, o principal diferencial é o uso da função *call-back*. Em uma regra (Rule) esse atributo identifica qual função deve receber o conteúdo da URL visitada. Em suma, a regra representado no item 1 encontra páginas de listagem de hotéis de uma cidade, em seguida ao acessar essas URLs intermediárias, a regra item 2 da Figura 9 busca URLs de páginas que possuem detalhes de cada hotel, ao acessar essas páginas a função *call-back* recebe o conteúdo HTML e segue para a etapa de extração de dados.

Na sequência dos componentes da arquitetura, o item 2 da Figura 8 apresenta o Downloader. Esse componente é responsável pela criação de requisições e direcionamento de respostas aos Crawlers. O Downloader do projeto também foi modificado. Desta forma, ao iniciar o processo de extração, esse componente recebe a informação de URLs conhecidas salvas no Banco de dados da plataforma. Isso é feito para que o sistema possa ignorar páginas que já passaram pelo processo de extração previamente e por isso não devem ser extraídas novamente, evitando assim duplicações nos dados e garantindo o funcionamento do processo de extração contínua.

Após receber do componente Downloader a resposta (*response*) com o conteúdo HTML, os Crawlers utilizam padrões para navegação dessa estrutura e obtenção de dados. O resultado do processamento feito no Crawler é um objeto do tipo *Item* contendo todos os dados obtidos ao fim do processamento. Esse item é então enviado para o componente Pipeline.

O item 3 da arquitetura apresentada na Figura 8 ilustra o Pipeline, que é o componente responsável pela lógica final de controle do processo de extração. Verificações feitas nessa etapa do processo definem se as informações extraídas pelos Crawlers serão salvas ou não. De forma sucinta, esse componente é responsável pelo controle de qualidade. Diante disso, tem o objetivo de garantir que os dados essenciais foram obtidos, em seguida os dados tratados e completos são armazenados no Banco de dados (BD) da plataforma definido pelo item 4 da Figura 8. Durante as verificações, se o componente identificar informações incompletas ou erros, a função *DropItem* é acionada, fazendo com que o processamento seja interrompido e o item seja descartado.

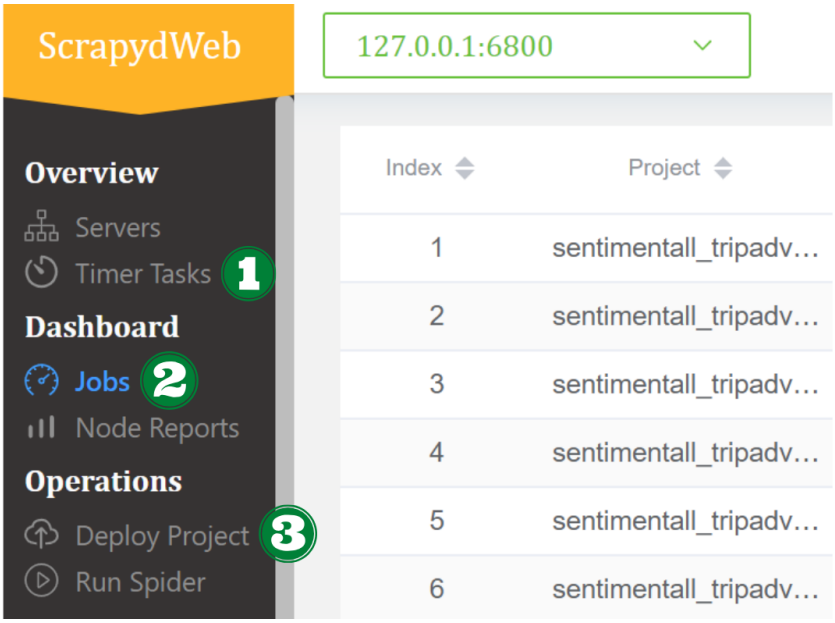
O item 5 da Figura 8 representa a camada superior de conexão ao projeto Scrapy, denominada Scrapyd. O framework Scrapy possui processos complexos, e o controle e monitoramento das extrações não é uma tarefa trivial, visto isso, os desenvolvedores do *framework* disponibilizam o Scrapyd como uma interface de comunicação com projetos Scrapy.

O sistema Scrapyd utiliza um modelo de API Json que oferece rotas para diversas funcionalidades de um projeto Scrapy, como, por exemplo, execução do processo de extração, controle de versões de projetos, agendamento de tarefas de extração, monitoramento em tempo real e controle de Logs etc. Por fim, o item 6 da arquitetura representa a interface gráfica web utilizada no processo de atualização de dados da plataforma, esse item será detalhado na seção seguinte do trabalho.

### 4.1.3 interface de gerenciamento da extração

Para que o processo de extração possa ser feito de forma consistente e contínua é preciso que o gerenciamento da extração aconteça em vários níveis, a interface Scrapyd Web foi utilizada com esse objetivo. Esse sistema oferece um método visual para interação com o processo de extração de dados do TripAdvisor. O Scrapyd Web é um sistema open-source que utiliza as rotas de acesso às funções do processo de extração que são oferecidas pela API json de gerenciamento de projetos Scrapy chamada de Scrapyd. A Figura 12 ilustra a tela do Scrapyd Web e suas principais funcionalidades no gerenciamento do módulo de extração.

**Figura 10 - Interface de gerenciamento do processo de extração**




Na Figura 10 é possível ver que as principais funções estão organizadas no menu direito da aplicação. Para acessar essa parte da plataforma, o usuário deve estar logado. O objetivo dessa interface é oferecer um acesso administrativo ao processo de extração, visto isso, essa parte da plataforma não é acessível ao público.

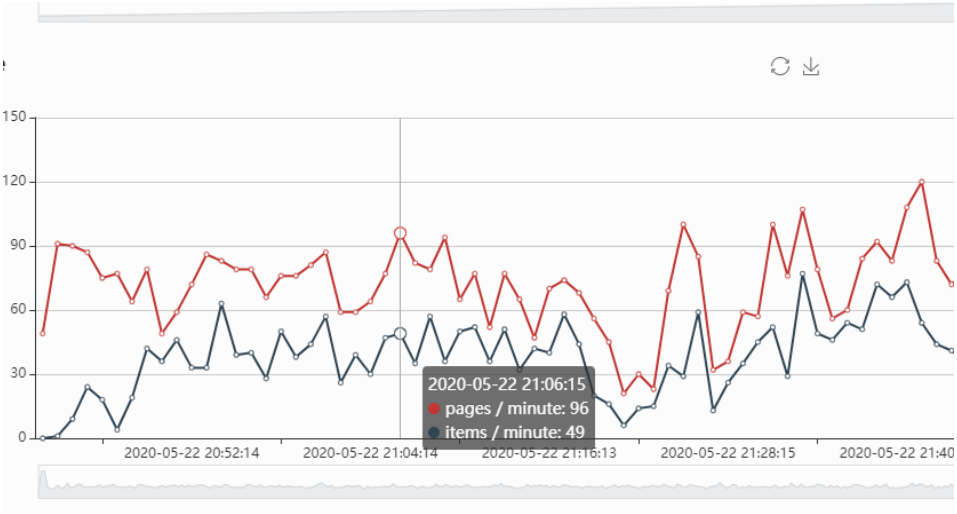
O item 1 da Figura 10 representa a opção para o uso de funções de agendamentos de execuções(Timer Tasks). Essa funcionalidade oferece opções de configurações como, por exemplo, definir para que o Crawler que busca por avaliações faça o processo de extração e atualização de dados uma vez por semana, em dia e horários específicos. O item 2 da Figura 10 ilustra a funcionalidade para monitoramento de execuções do processo de extração(Jobs). Para melhor ilustrar a funcionalidade, a Figura 11 detalha a listagem de tarefas executadas ou em execução para o processo de extração.

**Figura 11 - Listagem de Tarefas e interface de Monitoramento**

| Spider | Job | Pages | Items | Stats | Action |
|---|---|---|---|---|---|
| avaliacoes | 2020-05-28T20_58_30 | 5319 | 4278 | Stats | Start |
| restaurantes | 2020-05-24T11_21_05 | 639191 | 87882 | Stats | Start |
| restaurantes | 2020-05-23T04_12_37 | 959 | 164 | Stats | Start |
| restaurantes | 2020-05-23T04_03_29 | 1150 | 0 | Stats | Start |
| avaliacoes | 2020-05-22T23_46_11 | 18970 | 1098 | Stats | Start |
| restaurantes | 2020-05-23T02_38_48 | 760 | 0 | Stats | Start |
| avaliacoes | 2020-05-22T15_39_04 | 32910 | 9537 | Stats | Start |

A listagem de tarefas em execução ou executadas exibe informações importantes para o monitoramento desses processos, entre elas é possível ver qual o Crawler (Spider) foi executado, um número identificador da tarefa, quantas páginas foram visitadas e quantos itens foram extraídos. Além de monitorar a funcionalidade, também permite a execução de ações para iniciar o processo de extração ou parar uma extração em andamento. O item *stats* permite acessar a funcionalidade que oferece um monitoramento mais detalhado de uma execução específica. A Figura 12 a seguir ilustra opções de Status de uma execução do processo de extração.

**Figura 12 - Monitoramento de performance do Crawler**

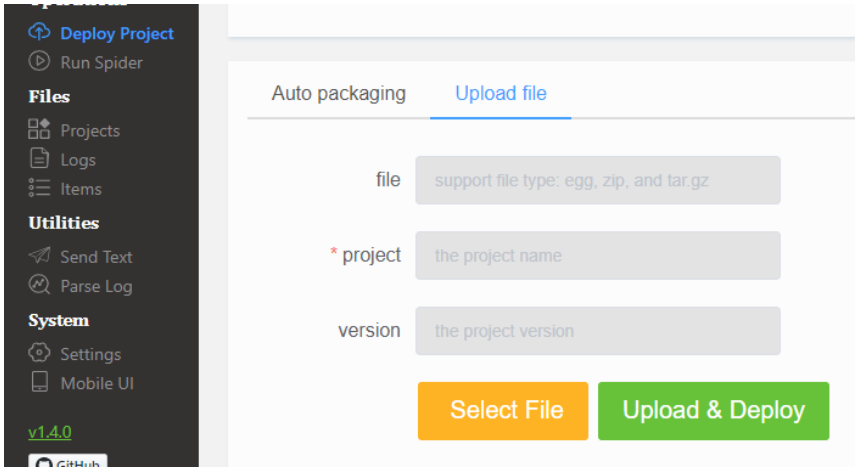



Ao abrir detalhes do *status* da execução de um dos processos de extração, o Scrapyd Web oferece um monitoramento mais detalhado, conforme é apresentado na Figura 12, que mostra o gráfico de performance de um Crawler. Com isso é possível acompanhar como um Crawler se comporta ou se comportou durante o tempo de execução, proporcionando parâmetros para que sejam feitos ajustes dos Crawlers. Além de dados relacionados a performance, o status de execução também dá acesso a detalhes de Logs de execução, oferecendo inclusive erros categorizados. Permite, ainda, que durante a execução, o usuário

administrador possa verificar como o Crawler se comporta e também ver logs que ajudam no processo de correção caso algum erro seja percebido ou o Crawler deixe de funcionar.

O item 3 da Figura 10 ilustra a funcionalidade de implantação de projetos chamada de *deploy*, que é usada no processo de atualização do projeto. Por se tratar de sistemas que visitam e analisam a estrutura atual do HTML das páginas do TripAdvisor, esse tipo de tecnologia tem um ponto negativo que é a necessidade de constante atualização, ou seja, caso o site passe por atualizações o projeto Scrapy responsável pela extração de dados também deve ser atualizado. A Figura 15 a seguir ilustra outra funcionalidade de *deploy* de projetos.

**Figura 13 - Implantação e versionamento de projetos**




Para auxiliar no processo de atualização dos Crawlers, a função deploy (implantação de projetos) oferece uma interface para *upload* de projetos. Dessa forma, o sistema permite que o desenvolvimento e atualização dos Crawlers seja feito separadamente, assim, após atualizados, os Crawlers podem ser implantados novamente como parte do módulo de extração de dados. Também é possível enviar várias versões do mesmo projeto, garantindo que o usuário tenha um controle maior sobre esse processo.

O módulo de extração de dados foi criado com o objetivo de oferecer um projeto base consistente que garanta a atualização contínua dos dados da plataforma, ligado a ferramentas que facilite o processo de execução e atualização de Crawlers da plataforma. Na seção seguinte do trabalho serão apresentadas alterações feitas no Banco de dados da plataforma. Essa atualização do modelo garante que o acesso aos dados obtidos do processo de extração e de processos subsequentes de análise de sentimentos sejam mais eficientes.

## 4.2 MODELAGEM E TESTE DO BANCO DE DADOS DA PLATAFORMA

Em execuções iniciais do processo de extração, apenas contando hotéis brasileiros, é esperado que 2.985.117 avaliações sejam analisadas pela plataforma. O grande volume de

dados resultado dos processos de extração e análise de sentimentos feitos pela plataforma também geram desafios quanto ao tempo de acesso.

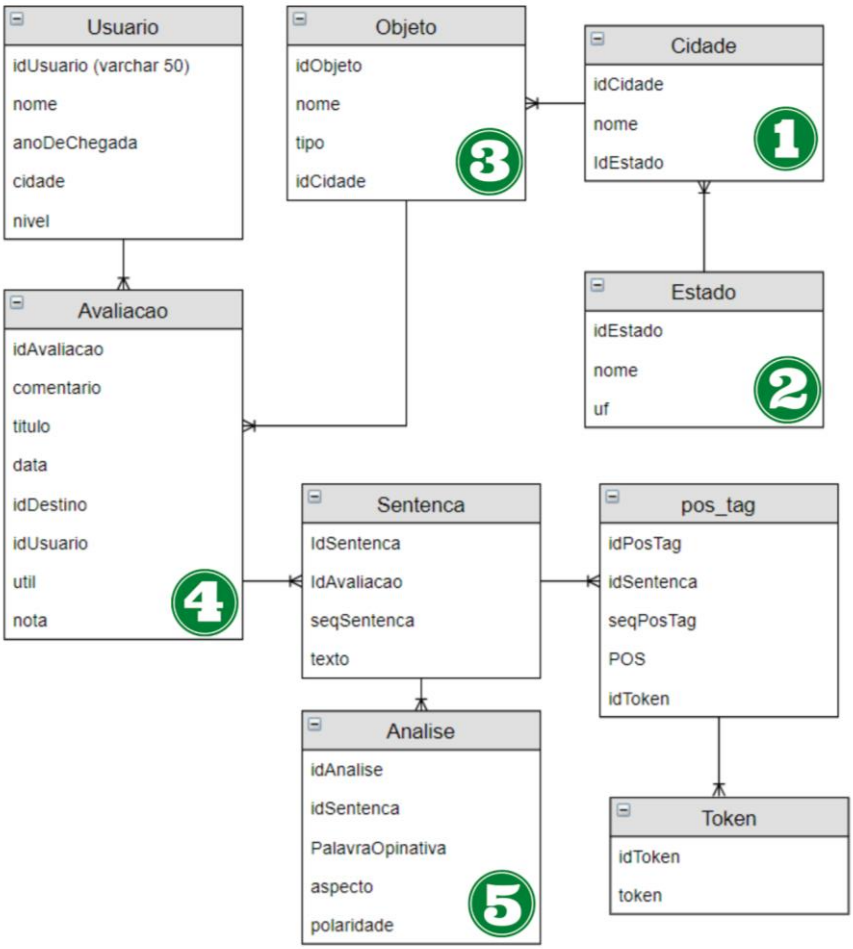
Para que a plataforma tenha um bom desempenho em etapas seguintes que envolvem o processo de visualização da informação, o banco de dados da SentimentALL passou por uma remodelagem. O objetivo desse procedimento foi adaptar o modelo de dados a arquitetura modificada da plataforma e oferecer uma estrutura mais simples e consistente. A simplificação do modelo também permite oferecer uma facilidade maior na manutenção do BD da plataforma e bons resultados quanto ao tempo de consulta aos dados. A seção seguinte detalha as principais modificações na remodelagem do banco de dados para a plataforma e como a simplificação do modelo também oferece benefícios quanto a criação de consultas para os dados.

### 4.2.1 Modelagem do banco de dados da SentimentALL

Técnicas de desnormalização foram escolhidas e aplicadas para que o modelo de dados se tornasse mais simples e eficiente. A normalização é geralmente utilizada durante o projeto conceitual de um Banco de dados com o objetivo de evitar redundâncias e gerar um modelo funcional coeso, porém nem sempre esse modelo é o mais indicado, pois a forma como os dados serão utilizados têm um peso maior quanto a eficiência do BD. A normalização é baseada na existência de dependências funcionais, porém isso não indica que esse tipo de dependências é significante em termos de uso (PURBA, 2000).

Durante a remodelagem, o primeiro passo foi analisar todo o modelo de dados da versão dois da SentimentALL. Isso foi feito com o intuito de identificar tabelas como possíveis candidatos a passarem por técnicas de desnormalização. A forma como os dados da plataforma serão consultados em processos para a visualização da informação, orientaram o processo de identificação das tabelas candidatas ao processo de desnormalização. Na Figura 14 é possível ter uma visão geral do modelo conceitual da segunda versão da SentimentALL, as marcações indicam alguns pontos identificados como candidatos ao processo de desnormalização e simplificação.

**Figura 14 - Modelo lógico do Banco de Dados da SentimentALL v2**




Fonte: Araújo (2017)

Considerando o modelo de dados da versão 2 da SentimentALL, algumas variáveis se destacaram, visto que são itens identificadores essenciais para avaliações que passam pelo processo de AS da SentimentALL e objetos avaliados. O local de um objeto avaliado, caracterizado pelas tabelas Cidade e Estado, identificadas nos itens 1 e 2 da Figura 14, são tabelas que armazenam dados essenciais para contextualização dos dados. Apesar da importância dessa informação, é possível notar que poucos dados são armazenados sobre essas entidades, de modo que possuem pouca utilidade de forma isolada, visto isso, os itens foram marcados como alvos do processo de desnormalização.

O item 3 da Figura 14 identifica a tabela Objeto, que, no modelo original, tinha poucas informações relevantes de comentários, porém durante a análise essa tabela também foi definida como essencial para contextualização de avaliações. Os itens 4 e 5 da Figura 14 identificam as tabelas Avaliação e Análise, que são os itens centralizadores de dados do processo de análise de sentimentos. Na tabela Avaliação são armazenadas informações de

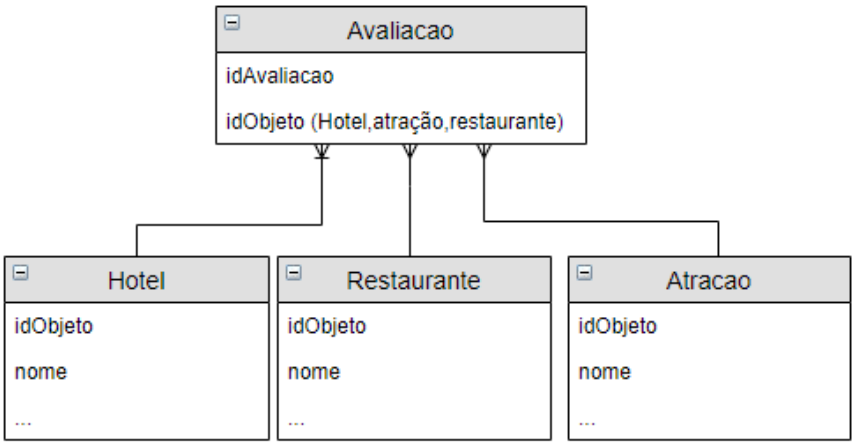
texto do comentário na sua forma bruta, e dados que ligam a avaliação ao autor e ao objeto/destino avaliado.

A análise de sentimentos realizada pela SentimentALL gera as informações mantidas na tabela Análise item 5 da Figura 14. Essa tabela identifica qual sentença avaliativa do texto comentário da avaliação passou pelo processo de análise, qual o aspecto avaliado foi identificado, a palavra opinativa expressada pelo autor e qual a polaridade, identificando se o autor avaliou o aspecto de forma positiva ou negativa. Essa tabela também foi identificada como candidata ao processo de desnormalização e simplificação do modelo. Após identificar todos os possíveis pontos a serem melhorados, o passo seguinte envolveu a aplicação de técnicas de desnormalização.

### 4.2.2 Desnormalização e definição do novo modelo

Os pontos avaliados durante a análise do modelo original foram alvo de processos para a simplificação do modelo de dados da SentimentALL. O processo de desnormalização utilizado garantiu que a reformulação fosse feita sem a perda de dados. As alterações realizadas proporcionaram um desempenho superior quanto ao tempo de consultas. Testes comparativos entre o modelo original e o novo modelo proposto serão apresentados em detalhes na seção 4.2.4 do trabalho. A Figura 15 mostra a reformulação da tabela Objeto presente no modelo original.

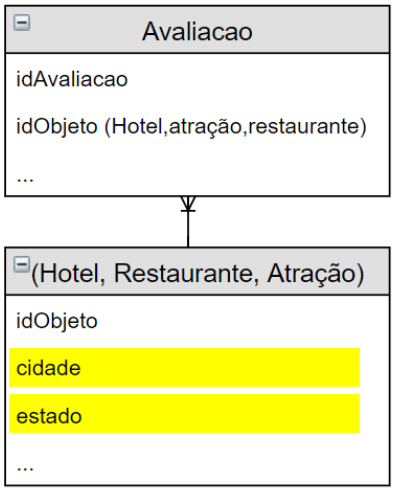
**Figura 15 - Desnormalização tabela Objeto**




A Figura 15 ilustra a criação de três novas tabelas, cada tabela representa um tipo específico de objeto avaliado. Na Figura 14 exibida anteriormente é possível notar que os objetos avaliados estão todos aglomerados na tabela Objeto, esta que possui o atributo Tipo, que até então era usado para identificar se o objeto avaliado era uma atração, hotel ou restaurante. No novo modelo cada tipo de dado recebe uma tabela específica, assim, cada tabela criada se torna mais significativa quanto ao seu contexto.

A técnica de desnormalização utilizada é chamada de “Fragmentação de Tabelas”, de forma sucinta essa técnica é usada para separar uma entidade em grupos menores e mais significativos. Apesar de aumentar a quantidade de tabelas no banco, esse tipo de aplicação reduz o número de buscas em consultas, isso acontece porque o número de linhas avaliadas durante uma consulta é diretamente proporcional ao número de linhas de valores armazenados nas tabelas utilizadas. Assim, fragmentar a tabela Objeto proporcionou uma diminuição considerável no número de linhas avaliadas, levando em conta que geralmente as buscas realizadas são sempre destinadas a um tipo de objeto avaliado. A Figura 16 a seguir ilustra a alteração feita para as tabelas de Cidade e Estado, representados nos itens 1 e 2 da Figura 14.

**Figura 16 - Desnormalização de tabelas Cidade e Estado**



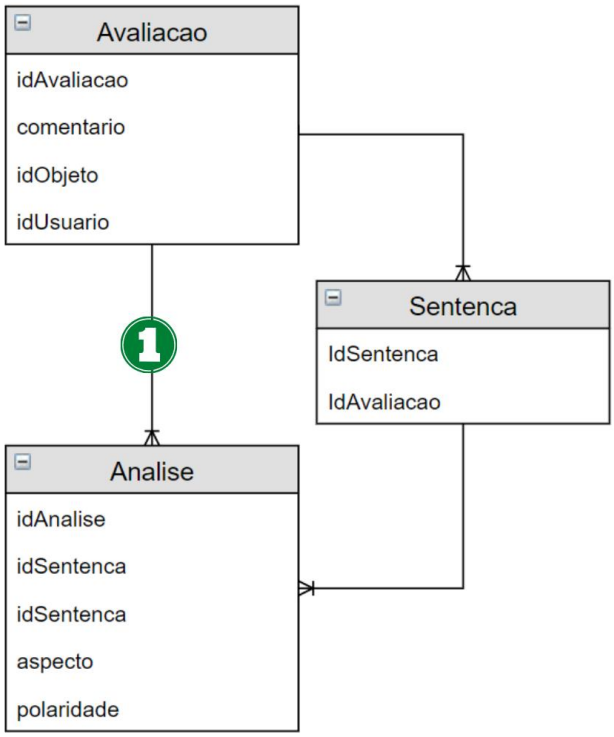

A Figura 16 mostra a tabela de um objeto avaliado que está dividida entre as tabelas de Hotel, Restaurante e Atracao. Em destaque de amarelo dois atributos foram adicionados a essas tabelas, que são referentes a informação do nome da cidade e sigla do estado, que antes estavam armazenados em duas outras tabelas. A técnica de desnormalização utilizada nesse processo é chamada de Pré-junção de tabelas, e tem o objetivo realizar a junção natural existente entre as informações das tabelas alvo e ter como resultado uma única tabela que mantém todos os registros antes separados em entidades diferentes do BD.

Essa alteração foi feita, pois durante a análise do modelo original, percebeu-se que as informações do nome da cidade e do estado só eram relevante quando utilizadas em junção com as tabelas de objetos avaliados, por esse motivo a pré-junção é uma forma de simplificar o modelo e ainda assim manter essas informações que são importantes pro contexto.

Um dos efeitos negativos disso é a geração de redundância dos dados, por exemplo, a informação do atributo cidade “Palmas”, estará presente em cada objeto avaliado pertencente a essa cidade. Contudo, o impacto negativo é irrelevante, principalmente levando

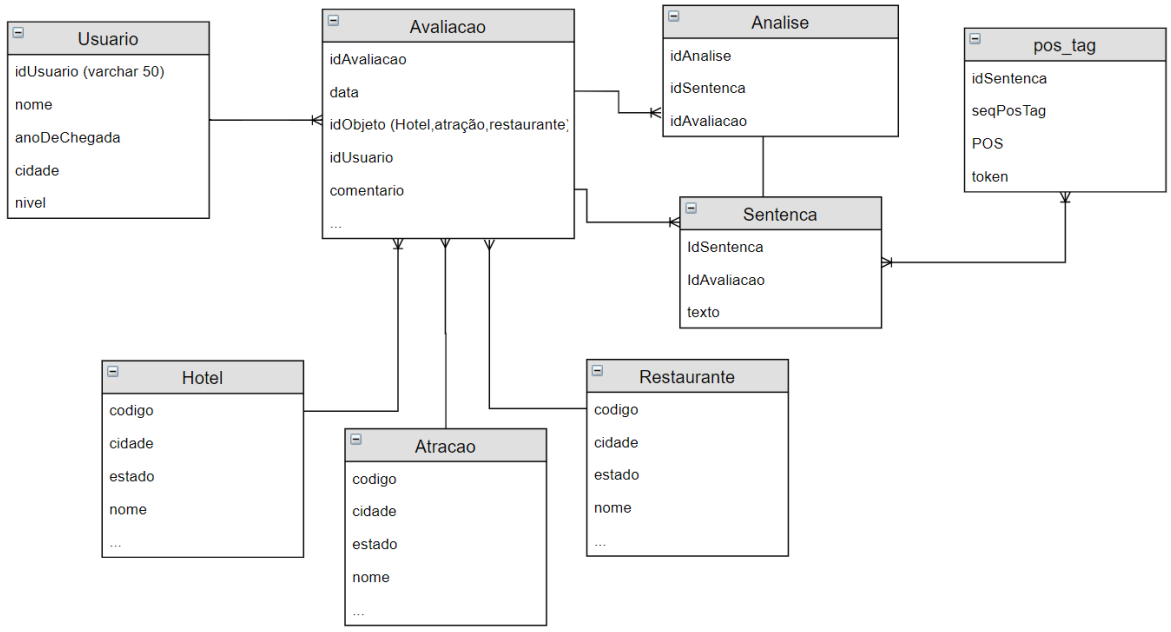
em conta o impacto positivo de simplificação  do modelo de dados e a melhora na eficiência ao consultar informações. A Figura 17 a seguir exibe a modificação do modelo direcionado a organização das informações de análise de sentimentos e sua ligação com as avaliações.

**Figura 17 - Criação de relacionamento direto**




Na Figura 17.  o item 1 ilustra um novo relacionamento criado. No modelo original do BD, a avaliação e a informação de análises feitas sobre essa avaliação estavam conectadas pela tabela intermediária Sentença, que guardava os dados de sentenças do corpus da avaliação. A utilização de uma tabela intermediária para conexão de avaliação e análises de uma avaliação gerava junções que nem sempre eram necessárias. Por isso, a criação da relação direta entre as entidades Avaliacao e Analise permitiu o uso de consultas mais diretas e com menos junções. Por fim, a figura 18 exibe uma visão geral do novo modelo resultados dos processos apresentados, algumas tabelas foram simplificadas nessa versão, o modelo completo está disponível nos apêndices do trabalho.

**Figura 18 - Modelo de dados simplificado da Plataforma SentimentALL**




O modelo resultado do processo de remodelagem da base de dados apresenta como modificações a exclusão das tabelas Cidade e Estado apresentado na Figura 14, que passaram a ser atributos nas tabelas de objetos avaliados, ilustrados pela Figura 16. Os dados de objetos avaliados também foram alterados e passaram pelo processo de fragmentação, gerando as tabelas Hotel, Atracao e Restaurante, e a nova ligação direta entre avaliação e análises dessa avaliação. A seção seguinte discorre sobre a criação de índices no modelo físico construído a partir do modelo conceitual apresentado na Figura 18. A criação de índices é utilizada como método para melhoria no tempo de consultas.

### 4.2.3 Criação de índices

A criação de índices é um dos procedimentos mais comuns para se obter uma melhor performance de um modelo de dados. Os bancos de dados contam com a indexação de estruturas de dados para executar com eficiência buscas diversas (BEUTEL, et. al, 2007). O objetivo principal dessa técnica é criar estruturas adjacentes as tabelas do banco para armazenar caminhos mais curtos para dados, com isso é possível recuperar e ordenar valores no BD de forma mais eficiente, eliminando a necessidade de, por exemplo, visitar todas as linhas de uma tabela durante uma busca. No processo de definição do novo modelo para o BD da SentimentALL, índices foram criados em duas tabelas.

A primeira tabela a receber essa melhoria foi a tabela Analise, que mantém registro de Aspectos avaliados, palavras opinativas e a polaridade definida durante a análise de sentimentos da SentimentALL. O índice criado para essa tabela é destinado a indexação de valores em conjunto de aspectos e polaridades. A decisão de criar um índice em conjunto para esses atributos foi feita pois as consultas nessa tabela geralmente buscam por esses

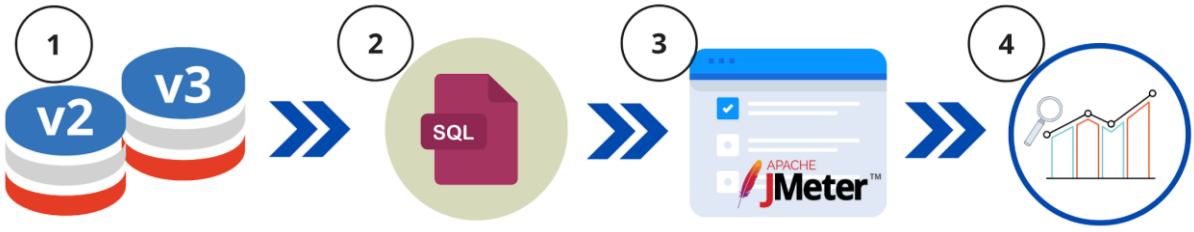
dados. A indexação também permite que a ordenação de buscas por aspectos avaliados seja mais rápida.

O segundo índice criado foi destinado a tabela Avaliacao. Esse índice foi usado para indexação de valores de identificação de objetos avaliados. O ID do objeto avaliado é um índice eficiente pois durante a recuperação de informações sobre avaliações obtidas pela SentimentALL a junção com objeto avaliado depende dessa chave estrangeira. Essa ligação é responsável por definir todo o contexto para aquele comentário analisado. Após a construção desses índices foram iniciadas as etapas para a aplicação de testes que possam comprovar a efetividade das modificações feitas entre as versões do BD da SentimentALL.

### 4.2.4 Testes de performance nos Bancos de Dados

Para comprovar a eficiência do novo modelo, após os processos de desnormalização de tabelas e criação de índices, o banco de dados passou por testes. Os testes foram usados para validar a eficiência do Banco de dados remodelado, principalmente em relação às consultas. O teste de desempenho é um tipo de teste que visa determinar a capacidade de resposta, a confiabilidade, o *throughput*, a interoperabilidade e a escalabilidade de um sistema e/ou aplicação operando com uma carga de trabalho específica (ERINLE, 2017, pag. 243). As etapas para os testes de performance dos BDs foram: preparação do ambiente de teste, planejamento de consultas teste, preparação do teste com JMeter e análise dos resultados. A Figura 19 ilustra as quatro etapas do processo de desenvolvimento e aplicação dos testes de performance, em seguida cada etapa será descrita em detalhes.

**Figura 19 - Metodologia dos testes de performance**




O item 1 da Figura 19 ilustra a etapa de preparação do ambiente onde os testes foram executados. Primeiro os dados presentes no banco de dados da versão 2 da SentimentALL foram replicados para o novo modelo de dados definido e construído para armazenar os dados da plataforma SentimentALL. Utilizar os mesmos dados para as duas versões foi importante, visto que o objetivo dos testes é comparar a performance dos dois modelos em ambientes e condições idênticas. Para essa migração de dados, um Script foi criado de forma que as conexões e informações se mantivessem, mesmo alterando o modelo dos dados.

O gerenciamento e instalação dos bancos teste foi feita pelo SGBD Microsoft SQL Server na versão 14.0.1000.169. A máquina utilizada como ambiente de testes utiliza o

sistema Operacional Windows 10 Pro 64Bits, com armazenamento tem um HDD de 1TB e 8GB de memória e rodando no processador Intel Pentium G4560 de 3.5Ghz. Os testes foram feitos a partir de instalações locais dos BDs e da ferramenta JMeter. A execução local dos testes assegura que o resultado não foi influenciado por variáveis externas como a disponibilidade da rede. O ambiente descrito foi mantido exatamente igual para os testes realizados em ambas versões do BD da SentimentALL.

O item 2 da Figura 19 representa a etapa de planejamento das consultas usadas como parâmetro para execução de testes de performance dos modelos. Cinco consultas foram utilizadas para testar o tempo de resposta dos modelos de dados. Para cada exemplo de consulta, duas instruções SQL (do inglês *Structured Query Language*) diferentes foram criadas, uma para cada versão do BD. A Tabela 5 a seguir descreve as consultas usadas na etapa de teste de performance e uma amostra de resultados obtidos por essas consultas.

**Tabela 5 - Consultas para teste de performance dos BDs**

<table>
<tr><th>ID</th><th>Descrição da Consulta</th><th>Ex. Resultado da SQL</th></tr>
<tr><td>1</td><td>Quantidade de avaliações por aspectos, direcionadas a hotéis da cidade de Caldas Novas, nas polaridades Positivo e Negativo</td><td>
<table>
<tr><th>Cidade</th><th>Aspecto</th><th>Pol.</th><th>Qtd</th></tr>
<tr><td>Caldas Novas</td><td>recepção</td><td>-1</td><td>32</td></tr>
<tr><td>Caldas Novas</td><td>infra-estrutura</td><td>1</td><td>33</td></tr>
</table>
</td></tr>
<tr><td>2</td><td>Quantidade de aspectos avaliados nas polaridades Positivo e Negativo para hotéis da cidade de Caldas Novas</td><td>
<table>
<tr><th>Polaridade</th><th>Quantidade</th></tr>
<tr><td>1</td><td>44763</td></tr>
<tr><td>-1</td><td>6913</td></tr>
</table>
</td></tr>
<tr><td>3</td><td>Quantidade de aspectos avaliados nas polaridades Positivo e Negativo por hotel, na cidade de Caldas Novas</td><td>
<table>
<tr><th>Hotel</th><th>Polaridade</th><th>Quantidade</th></tr>
<tr><td>d10350109</td><td>-1</td><td>4</td></tr>
<tr><td>d10350109</td><td>1</td><td>53</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>Texto de sentença avaliativa direcionada a hotéis da cidade de Caldas novas que contém o aspecto cama avaliado positivamente</td><td>
<table>
<tr><th>Texto da sentença avaliativa</th></tr>
<tr><td>...limpeza impecável , e com certeza a melhor cama que já dormi !</td></tr>
<tr><td>.. cama muito boa , café da manhã excelente , elevadores não consegue atender a demanda .</td></tr>
</table>
</td></tr>
<tr><td>5</td><td>Quantidade de usuários de nível 1 que deram nota 1 para hotéis da cidade de Caldas Novas</td><td>
<table>
<tr><th>Total de usuários</th></tr>
<tr><td>195</td></tr>
</table>
</td></tr>
</table>

A Tabela 5 descreve as cinco consultas que foram aplicadas em ambas as versões dos bancos de dados de teste, as instruções são apresentadas de forma completa nos apêndices deste trabalho. A criação dessas instruções SQL teve a finalidade de simular o processo de recuperação de dados feito pela plataforma SentimentALL, consequentemente permitindo

um comparativo de performance dos modelos testados em condições semelhantes as utilizadas pelo módulo de visualização da informação que será descrito em seções subsequentes. Cada teste descrito representa duas consultas SQL a serem executadas nos BDs nas versões 2 e na versão da plataforma SentimentALL. A criação dessas duas instruções SQL distintas para cada BD foi necessária visto que apesar de semelhante os modelos dos dados mudaram. A Figura 20 apresenta duas consultas criadas para execução dos testes.

**Figura 20 - Instruções SQL usadas nos testes**

**Consultas #1**

**SentimentALLv2**

```
SELECT o.idCidade, a.aspecto, a.polaridade, COUNT(*) AS Total
FROM SentimentALLv2.dbo.análise as a
INNER JOIN SentimentALLv2.dbo.sentença as s on a.idSentença=s.idSentença
INNER JOIN SentimentALLv2.dbo.avaliação as av on av.idAvaliação=s.idAvaliação
INNER JOIN SentimentALLv2.dbo.objeto as o on o.idObjeto =  av.idDestino
WHERE o.idCidade = 'g1012170' AND o.tipo = 'Hotel'
GROUP BY o.idCidade, a.aspecto, a.polaridade
```

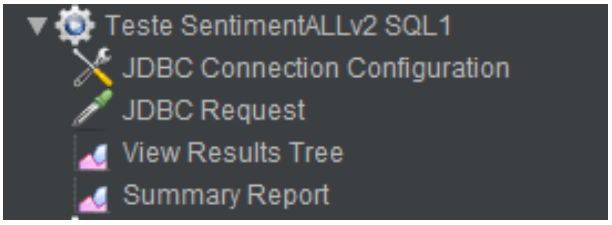
**SentimentALLv3**

```
SELECT av.cidadeObjeto, a.aspecto, a.polaridade, COUNT(*) AS Total
FROM SentimentALLTeste.dbo.analise as a
INNER JOIN SentimentALLTeste.dbo.avaliacao as av on av.idAvaliacao = a.idAvaliacao
INNER JOIN SentimentALLTeste.dbo.hotel as h on h.codigo = av.idObjeto
WHERE h.cidade = 'Caldas Novas'
GROUP BY av.cidadeObjeto, a.aspecto, a.polaridade
```

A Figura 20 apresenta SQLs criadas para a SentimentALLv2 e SentimetALLv3 (A versão 3 define o modelo de dados da plataforma SentimentALL), o caso de teste ilustrado é o caso de teste número 1 descrito na Tabela 5. As alterações no modelo de dados fizeram com que o acesso aos dados ficasse diferente.  Essa diferença pode ser notada na quantidade de junções necessárias para consultar os mesmos dados, por exemplo,  no BD após a remodelagem a consulta SQL necessita de uma junção a menos, em comparação a versão 2. Essa simplificação também é um ponto positivo do processo de remodelagem do modelo de dados para a plataforma. O objetivo dessa e das outras consultas é comparar o tempo de resposta das duas versões ao buscar os mesmos dados.

Na sequência da metodologia utilizada para testes dos BDs, O item 3 da Figura 19 representa o processo de desenvolvimento dos testes utilizando a ferramenta JMeter. Essa ferramenta foi usada por possibilitar a execução consecutiva de várias requisições de acessos utilizando uma mesma instrução SQL, assim, a cada requisição o tempo de resposta é monitorado e armazenado. Ao executar várias vezes a mesma instrução, o resultado oferece uma média de tempos de resposta, garantindo um resultado mais preciso. A Figura 21 ilustra os componentes utilizados para a aplicação de testes utilizando o JMeter.

**Figura 21 - Testes desenvolvidos com a ferramenta JMeter**




No desenvolvimento dos testes utilizando essa ferramenta JMeter 5, alguns componentes foram utilizados. O primeiro componente é chamado de Thread Group (Grupo de Thread), na imagem exemplo o componente recebe o nome de Teste SentimentALLv2 SQL1, e é responsável pelas configurações gerais para a execução de um teste. Dentre as configurações definidas neste componente, as principais foram a quantidade de usuários simultâneos que serão usados para simular acessos ao BD e quantos acessos cada usuário deve fazer.

Os testes foram realizados utilizando a simulação com 1 usuário simultâneo e, para cada execução, 10 requisições foram realizadas. Os testes receberam essa configuração pois o objetivo do teste foi obter o tempo de resposta de cada requisição, procurando obter informações sobre a performance no tempo de execução dessas consultas, diferente de outros testes disponibilizados pela ferramenta direcionado a avaliação de BDs e sistemas em momentos de estresse simulando muitos acessos simultâneos.

O segundo componente é o "JDBC Connection Configuration", que é um componente destinado a configuração da conexão da ferramenta com o banco de dados testado. Essa conexão é feita através de *drivers* em Java oferecendo um método para realização de uma consulta direta ao Banco de Dados testado. O Terceiro componente é o "JDBC Request", em que é definido qual o tipo de consulta será feito e qual a instrução usada para realização de um teste. O tipo de instrução utilizada para esse teste de performance foi o *select* e as instruções são referentes às consultas SQL criadas a partir dos testes descritos e apresentados na Tabela 5.

Os componentes "View Results Tree" e "Summary Report" são chamados de Listeners e têm como objetivo coletar dados e gerar relatórios das execuções dos testes. Os dados coletados são, por exemplo, o momento em que o teste foi executado, o tempo em milissegundos decorrido durante cada requisição de uma consulta, o código de resposta do servidor e a quantidade de *bytes* de resposta. Além de apresentar o relatório em tempo real durante a execução, esses componentes também permitiram salvar o resultado de cada teste em um arquivo de texto no formato CSV, facilitando as etapas seguintes de análise.

Após a criação dos testes com a ferramenta JMeter, a execução foi feita. Durante esse processo foi percebido que um dos mecanismos do SGBD SQL Server funciona de modo

que, após a primeira vez que os dados são consultados, as consultas subsequentes terão uma diminuição considerável no tempo de resposta. Em virtude disso, foi necessário que, para cada execução do teste, a máquina usada fosse totalmente reiniciada, de modo a garantir que nenhum dado fosse previamente carregado na memória. Isso evitou que dados tivessem pré-carregados, garantindo a integridade dos testes. Cinco testes descritos foram executados para ambas versões, gerando dez testes e um total de cem requisições aos Banco de Dados testados. Para cada teste, a ferramenta JMeter emitiu um relatório com dados do procedimento. A Tabela 6 exemplifica o relatório resultado da execução da SQL instrução para o teste número 1.

**Tabela 6 - Relatório resultado de testes com JMeter**

| timeStamp | elapsed | dataType | bytes | Latency | Connect |
|---|---|---|---|---|---|
| **1,59027E+12** | 164458 | text | 97803 | 164334 | 652 |
| **1,59027E+12** | 4839 | text | 97803 | 4724 | 0 |
| **1,59027E+12** | 4817 | text | 97803 | 4728 | 0 |
| **1,59027E+12** | 4717 | text | 97803 | 4669 | 0 |
| **1,59027E+12** | 4716 | text | 97803 | 4672 | 0 |
| **1,59027E+12** | 4668 | text | 97803 | 4621 | 0 |
| **1,59027E+12** | 4713 | text | 97803 | 4671 | 0 |
| **1,59027E+12** | 4707 | text | 97803 | 4663 | 0 |
| **1,59027E+12** | 4732 | text | 97803 | 4689 | 0 |
| **1,59027E+12** | 4683 | text | 97803 | 4638 | 0 |

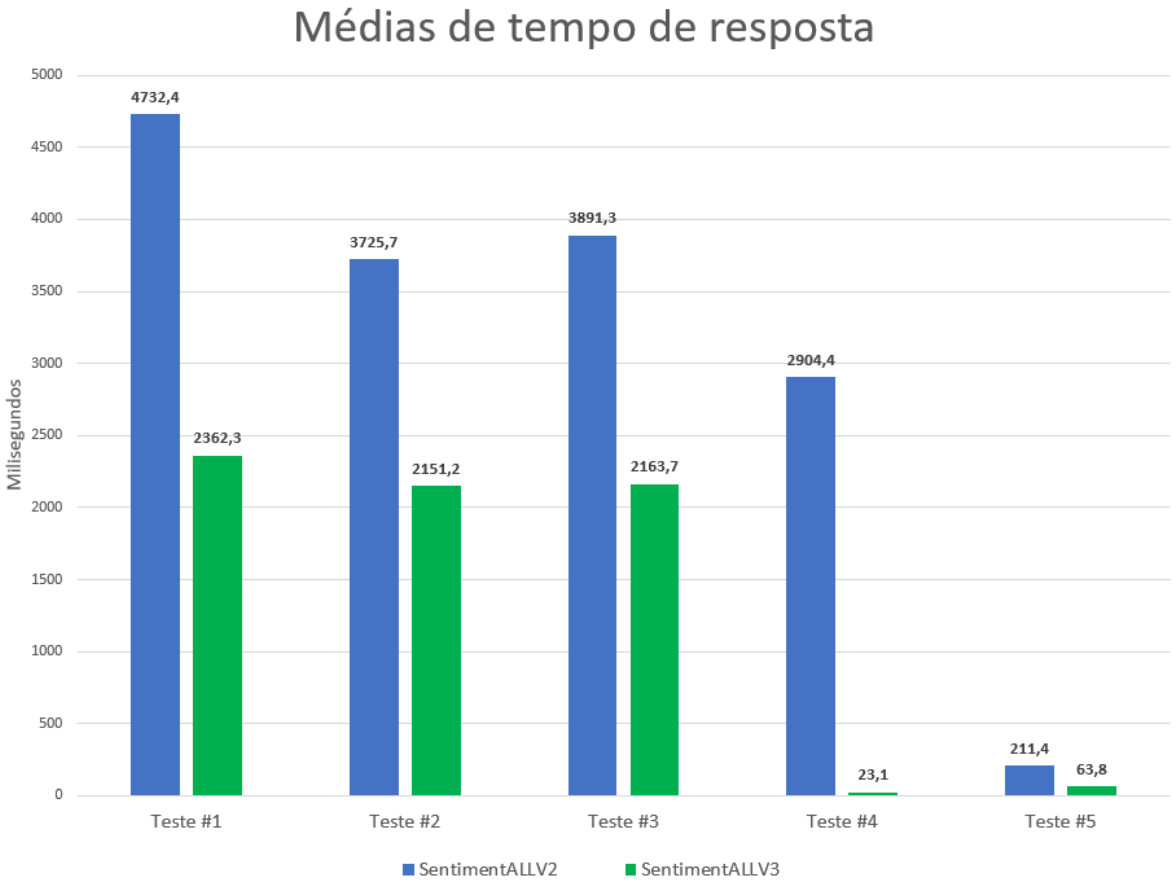
A Tabela 6 apresenta o relatório resultado da execução de testes no JMeter de forma abreviada, isso porque além das informações presentes na tabela, outros dados, como códigos de resposta, mensagens de resposta. entre outros. também estavam disponíveis. Para os testes de performance, a informação *elapsed* (Tempo decorrido) foi a principal variável para comparação dos resultados entre a versões do BD. O JMeter mede o tempo que o Banco de Dados demora para realizar a consulta e armazena essa informação para cada uma das execuções. As tabelas completas dos relatórios resultados dos testes estão disponíveis nos itens de apêndice deste trabalho.

No relatório apresentado também é possível notar que a primeira execução para a consulta teste tem um tempo consideravelmente maior se considerado a outros tempos pertencentes a mesma consulta, isso é resultado do processo de armazenamento de dados em *buffer* citado anteriormente. Por ter esse comportamento nativo, os resultados do relatório consideram o tempo de armazenamento no *buffer*, porém para que os resultados tenham

dados mais uniformes, o tempo de execução do primeiro *loop* de teste foi desconsiderado no processo de análise dos testes.

Por fim, os resultados de cada teste foram analisados e comparados. Para melhor ilustrar os resultados desse processo a Figura 22 exibe o gráfico de performance das duas versões do BD quanto a média de tempo de resposta das instruções SQL utilizadas.

**Figura 22 - Resultado dos testes de Performance do BD**




Ao analisar o gráfico da Figura 22, é possível perceber que os resultados dos testes mostram uma redução considerável no tempo médio de resposta, assim como a melhora de performance é consistente e ocorre em todos os testes aplicados para o modelo do Banco de Dados na SentimentALL versão 3 (Após a reformulação) e na sua versão 2. No teste onde a discrepância de tempo foi maior, a instrução SQL do Teste #4 teve resultado favorável ao novo modelo com uma redução de 99,2% do tempo de resposta em relação a mesma consulta executada para a versão 2.

A média de tempo no Teste #2 apresentou o menor percentual de melhora entre os modelos, ainda assim, a versão 3 reduziu o tempo de consulta aos dados em 42,3%. Considerando todos os testes, em média o tempo de resposta da versão 3 em relação a versão 2 do BD apresenta uma melhora de 56,3%. Por utilizar consultas complexas e dados reais

durante o processo de teste, os resultados demonstram efetivamente que as técnicas de desnormalização e a criação de índices tiveram um efeito positivo quanto a eficiência na consulta de dados da SentimentALL. Após a comprovação da melhora no modelo de dados através dos testes de performance, um algoritmo para normalização de palavras foi incluído no processamento dos dados para a plataforma, essa adição ao processo será apresentada na próxima seção do trabalho.

## 4.3 NORMALIZAÇÃO DA FORMA DE PALAVRAS

Dentre as melhorias propostas nessa versão da SentimentALL, uma delas é a introdução de um algoritmo para a normalização da forma de palavras. A técnica escolhida para esse procedimento foi a lematização. O objetivo primário para a aplicação desse tipo de algoritmo é gerar um termo único para palavras com significados idênticos, assim, o efeito esperado é que em etapas de recuperação e visualização de dados, várias flexões de uma palavra sejam recuperadas como similares e contabilizadas como um termo único. O algoritmo foi aplicado para a normalização de palavras opinativas e aspectos avaliados.

O algoritmo da biblioteca *open-source* Spacy foi utilizado na realização dessa tarefa. Essa biblioteca Python oferece um modelo treinado e pronto para a tarefa de Lematização. Porém, alguns cuidados foram tomados quanto a aplicação desse algoritmo. Por ser uma etapa ainda pouco estudada no português brasileiro, o algoritmo de lematização apresentou dificuldade na normalização de termos. A Tabela 7 apresenta um exemplo do resultado do processo de lematização ao ser aplicado em dados da plataforma.

**Tabela 7 - Resultado da Normalização de Palavras**

| Palavra opinativa | Palavra Op. Normalizada | Aspecto | Aspecto Normalizado |
|---|---|---|---|
| limpas | limpo | acomodações | acomodação |
| excelente | estrutura | excelente | estruturar |
| acolhedor | acolhedor | ambiente | ambientar |
| gelado | gelar | apartamento | apartamento |

A normalização das palavras ambiente e estrutura, mostrou que em alguns casos o resultado da Lematização não é ideal. Para essas palavras exemplo o resultado esperado seria

manter a palavra original. Termos flexionados da palavra ambiente, por exemplo, são as palavras ambientes e ambientado, porém o real resultado foi a normalização para o verbo no infinitivo, assim, o aspecto normalizado desses termos foi definido como ambientar e estruturar. Esse padrão se manteve para outras palavras normalizadas pelo algoritmo. Visto que essas palavras são comumente utilizadas para representar aspectos do local, e que esses aspectos são geralmente representados por substantivos, o resultado pode ser confuso, pois identifica o termo normalizado como um verbo. Contudo outras palavras mostraram ter sucesso ao serem normalizadas, por exemplo acomodações e limpas que receberam como palavra normalizada os termos acomodação e limpo.

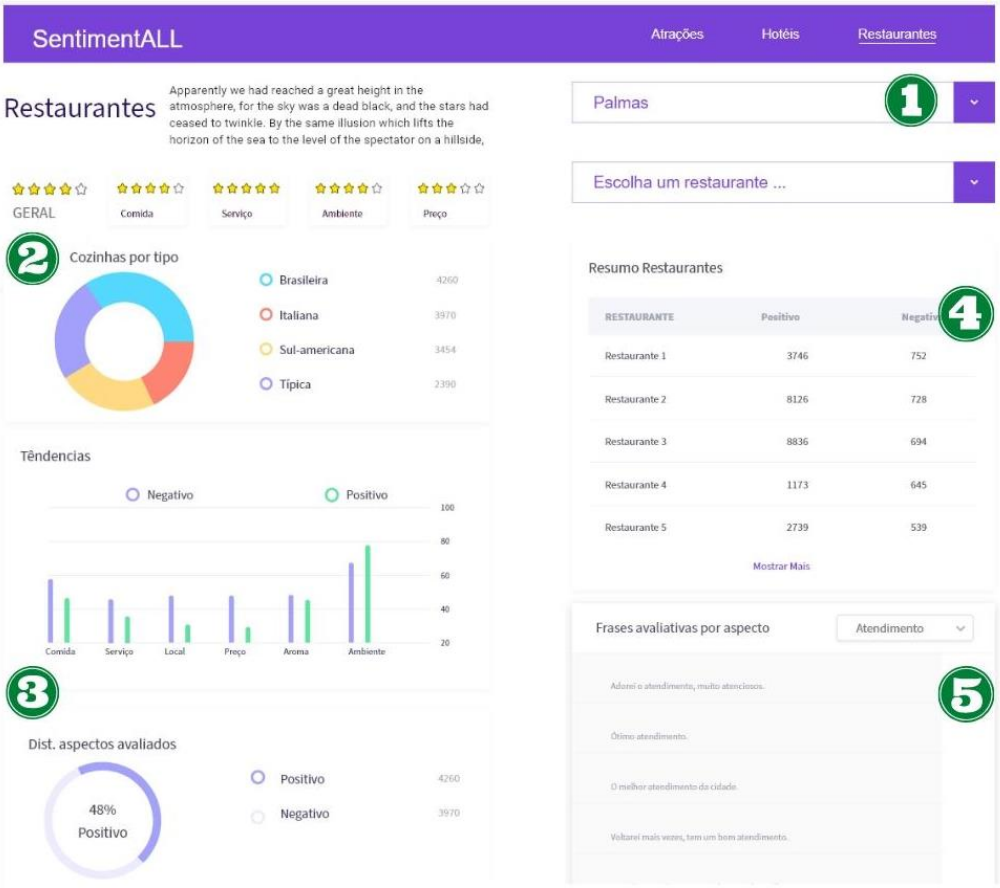
O comportamento do algoritmo também se mostrou ineficiente na normalização de termos multi-palavras. Isso não representa necessariamente um erro, porém, ao normalizar um termo constituído por várias palavras, o algoritmo da biblioteca Spacy realiza o processo em cada palavra separadamente, gerando um termo normalizado que perde a representatividade do termo original, por esse motivo o algoritmo de normalização não é utilizado para normalizar termos multi-palavras identificados pela SentimentALL. Após a inclusão do algoritmo para a normalização de palavras, foi feita a construção do protótipo para o módulo de visualização da informação, os resultados dessa etapa são apresentados na próxima seção.

## 4.4 PROTÓTIPO MÓDULO DE VISUALIZAÇÃO DA INFORMAÇÃO

Protótipos de telas do módulo de visualização de dados da plataforma SentimentALL foram criados para guiar o processo de desenvolvimento. O objetivo da prototipação é definir de forma visual quais dados serão exibidos e de que forma.

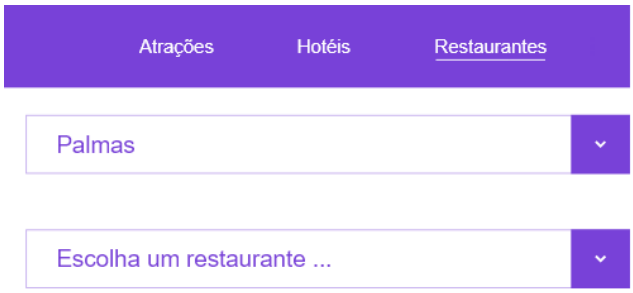
Os resultados da etapa de análise de páginas do TripAdvisor descrita na subseção 4.1.1 do trabalho definiu um novo conjunto para os objetos avaliados (hotéis, restaurantes e atrações). Para cada objeto, atributos distintos foram extraídos, e essa característica orientou o processo de desenvolvimento do protótipo, que foi dividido em três telas principais, uma para cada contexto do tipo de objeto avaliado. A Figura 23 apresenta o protótipo da tela do componente para visualização da informação de dados da SentimentALL sobre restaurantes.

**Figura 23 - Protótipo de tela de Restaurantes**




A tela protótipo do  módulo de visualização da informação para exibição de dados sobre restaurantes distribui os dados em 5 áreas principais, cada área será descrita com mais detalhes a seguir. A disposição dos elementos foi mantida como padrão para as telas protótipo de hotéis e atrações, porém cada tela é direcionada a exibir dados do seu tipo de objeto em destaque. A Figura 24 ilustra em detalhes o item 1 da Figura 23.

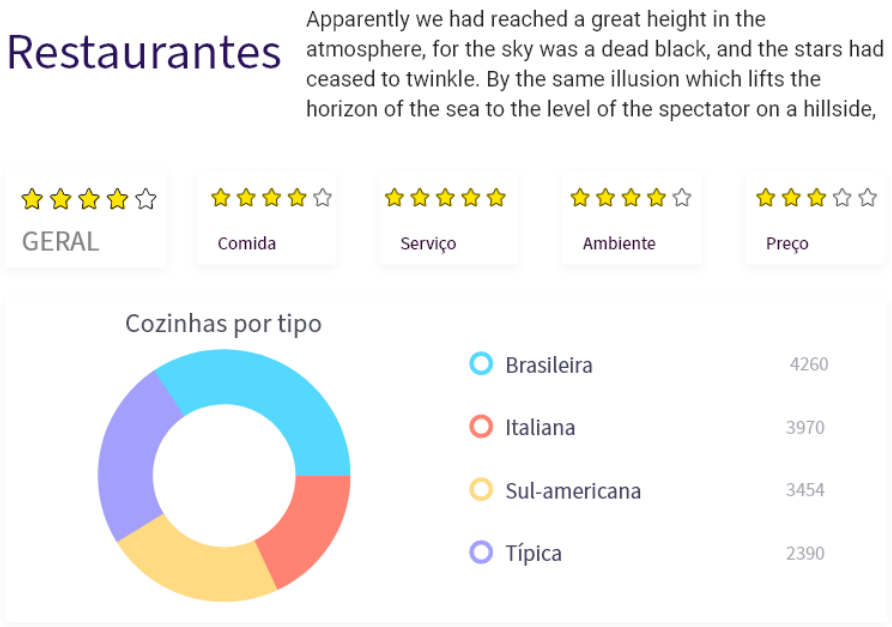
**Figura 24 - Seletor de cidade e alvo específico**




A primeira área apresentada no item 1 da Figura 23 representa parte da interface que permite interações do usuário com os dados, principalmente quanto a definição de contexto para quais informações serão apresentadas. A primeira opção de interação é através do menu superior que oferece as opções de navegação entre dados de restaurantes, hotéis e atrações.

Ao acessar avaliações de restaurantes como no exemplo, dois seletores são apresentados para o usuário, o primeiro permite que sejam visualizados dados de restaurantes de uma cidade, após a seleção de uma cidade, o seletor de hotéis da cidade será habilitado, nesse caso o usuário pode focar também em um estabelecimento em específico. Esse padrão da interface é mantido também para as telas de atrações e hotéis. A segunda área da tela do contexto restaurantes apresentada no Item 2 da Figura 23 é detalhada na Figura 25 a seguir.

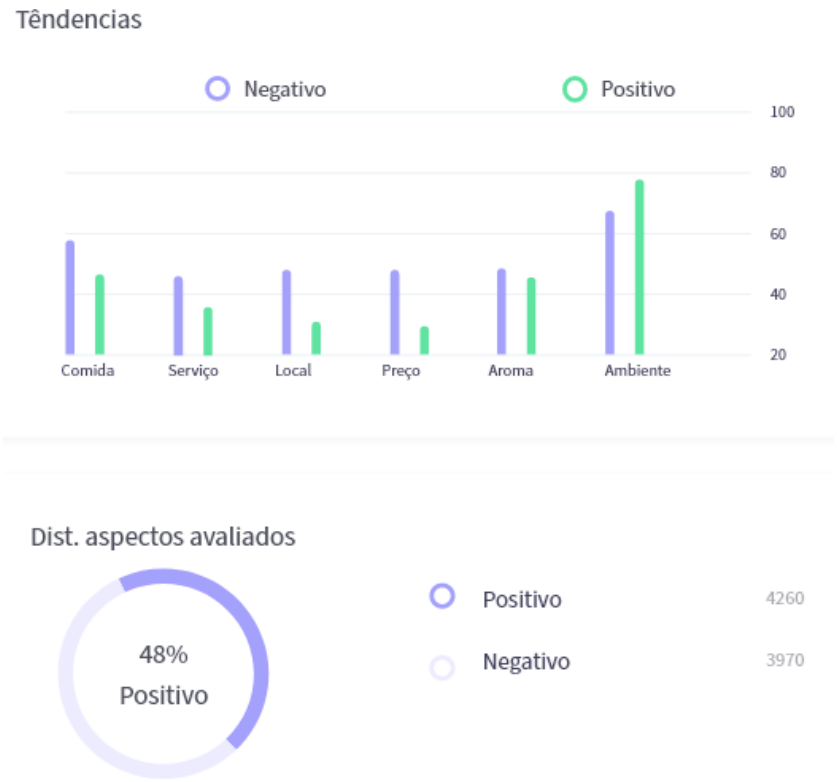
**Figura 25 - Avaliações Likert e distribuição por tipo para Restaurantes**




A Figura 25 ilustra o protótipo para dados relacionados a restaurantes avaliados. No topo da Figura 25 o protótipo apresenta um título e uma descrição breve sobre as informações nessa seção da plataforma, abaixo estão dispostas as avaliações Likert para esse contexto. Na área em destaque, essas avaliações na escala Likert disponíveis feitas por usuários do site para restaurantes são: Geral, comida, serviço, ambiente e preço. O trabalho de OLIVEIRA (2019) mostra uma forte correlação entre a nota na escala Likert dada pelo autor da avaliação no site TripAdvisor e o comentário textual. Por esse motivo, esses dados são extremamente relevantes quanto a opinião geral dos usuários sobre determinado destino.

Um gráfico do tipo *donut* ilustra a distribuição dos tipos de cozinha disponíveis, cada tipo representa a quantidade de restaurantes que servem esse tipo de comida na cidade escolhida. As informações apresentadas são agregadas e calculadas quando o alvo é apenas a cidade, ou direcionadas a apenas um hotel em específico.  A Figura 26 apresenta detalhes do item 3 da página de informações para restaurantes.

**Figura 26 - Distribuição por aspecto e polaridade**




Na Figura 26, a área de dados de restaurantes exibe um gráfico de barras com o comparativo da quantidade de avaliações positivas e negativas feitas para os principais aspectos avaliados. O gráfico de distribuição exibido abaixo representa a somatória dessas avaliações distribuídas por polaridade, possibilitando que o usuário tenha uma informação mais direta do total de aspectos avaliados de forma positiva ou negativa. Essa parte da interface também é mantida nos contextos de hotéis e atrações. A Figura 27 ilustra o item 4 do protótipo para a página de restaurantes.

**Figura 27 - Resumo de aspectos avaliados por restaurante**

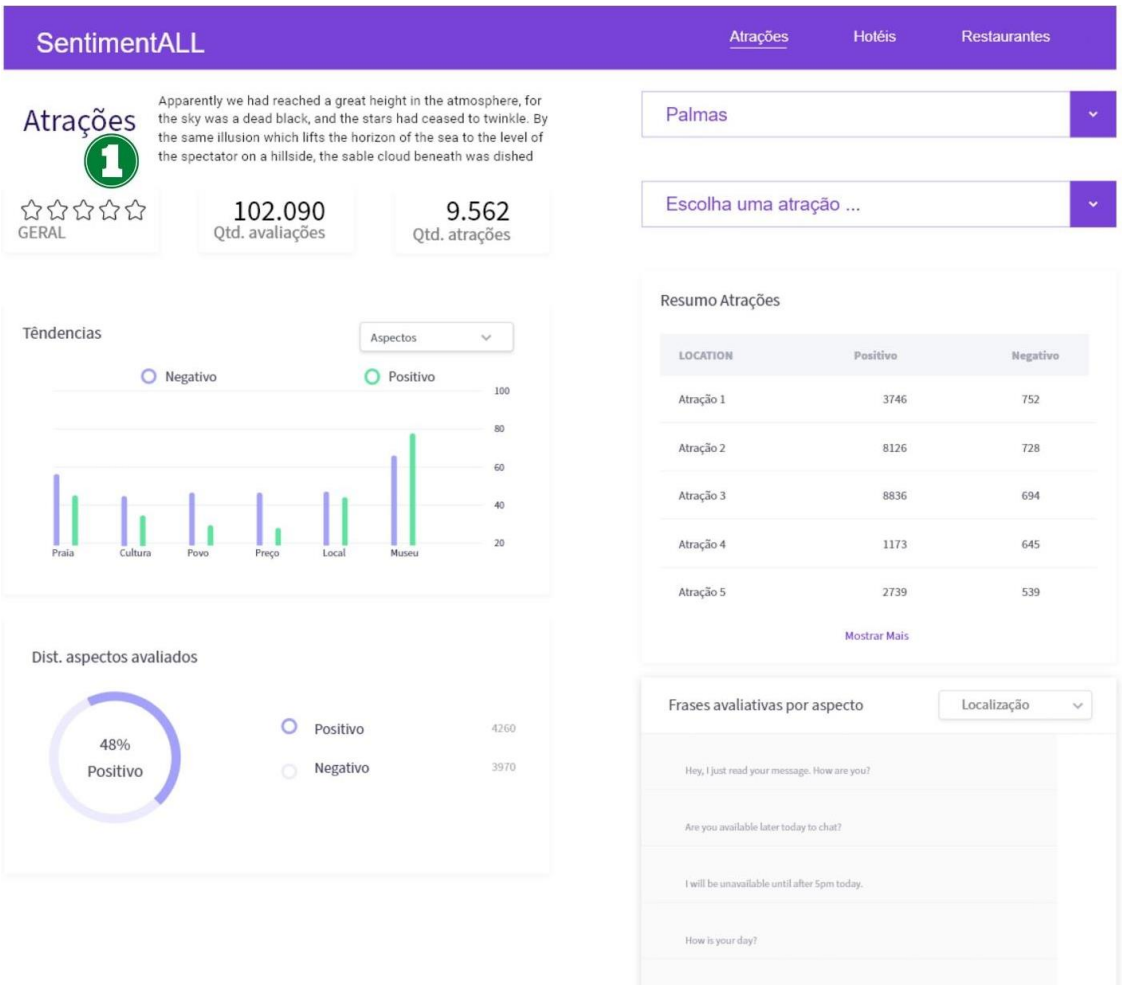

A Tabela ilustrada na Figura 27 apresenta dados detalhados com totalizadores da quantidade de aspectos avaliados de forma positiva e negativa para hotéis de uma cidade. Essa informação condensada permite que os estabelecimentos sejam comparados lado a lado de forma rápida e concisa. Os dados exibidos nessa parte da interface também serão exibidos para os contextos de atrações e hotéis avaliados. A Figura 28 apresenta o item 5 da tela para a página de restaurantes.

**Figura 28 - Frases exemplo com base no aspecto selecionado**


Além de mostrar dados condensados e de forma visual, na Figura 28 são exibidas frases avaliativas direcionadas a um aspecto que pode ser escolhido, esse tipo de interação permite que o usuário possa ver a estrutura original da frase em que o aspecto foi avaliado. O protótipo apresentado é direcionado apenas a tela de restaurantes avaliados, a estrutura foi

mantida para as páginas de Hotéis e Atrações, exceto em relação a informações específicas do tipo de objeto. A Figura 29 ilustra a tela do contexto de atrações.

**Figura 29 - Protótipo Tela de Atrações**



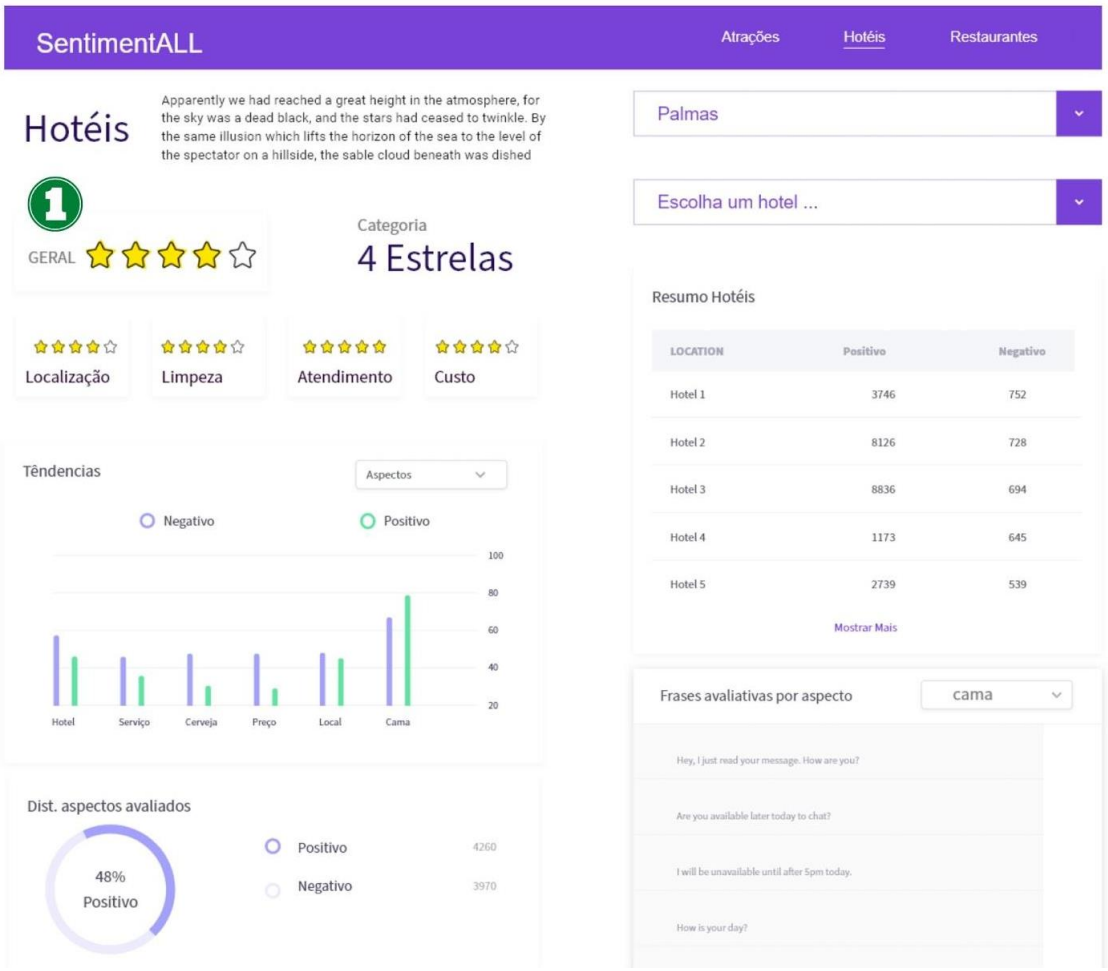

A Figura 29 apresenta a tela do contexto para dados de atrações. No TripAdvisor esse tipo de objeto avaliado oferece o menor número de dados relacionados, por esse motivo essa área tem um conjunto mais simples de dados. Como mencionado anteriormente, alguns dos itens de exibição foram mantidos com exceção do item 1 em destaque na Figura 30.

**Figura 30 - Totalizadores de atrações**


A Figura 30 ilustra parte de dados específicos para o contexto de atrações. Nessa área da interface serão exibidos a nota Likert geral que representa a nota média de uma cidade ou a nota específica de uma atração selecionada. A interface também exibe um texto introdutório

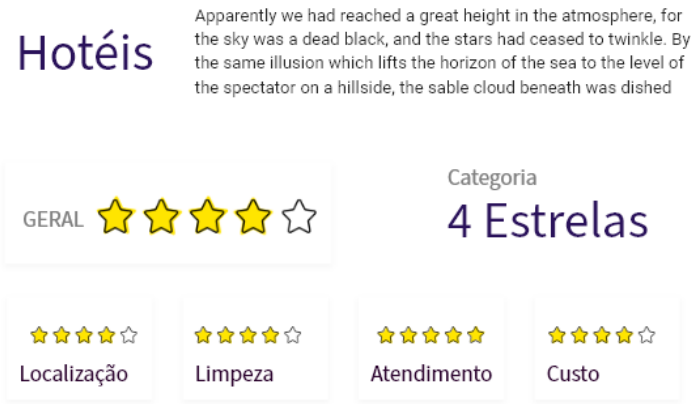
sobre o contexto dos dados e totalizadores que informam a quantidade total de avaliações para o contexto escolhido e no nível de cidade a interface também exibe o número total de atrações disponíveis para o destino. A Figura 31 a seguir exibe a tela de dados para Hotéis avaliados.

**Figura 31 - Protótipo Tela de Hotéis**




Conforme apresentado na Figura 31, a interface para dados de hotéis mantém o padrão em relação aos outros contextos, exibindo dados sobre aspectos avaliados, distribuição por polaridade e resumo de hotéis e frases avaliativas, assim como os menus e áreas de interação do usuário. O item 1 da Figura indica alguns dados que são específicos para o contexto de hotéis, essa parte da interface é detalhada na Figura 32 abaixo.

**Figura 32 - Categorias e totalizadores de hotéis**



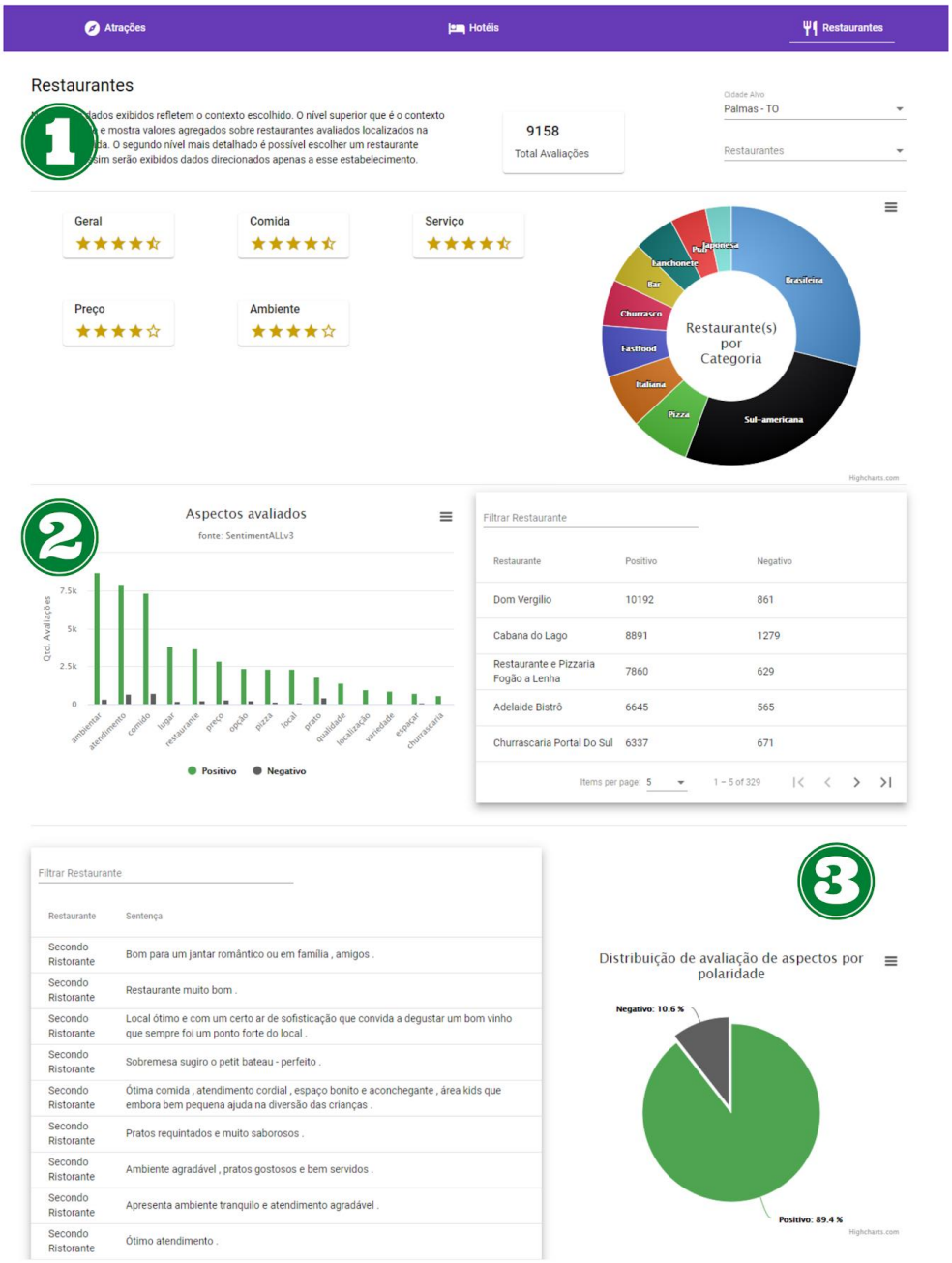

Para hotéis, os usuários do TripAdvisor podem avaliar em pontuações da escala Likert alguns itens específicos que são: Geral, Localização, Limpeza, Atendimento e Custo. No item da tela de Hotéis ilustrado em detalhes na Figura 32, os votos dos usuários são exibidos de forma visual. Além dessas informações para votos na escala Likert, essa área da interface também apresenta a categoria predominante de hotéis de uma cidade, ou a categoria de um hotel em específico. Na tela de Hotéis também é exibido um texto introdutório sobre os dados apresentados. Essas três telas protótipos compõem o módulo de visualização da informação. A seção seguinte descreve o desenvolvimento do módulo de visualização da informação com base nos protótipos apresentados.

## 4.4 MÓDULO DE VISUALIZAÇÃO DA INFORMAÇÃO

O módulo de visualização da informação tem o objetivo de exibir de forma resumida e clara dados obtidos do processo de análise de sentimento feito pela SentimentALL. O foco dessa versão para a visualização da informação é apresentar dados relacionados aos objetos avaliados e aspectos avaliados de atrações, hotéis e restaurantes brasileiros. Busca-se, por meio da apresentação de dados de restaurantes, atrações ou hotéis, oferecer um contexto mais abrangente para o entendimento de gráficos relacionados aos principais aspectos avaliados para esses objetos, oferecendo também uma percepção mais clara de como esses aspectos são relevantes.

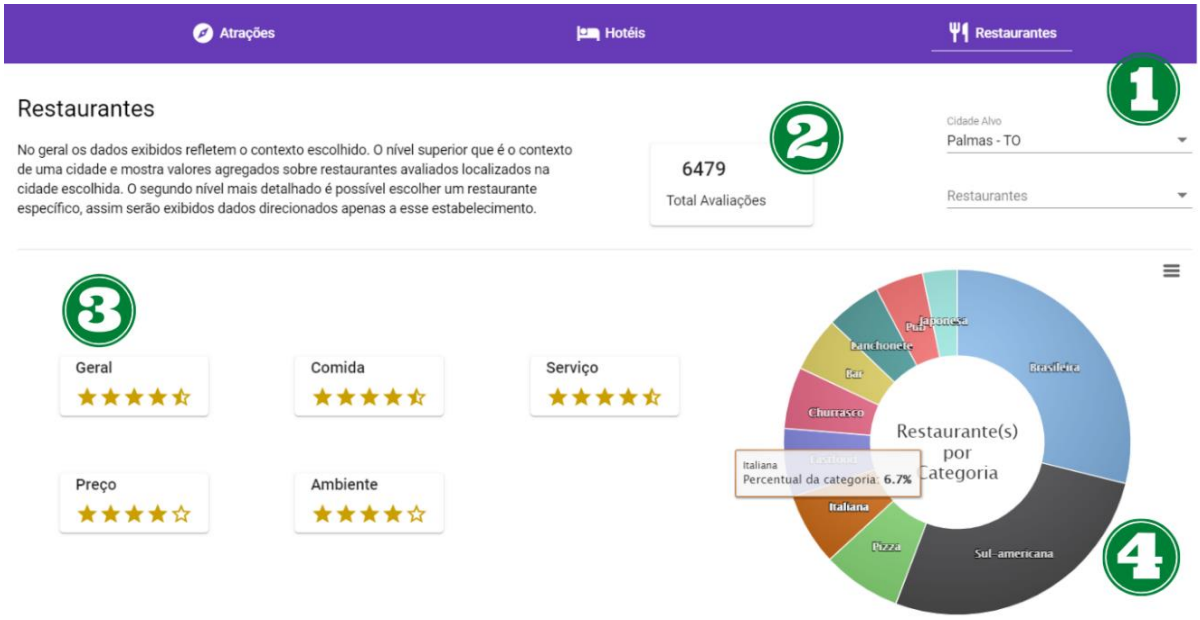
O projeto desenvolvido é composto por módulos e componentes desenvolvidos com o *framework* Angular. O formato modularizado desse *framework* facilita a reutilização de componentes gráficos. Três componentes principais agregam informações para atrações, hotéis e restaurantes, respectivamente. Para um melhor entendimento a Figura 33 ilustra a página principal para informações direcionadas a restaurantes avaliados.

**Figura 33 - Visualização da informação para Restaurantes avaliados**




A Figura 33 possui três itens que demarcam áreas que apresentam informações presentes na página de restaurantes. A proposta dessa interface para visualização de dados da SentimentALL é oferecer informações de um contexto em dois níveis, cidade ou estabelecimento. A Figura 34 exibe em detalhe da área representada pelo Item 1 na Figura 33.

**Figura 34 - Interface para seleção de contexto e gráficos para dados de restaurantes**




O item 1 da Figura 34 destaca os menus para seleção do contexto para os dados exibidos, o usuário pode selecionar entre dois níveis de detalhamento dos dados. O nível mais abrangente é direcionado a cidades, ou seja, ao selecionar uma cidade alvo o usuário poderá ver dados relacionados a restaurantes daquela localidade. O segundo nível de detalhamento possibilita ao usuário ver dados direcionados a um estabelecimento em específico, assim, após selecionar uma cidade, serão listados estabelecimentos da cidade selecionada. Sempre que o contexto da visualização de dados for modificado, os gráficos e informações na tela são atualizadas para refletir informações pertencentes a seleção do usuário. O comportamento de seleção de contexto é mantido também nas telas de Atração e Hotéis.

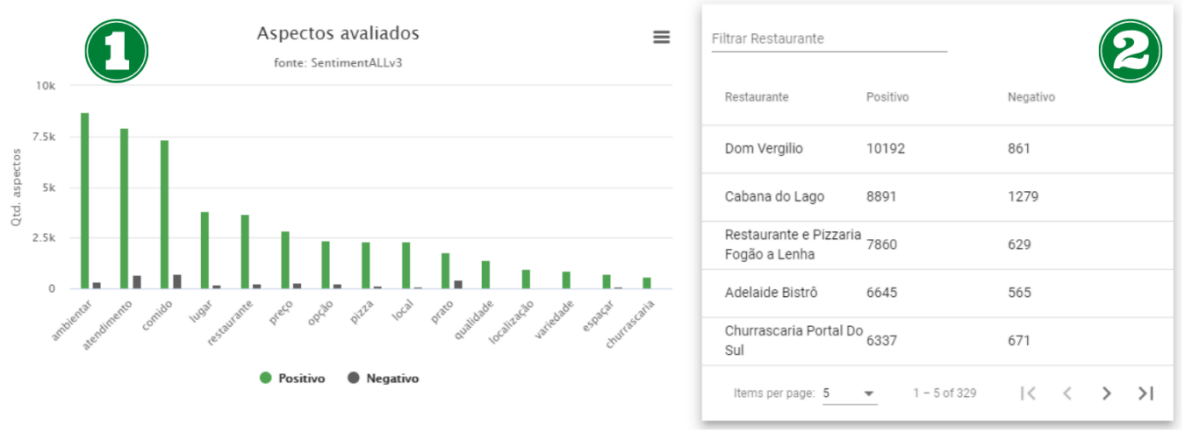
O item 2 da Figura 34 ilustra o componente que apresenta o total de avaliações para o contexto selecionado. Esse número corresponde à quantidade de avaliações que passaram pelo processo de análise de sentimentos executado pela SentimentALL relacionado ao contexto selecionado, ou seja, o total de avaliações usadas para a geração dos gráficos abaixo. O uso desse item oferece uma perspectiva maior sobre o volume de dados representados de forma gráfica e resumida.

O item 3 da Figura 34 representa os dados de votos na escala Likert para atributos de restaurantes. Os atributos para esse tipo de objeto avaliado são: geral, comida, serviço, preço e ambiente. Usuários no site TripAdvisor podem avaliar atributos entre 1 e 5 estrelas. Os dados desta tela são apresentados de duas formas, caso esteja no contexto de uma cidade, a média de cada atributo é calculada, exibindo então um resumo de como os restaurantes da

cidade foram avaliados nessas categorias. Quando um restaurante em específico é selecionado, os dados passam a ser direcionados apenas ao estabelecimento alvo.

O item 4 da Figura 34 apresenta o gráfico de distribuição percentual das 10 categorias mais presentes em uma cidade. O gráfico de rosca oferece uma visão clara da proporção de cada categoria em relação ao contexto visualizado. O Gráfico de distribuição também é exibido quando um restaurante específico é selecionado, isso porque um restaurante pode ter mais de uma categoria, nesse caso o percentual de distribuição será igual entre as categorias do estabelecimento alvo. A Figura 35 apresenta a segunda área de destaque de restaurantes da plataforma, apresentada no Item 2 da Figura 33.

**Figura 35 - Principais aspectos avaliados e tabela resumo de restaurantes**




O item 1 da Figura 35 apresenta o gráfico de barras que mostra os 15 aspectos mais avaliados para o contexto selecionado. Para cada aspecto é possível ver a quantidade de avaliações entre positivo e negativo. O gráfico de barras oferece um método simplificado para comparar a quantidade de avaliações positivas ou negativas para um aspecto e permite que sejam feitas comparações entre os aspectos avaliados.

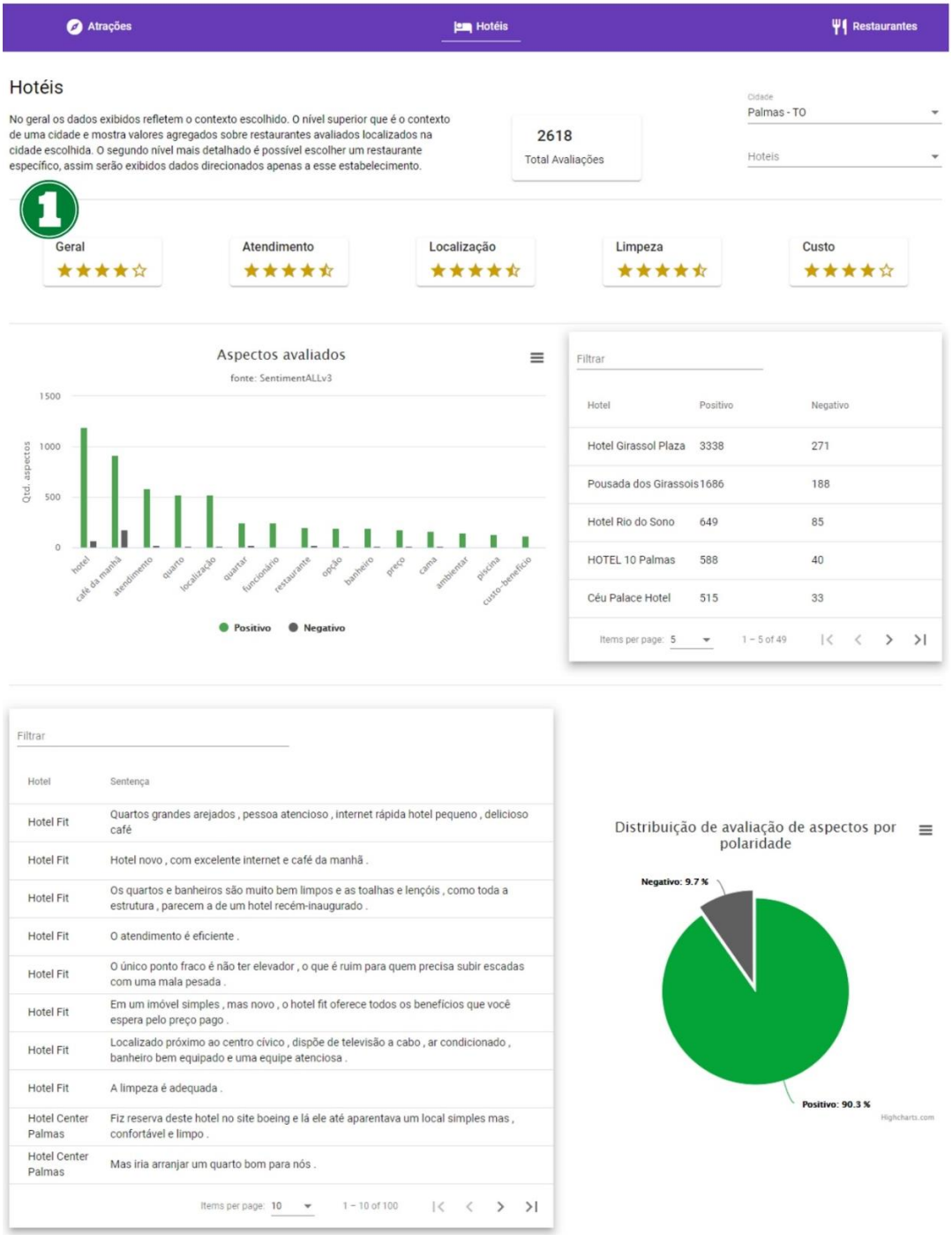
O Item 2 da Figura 35 mostra a tabela de restaurantes avaliados para o destino selecionado. Nessa tabela é possível ver uma somatória com o total de aspectos que foram avaliados de forma positiva e negativa para cada restaurante. A tabela também possui um campo de filtro para que o usuário possa pesquisar por um dos restaurantes do contexto inserindo o nome do restaurante nesse campo. O gráfico de barras para aspectos avaliados e a tabela de resumo de estabelecimentos estão disponíveis também para hotéis e atrações. A Figura 36 mostra em detalhes a área representada pelo Item 3 da Figura 33.

**Figura 36 - Tabela de sentenças exemplo e distribuição de geral de aspectos avaliados**


O item 1 da Figura 36 exibe a tabela de sentenças avaliativas do contexto. Os dados da tabela são do nome do estabelecimento e sentenças que possuem menções a aspectos avaliados. Sentenças exemplo mostram de forma concreta as avaliações obtidas e tratadas pela SentimentALL. A funcionalidade de filtro também permite uma pesquisa por termos que podem estar direcionados a um aspecto presente nas sentenças exemplo, na imagem o aspecto “Atendimento” é utilizado para filtrar e mostrar apenas frases que possuem esse termo.

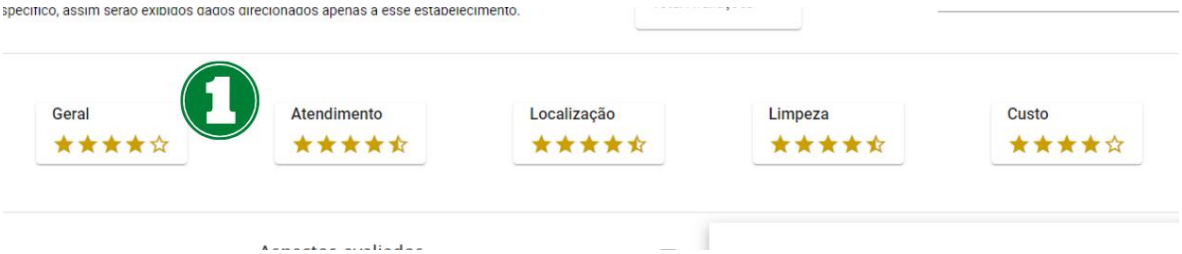
O item 2 da Figura 36 mostra o gráfico de pizza com a distribuição total de aspectos avaliados como positivo ou negativo dentro do contexto. Esse gráfico representa a distribuição de como o contexto é avaliado como um todo. Por considerar a polaridade de cada aspecto avaliado individualmente, o número também representa um percentual real da opinião de consumidores no nível de aspecto. Os itens apresentados na Figura 36 também estão presentes para avaliações destinadas a hotéis e atrações. A Figura 37 a seguir mostra em detalhes a tela para dados relacionados a hotéis.

**Figura 37 - Visualização da informação para Hotéis avaliados**



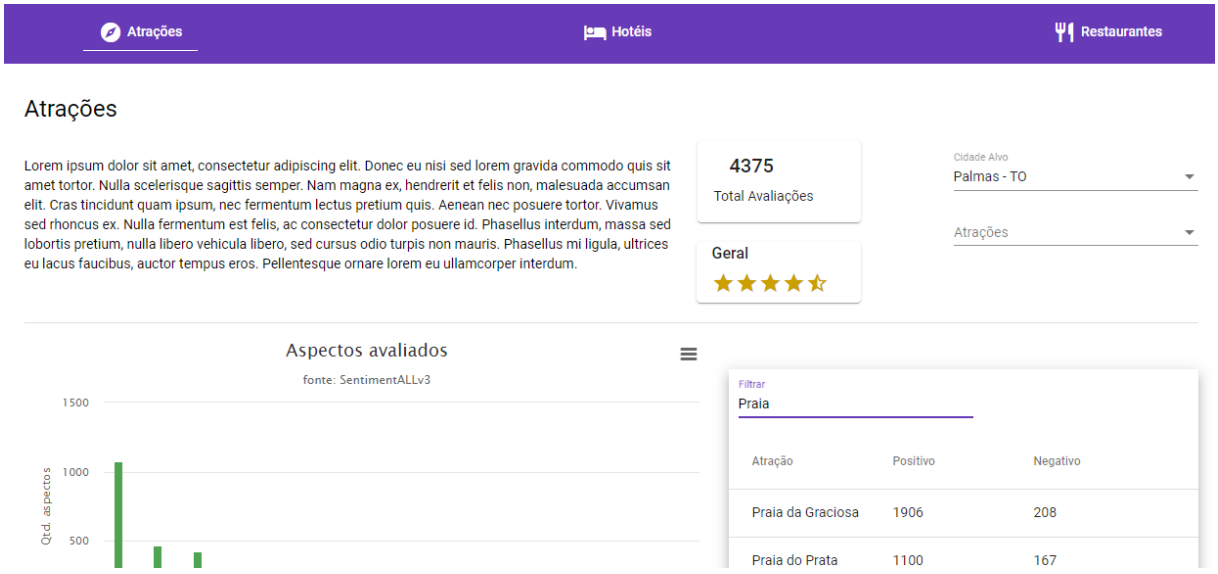

Para dados relacionados a hotéis, a maior parte da estrutura se manteve em relação a forma de apresentação da informação, o Item 1 da Figura 37 marca o item que apresenta dados específicos de hotéis. A Figura 38 exibe a área em detalhes e no item 1 são destacados os votos Likert de hotéis, considerando os atributos geral, atendimento, localização, limpeza e custo.

**Figura 38 - Atributos na escala Likert para Hotéis avaliados**




Assim como na tela restaurantes, os valores são exibidos conforme o contexto escolhido. Outras partes da interface se mantiveram iguais em relação ao apresentado na página de restaurantes avaliados, porém os dados desta tela representam informações direcionadas apenas aos hotéis avaliados. A terceira e última página do módulo de visualização da informação destaca dados direcionados a atrações avaliadas, a Figura 39, a seguir, ilustra parte dessa tela.

**Figura 39 - Visualização da informação para Atrações avaliadas**

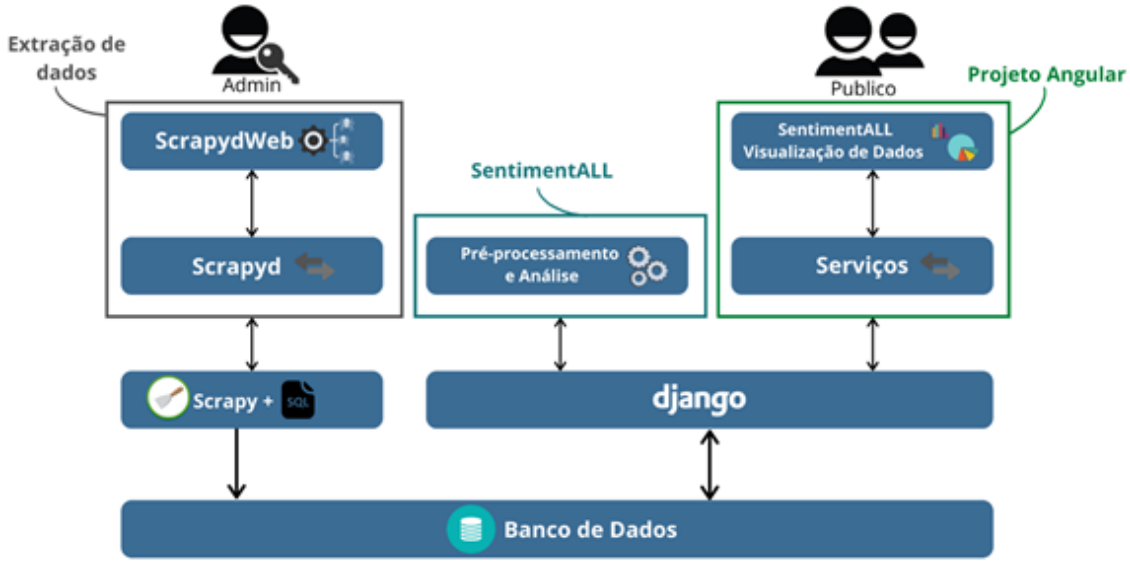



Assim como nas telas de hotéis e restaurantes, a tela para atrações manteve o padrão do tipo de gráfico exibidos, mostrando também um gráfico de barras para os aspectos mais avaliados, tabelas com exemplos de sentenças, tabela de resumo geral de atrações avaliada e o gráfico de pizza para exibir o percentual de aspectos avaliados de forma positiva e negativa. A diferença mais notável em relação aos outros tipos de objetos avaliados, é que para atrações o único voto disponível para autores dentro do TripAdvisor utilizando a escala Likert é direcionado à percepção em geral sobre atração, por esse motivo a tela de atrações é mais simples. A forma de interação e seleção do contexto se manteve também para essa tela.

## 4.5 ARQUITETURA DA PLATAFORMA

Para melhor entender como os módulos apresentados estão dispostos na plataforma desenvolvida, a Figura 40 a seguir ilustra a arquitetura final, considerando os módulos de extração de dados, processamento e AS, e  visualização da informação.

**Figura 40 - Arquitetura da Plataforma SentimentALL**




Conforme é apresentado na Figura 50, a arquitetura final da plataforma oferece dois tipos de acesso. O acesso administrativo é destinado aos usuários que terão acesso a interface ScrapydWeb, estes são responsáveis pelo gerenciamento do processo de extração de dados. A interface utiliza a camada de conexão Scrapyd e está conectada ao projeto Scrapy que é responsável pela obtenção de dados brutos do site TripAdvisor. Após obtidos, esses dados brutos são então armazenados no banco de dados da plataforma.

A ferramenta SentimentALL é parte do *back-end* da plataforma. É utilizada no processamento de comentários textuais e no processo de análise de sentimentos. Os usuários não têm interação direta com esse componente da plataforma. Os processos dessa parte são executados periodicamente, assim, de tempos em tempos a ferramenta SentimentALL faz uma busca por novas avaliações que ainda não passaram pelo processo de AS, após processar e analisar os comentários os dados resultados são então inseridos no BD. Para cada avaliação extraída do site TripAdvisor o valor do atributo “novo” é definido como verdadeiro, isso permite que a SentimentALL possa encontrar avaliações novas que ainda não passaram pelo processo de AS. Ao encontrar essas avaliações, é possível selecioná-las como alvo do processo no momento. Após realizar todas as etapas de processamento e análise dessas

avaliações, o atributo novo é definido como falso, assim, cada avaliação será avaliada apenas uma vez, evitando duplicações de dados.

O módulo público da plataforma é destinado a visualização de informações obtidas no processo de AS feito pela SentimentALL. Para obter os dados a serem visualizados, uma API Django foi desenvolvida. Essa camada de conexão é responsável por consultar os dados utilizados na geração dos gráficos e tabelas do módulo de visualização da informação. Toda a parte de agregação de dados e totalizadores é feita em consultas SQL, permitindo um bom desempenho no tempo de obtenção dessas informações. A camada de serviços do *front-end* angular é responsável por acessar as rotas dessa API e obter dados que serão então renderizados no formato de gráficos e então exibidos para o usuário.

# 5. CONSIDERAÇÕES FINAIS

Por envolver várias etapas que começam na obtenção de dados de um site alvo até a análise e visualização de informações obtidas, o desenvolvimento de sistemas usados no processamento e mineração de dados textuais oriundos da internet é um processo complexo. O desenvolvimento da plataforma para SentimentALL fornece uma maturidade maior ao sistema, que agora conta com uma arquitetura coesa, eficiente e escalável. Para o módulo de extração, boa parte do trabalho nessa parte do processo foi direcionado a esforços para manutenção dos Crawlers. Trabalhos futuros para o módulo de extração de dados podem envolver o uso de computação distribuída com o objetivo de otimizar o tempo para extração de dados, isso é importante pois essa é uma das etapas mais demoradas entre os processos da SentimentALL. O trabalho de Hagri e Djeraba (2004) descreve o desenvolvimento e implementação de um sistema para extração de dados com o uso de computação distribuição. O *framework* Scrapy também possui várias configurações e formas para tornar mais eficiente o processo de obtenção de URLs e dados, um estudo mais aprofundado deste *framework* pode trazer bons resultados quanto a performance do módulo de extração.

A desnormalização de tabelas e a criação de índices mostraram-se eficientes na melhora de performance para a consulta de dados, contudo a aplicação de técnicas para melhora da base de dados pode trazer resultados ainda melhores para a performance do BD. A aplicação de testes de performance foi necessária para comprovar que a desnormalização do modelo obteve bons resultados quanto ao tempo de resposta para consultas complexas. A contínua otimização da base de dados da plataforma é importante, pois a quantidade de dados relacionados ao contexto do turismo no Brasil cresce diariamente. Para trabalhos futuros em relação a melhorias do modelo de dados, a utilização e testes em outros SGBDs podem ser realizados como uma das formas de aprimorar os processos da plataforma. A estrutura definida neste trabalho garante um modelo simplificado, podendo ser adaptado para outras formas de bancos de dados, incluindo bancos de dados não relacionais. A pesquisa de Jantana et. al (2012) oferece um bom comparativo entre os modelos relacionais e não relacionais de bancos de dados, mostrando que a flexibilidade, escalabilidade e rapidez na consulta a dados são pontos positivos a favor de modelos não relacionais.

O uso da Lematização para a criação de índices direcionados a palavras opinativas e aspectos avaliados também se mostrou eficiente, tornando mais significativo a visualização de dados para o contexto de aspectos avaliados, porém, é importante ressaltar que o resultado do algoritmo utilizado ainda não é ideal. O resultado da Lematização utilizado produziu

termos como: “ambientar”, “comido”, “cachoeirar”, “espaçar”, esses termos normalizados pertencem a aspectos avaliados. É possível perceber que os termos resultantes desse processo tendem a ser convertidos para termos mais próximos a verbos, isso é pouco intuitivo para quem visualiza essa informação, principalmente levando em conta que em geral os aspectos avaliados são da classe gramatical de substantivos. Para trabalhos futuros, o estudo de outros algoritmos de Lematização e Stemming poderá ser feito. É importante que os algoritmos sejam utilizados no contexto do turismo e posteriormente os resultados sejam comparados para se obter o algoritmo com os melhores resultados pro contexto.

O uso de tecnologias modernas para o desenvolvimento do módulo de visualização da informação permitiu que fossem criadas em pouco tempo várias formas de visualizar para os dados da SentimentALL. O formato modular do *framework* Angular usado no desenvolvimento do *front-end* da aplicação permitiu que um tipo de visualização fosse criado e posteriormente replicado para contextos diferentes. Essa flexibilidade e facilidade de interação entre componentes permitiu também que novas formas de apresentação de dados fossem implementadas. Para trabalhos futuros é importante que seja feito um estudo para melhor definir quais os métodos ideais para apresentar dados relacionados ao turismo brasileiro. O volume de variáveis e dados disponíveis para os diversos contextos dentre os dados presentes na sentimentALL tem potencial para que esses dados sejam explorados de diversas formas.

A sistematização da SentimentALL em uma plataforma unificada permitiu que os diversos processos estivessem mais bem organizados e fossem mais facilmente gerenciados. Porém, a parte que envolve o pré-processamento de dados e análise de sentimentos foi pouco evoluída desde a versão de Araújo (2017). Para trabalhos futuros é importante que algoritmos sejam testados possibilitando verificar se algoritmos disponíveis atualmente possuem uma performance melhor em comparação aos algoritmos presentes nos processos da SentimentALL na sua versão atual. O aprimoramento de etapas como a identificação de palavras opinativas e aspectos, a definição do relacionamento de dependência sintática entre palavras e o uso de algoritmos de aprendizado de máquina podem ajudar a melhorar ainda mais a precisão com que a ferramenta processa dados e realiza a análise de sentimentos.

## REFERÊNCIAS


AGGARWAL, C. C.; ZHAI, C. X. (Ed.). **Mining text data**. Springer Science & Business Media, 2012.

ALUÍSIO, S.; PELIZZONI, J.; MARCHI, A.R.; de OLIVEIRA, L.; MANENTI, R.; MARQUIAFÁVEL, V. 2003. An account of the challenge of tagging a reference corpus for brazilian portuguese. In: Proceedings of the 6th International Conference on Computational Processing of the Portuguese Language. PROPOR 2003

ARAÚJO, L. G. de A. **SENTIMENTALL VERSÃO 2: Desenvolvimento de Análise de Sentimentos em Python.** 2019. 123 f. TCC (Graduação) - Curso de Ciência da Computação, Centro Universitário Luterano de Palmas, Palmas, 2017.

BAHMANI, A. H.; NAGHIBZADEH, M.; BAHMANI, B. Automatic database normalization and primary key generation. **2008 Canadian Conference On Electrical And Computer Engineering**, p.11-16, maio 2008. IEEE.

BERTIN, J. **Semiology of graphics; diagrams networks maps**. Wisconsin: Esri Press, 2010. 438 p.

BEUTEL, A. et al. A Machine Learning Approach to Databases Indexes. In: **ML Systems Workshop**. NIPS, 2017.

BHATIA, M. P. S.; GUPTA, D. Discussion on web Crawlers of search engine. In: **COIT-2008**, 2008.

BRITO, P. F. de. **RELATOS VERBAIS DE CONSUMIDORES EM AVALIAÇÕES ON-LINE:** PROSPECÇÃO COMPUTACIONAL E INTERPRETAÇÕES COM BASE NO BEHAVIORAL PERSPECTIVE MODEL (BPM). 2018. 182 f. Tese (Programa de Pós-Graduação STRICTO SENSU em Psicologia) - Pontifícia Universidade Católica de Goiás, Goiânia-GO.

CARD, S. K. INFORMATION VISUALIZATION. In: SEARS, Andrew; JACKO, Julie A. **The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications**. 2. ed. Mahwah, Nj: Lawrence Erlbaum Associates, 2008. Cap. 26. p. 509-543.

CARD, S. K.; Mackinlay, J. D.; Shneiderman, B. **Information visualization: Using vision to think**. San Francisco: Morgan Kaufmann Publishers, 1999.

CARVALHO, P.; SILVA, M. J. **SentiLex-PT: Principais características e potencialidades**. Oslo Studies in Language, v. 7, n. 1, 2015.

CHEN, M. et al. **Data, information, and knowledge in visualization**. IEEE Computer Graphics and Applications, v. 29, n. 1, p. 12-19, 2008. Disponível em: <https://homepages.cwi.nl/~robertl/articles/cga2009.pdf>. Acesso em: 30 Out. 2019.

DAMERAU F. J.; MAYS E. **An examination of undetected typing errors. Information Processing and Management**, Vol. 25, No. 6, pp. 659-664, (1989).

DIKAIAKOS, M. D.; STASSOPOULOU, A.; PAPAGEORGIOU, L. An investigation of web crawler behavior: characterization and metrics. **Computer Communications**, v. 28, n. 8, p. 880-897, 2005.

ERINLE, B. **Teste de desempenho com JMeter 3**: melhore o desempenho de sua aplicação web. [s. L.]: Novatec, 2017.

FERRARA, E. et al. Web data extraction, applications and techniques: A survey. **Knowledge-based systems**, v. 70, p. 301-323, 2014.

FONSECA, E. R.; ROSA, J. L. G. Mac-morpho revisited: Towards robust part-of-speech tagging. In: **Proceedings of the 9th Brazilian symposium in information and human language technology**. 2013.

GRUS, J. **Data Science do Zero**: Primeiras Regras com o Python. Rio de Janeiro: Alta Books, 2016. 336 p. Tradução de: Welington Nascimento.

HAFRI, Younes; DJERABA, Chabane. High performance crawling system. In: **Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval**. 2004. p. 299-306.

HU, X. **Casual Information Visualization-Based Major Consulting System Design**. 2017. Tese de Doutorado. Purdue University.

HUANG, Z.; EIDELMAN, V.; HARPER, M. Improving a simple bigram HMM part-of-speech tagger by latent annotation and self-training. In: **Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics**, Companion Volume: Short Papers. Association for Computational Linguistics, 2009. p. 213-216.

INGASON, A. K. et al. A mixed method lemmatization algorithm using a hierarchy of linguistic identities (HOLI). In: **International Conference on Natural Language Processing**. Springer, Berlin, Heidelberg, 2008. p. 205-216.

JATANA, N. et al. A survey and comparison of relational and non-relational database. **International Journal of Engineering Research & Technology**, v. 1, n. 6, p. 1-5, 2012. Disponível em: <https://www.ijert.org/research/a-survey-and-comparison-of-relational-and-non-relational-database-IJERTV1IS6024.pdf> Acesso em: 19 Jun. 2020.

JURISH, B.; WÜRZNER, K. **Word and Sentence Tokenization with Hidden Markov Models**. JLCL, v. 28, n. 2, p. 61-83, 2013. Disponível em: <https://www.researchgate.net/publication/259772781_Word_and_Sentence_Tokenization_with_Hidden_Markov_Models> Acesso em: 16 Out. 2019.

KORENIUS, T. et al. Stemming and lemmatization in the clustering of finnish text documents. In: **Proceedings of the thirteenth ACM international conference on Information and knowledge management**. ACM, 2004. p. 625-633.

LIU, B. **Sentiment Analysis and Opinion Mining**. [s.l.]: Morgan & Claypool Publishers, 2012. 167 p. Disponível em:<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.244.9480&rep=rep1&type=pdf > . Acesso em: 26 Ago. 2019.

LIU SHI, S. et al. **A survey on information visualization: recent advances and challenges**. The Visual Computer, v. 30, n. 12, p. 1373-1393, 2014.

MENCZER, F. et al. Evaluating topic-driven web crawlers. In: **Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval**. ACM, 2001. p. 241-249.

NEVEDROV, D. **Using JMeter to Performance Test Web Services**. Published on dev2dev, 2006.

OLIVEIRA, Taylor Santos. **DESENVOLVIMENTO E VERIFICAÇÃO DO MÓDULO DE ANÁLISE DE SENTIMENTOS - NÍVEL DE DOCUMENTO DA SENTIMENTALL**. 2019. 76 f. TCC (Graduação) - Curso de Ciência da Computação, Centro Universitário Luterano de Palmas, Palmas, 2019. Disponível em: <https://ulbra-to.br/bibliotecadigital/publico/home/documento/685> Acesso em: 21 Jun. 2020.

ORENGO, V. M.; HUYCK, C. A stemming algorithm for the portuguese language. In: **Proceedings Eighth Symposium on String Processing and Information Retrieval**. IEEE, 2001. p. 186-193.

PANG B. e LEE L., Opinion Mining and Sentiment Analysis, **Foundations and Trends® in Information Retrieval**, vol 2, nos 1–2, p. 1–135, 2008.

PINTO, Yma. A framework for systematic database de-normalization. In: **Global Journal of Computer Science and Technology**, 2009, p. 44-52.

PURBA, S. **High-Performance Web Databases**: design, development, and deployment. Nova York: Auerbach Publications, 2000.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. 3. ed. Rio de Janeiro: Elsevier, 2013. 1016 p. Tradução de: Regina Célia Simille.

SANDERS, G.; SHIN, S. Denormalization Effects on Performance of RDBMS. In: **Proceedings of the 34th Annual Hawaii International Conference on System Sciences**. 2001. p. 3013-3013.

SOUSA, F. R. M. de. **Implementação de um Dashboard para a ferramenta SentimentALL**. 2017. 61 f. TCC (Graduação) - Curso de Sistemas de Informação, Centro Universitário Luterano de Palmas, Palmas, 2017.

SCHROEDER, W. J.; LORENSEN, B.; MARTIN, Ken. **The visualization toolkit: an object-oriented approach to 3D graphics**. Kitware, 2004.

SHIN, S. K.; SANDERS, G. L. **Denormalization strategies for data retrieval from data warehouses**. Decision Support Systems, v. 42, n. 1, p. 267-282, 2006.

TRIPADVISOR, **TripAdvisor Reports Third Quarter 2019 Financial Results**. Needham, MA. 2019. Disponível em: <http://ir.tripadvisor.com/static-files/7adc41a2-c5bb-418b-bfc1-3f3d12e8d075> Acesso em: 15 Nov. 2019.

TUFTE, E. R. **The visual display of quantitative information**. Cheshire: Graphics Press, 1983. 197 p.

UNWIN, A.; CHEN, C.; HÄRDLE, W. **Computational Statistics and Data Visualization**. Humboldt-Universität zu Berlin, Wirtschaftswissenschaftliche Fakultät, 2007.

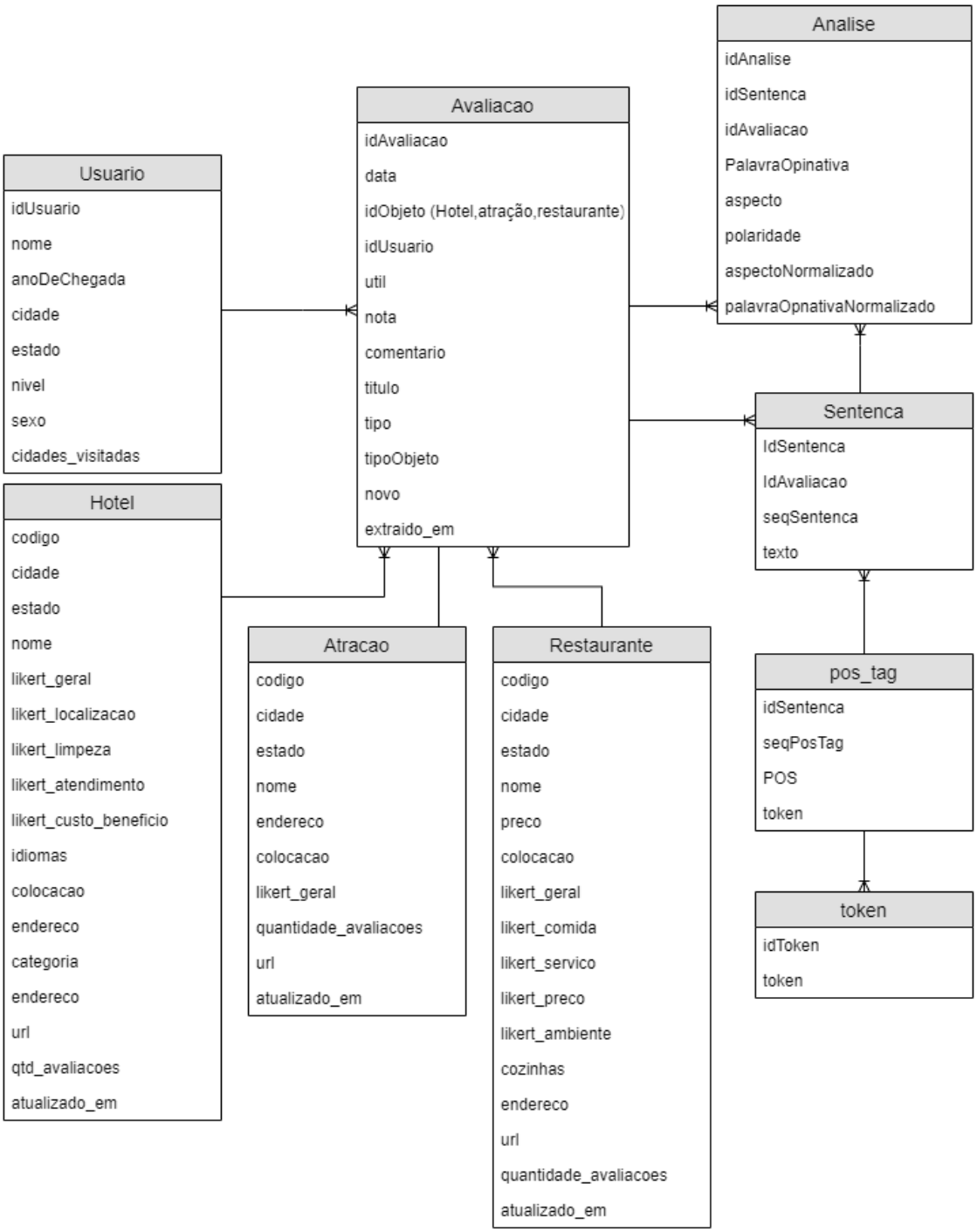
VARGIU, E.; URRU, M. Exploiting web scraping in a collaborative filtering-based approach to web advertising. **Artif. Intell. Research**, v. 2, n. 1, p. 44-54, 2013.

WATZLAWIK, M.; VALSINER, J. **A Comparative Study of Stemming Algorithms**. The Oxford Handbook of Culture and Psychology, v. 2, n. 6, p. 1930–1938, 2012.

XIE, D. X.; XIA, W. F. Design and Implementation of the Topic-Focused Crawler Based on Scrapy. In: **advanced materials research**. Trans Tech Publications, 2014. p. 487-490.

**APÊNDICES**

APÊNDICE A – Modelo completo do banco de dados da plataforma



APÊNDICE B – Instruções SQL utilizadas para teste de performance dos BDs

| SQL TESTE #1 | |
|---|---|
| v2 | **SELECT** o.idCidade, a.aspecto, a.polaridade, COUNT(*) as total<br>**FROM** SentimentALLv2.dbo.análise as a<br>**INNER JOIN** SentimentALLv2.dbo.sentença as s on a.idSentença=s.idSentença<br>**INNER JOIN** SentimentALLv2.dbo.avaliação as av on av.idAvaliação=s.idAvaliação<br>**INNER JOIN** SentimentALLv2.dbo.objeto as o on o.idObjeto = av.idDestino<br>**WHERE** o.idCidade = 'g1012170' AND o.tipo = 'Hotel'<br>**GROUP BY** o.idCidade, a.aspecto, a.polaridade |
| v3 | **SELECT** av.cidadeObjeto, a.aspecto, a.polaridade, COUNT(*) as Total<br>**FROM** SentimentALLv3.dbo.analise as a<br>**INNER JOIN** SentimentALLv3.dbo.avaliacao as av<br>on av.idAvaliacao = a.idAvaliacao<br>**INNER JOIN** SentimentALLv3.dbo.hotel as h on h.codigo = av.idObjeto<br>**WHERE** h.cidade = 'Caldas Novas'<br>**GROUP BY** av.cidadeObjeto, a.aspecto, a.polaridade |
| SQL TESTE #2 | |
| v2 | **SELECT** a.polaridade, COUNT(*) as total **FROM** SentimentALLv2.dbo.análise as a<br>**INNER JOIN** SentimentALLv2.dbo.sentença as s on a.idSentença = s.idSentença<br>**INNER JOIN** SentimentALLv2.dbo.avaliação as av on av.idAvaliação = s.idAvaliação<br>**INNER JOIN** SentimentALLv2.dbo.objeto as o on o.idObjeto = av.idDestino<br>**WHERE** o.idCidade = 'g1012170' AND o.tipo = 'Hotel'<br>**GROUP BY** a.polaridade |
| v3 | **SELECT** a.polaridade, COUNT(*) AS Total<br>**FROM** SentimentALLv3.dbo.analise as a<br>**INNER JOIN** SentimentALLv3.dbo.avaliacao as av<br>on av.idAvaliacao = a.idAvaliacao<br>**INNER JOIN** SentimentALLv3.dbo.hotel as o<br>on o.codigo = av.idObjeto<br>**WHERE** o.cidade = 'Caldas Novas'<br>**GROUP BY** a.polaridade |
| SQL TESTE #3 | |
| v2 | **SELECT** o.idObjeto, a.polaridade, COUNT(*) as total<br>**FROM** SentimentALLv2.dbo.análise as a<br>**INNER JOIN** SentimentALLv2.dbo.sentença as s<br>on a.idSentença=s.idSentença<br>**INNER JOIN** SentimentALLv2.dbo.avaliação as av<br>on av.idAvaliação=s.idAvaliação<br>**INNER JOIN** SentimentALLv2.dbo.objeto as o on o.idObjeto = av.idDestino<br>**WHERE** o.idCidade = 'g1012170' AND o.tipo = 'Hotel'<br>**GROUP BY** o.idObjeto, a.polaridade<br>**ORDER BY** o.idObjeto, a.polaridade |

| | |
|---|---|
| v3 | **SELECT** h.codigo, a.polaridade, COUNT(*) as total<br>**FROM** SentimentALLv3.dbo.analise as a<br>**INNER JOIN** SentimentALLv3.dbo.avaliacao as av on<br>av.idAvaliacao=a.idAvaliacao<br>**INNER JOIN** SentimentALLv3.dbo.hotel as h on h.codigo = av.idObjeto<br>**WHERE** h.cidade = 'Caldas Novas'<br>**GROUP BY** h.codigo, a.polaridade<br>**ORDER BY** h.codigo, a.polaridade |
| SQL TESTE #4 | |
| v2 | **SELECT** texto **FROM** SentimentALLv2.dbo.sentença WHERE idSentença in (<br>**SELECT DISTINCT** TOP(10) s.idSentença<br>**FROM** SentimentALLv2.dbo.análise as a<br>**INNER JOIN** SentimentALLv2.dbo.sentença as s<br>on a.idSentença=s.idSentença<br>**INNER JOIN** SentimentALLv2.dbo.avaliação as av on<br>av.idAvaliação=s.idAvaliação<br>**INNER JOIN** SentimentALLv2.dbo.objeto as o on o.idObjeto = av.idDestino<br>**WHERE** o.idCidade = 'g1012170' AND o.tipo = 'Hotel' AND a.polaridade = 1<br>AND a.aspecto = 'cama'<br>) |
| v3 | **SELECT** texto **FROM** SentimentALLv3.dbo.sentenca WHERE idSentenca in (<br>**SELECT DISTINCT** TOP(10) s.idSentenca<br>**FROM** SentimentALLv3.dbo.analise as a<br>**INNER JOIN** SentimentALLv3.dbo.sentenca as s on<br>a.idSentenca = s.idSentenca<br>**INNER JOIN** SentimentALLv3.dbo.avaliacao as av on<br>av.idAvaliacao=s.idAvaliacao<br>**INNER JOIN** SentimentALLv3.dbo.hotel as h on h.codigo = av.idObjeto<br>**WHERE** h.cidade = 'Caldas Novas' AND a.polaridade = 1<br>AND a.aspecto = 'cama'<br>) |
| SQL TESTE #5 | |
| v2 | **SELECT** COUNT(*) as 'Total usuario nível 1' **FROM**<br>SentimentALLv2.dbo.objeto o<br>**INNER JOIN** SentimentALLv2.dbo.avaliação a ON a.idDestino = o.idObjeto<br>**INNER JOIN** SentimentALLv2.dbo.usuário u ON u.idUsuário = a.idUsuário<br>**INNER JOIN** SentimentALLv2.dbo.cidade c ON c.idCidade = o.idCidade<br>**WHERE** o.idCidade = 'g1012170' AND o.tipo = 'Hotel' AND a.nota = 1 AND<br>u.nível = 0 |
| v3 | **SELECT** COUNT(*) as 'Total usuario nível 1'<br>**FROM** SentimentALLv3.dbo.hotel as h<br>**INNER JOIN** SentimentALLv3.dbo.avaliacao a ON a.idObjeto = h.codigo<br>**INNER JOIN** SentimentALLv3.dbo.usuario u ON u.idUsuario = a.idUsuario<br>**WHERE** h.cidade = 'Caldas Novas' AND a.nota = 1 AND u.nível = 0 |

APÊNDICE C – Tabelas resultado da execução de testes de Performance com JMeter

**TESTE #1**

**V2**

| timeStamp | elapsed | dataType | bytes | Latency | Connect |
|---|---|---|---|---|---|
| 1,59027E+12 | 164458 | text | 97803 | 164334 | 652 |
| 1,59027E+12 | 4839 | text | 97803 | 4724 | 0 |
| 1,59027E+12 | 4817 | text | 97803 | 4728 | 0 |
| 1,59027E+12 | 4717 | text | 97803 | 4669 | 0 |
| 1,59027E+12 | 4716 | text | 97803 | 4672 | 0 |
| 1,59027E+12 | 4668 | text | 97803 | 4621 | 0 |
| 1,59027E+12 | 4713 | text | 97803 | 4671 | 0 |
| 1,59027E+12 | 4707 | text | 97803 | 4663 | 0 |
| 1,59027E+12 | 4732 | text | 97803 | 4689 | 0 |
| 1,59027E+12 | 4683 | text | 97803 | 4638 | 0 |

**V3**

| timeStamp | elapsed | dataType | bytes | Latency | Connect |
|---|---|---|---|---|---|
| 1,59027E+12 | 59455 | text | 114547 | 59389 | 737 |
| 1,59027E+12 | 2450 | text | 114547 | 2420 | 0 |
| 1,59027E+12 | 2451 | text | 114547 | 2426 | 0 |
| 1,59027E+12 | 2437 | text | 114547 | 2426 | 0 |
| 1,59027E+12 | 2290 | text | 114547 | 2278 | 0 |
| 1,59027E+12 | 2323 | text | 114547 | 2311 | 0 |
| 1,59027E+12 | 2375 | text | 114547 | 2362 | 0 |
| 1,59027E+12 | 2296 | text | 114547 | 2286 | 0 |
| 1,59027E+12 | 2285 | text | 114547 | 2273 | 0 |
| 1,59027E+12 | 2354 | text | 114547 | 2343 | 0 |

**TESTE #2**

**V2**

| timeStamp | elapsed | dataType | bytes | Latency | Connect |
|---|---|---|---|---|---|
| 1,59027E+12 | 141811 | text | 28 | 141788 | 605 |
| 1,59027E+12 | 3856 | text | 28 | 3856 | 0 |
| 1,59027E+12 | 3691 | text | 28 | 3691 | 0 |
| 1,59027E+12 | 3683 | text | 28 | 3683 | 0 |
| 1,59027E+12 | 3746 | text | 28 | 3746 | 0 |
| 1,59027E+12 | 3704 | text | 28 | 3704 | 0 |
| 1,59027E+12 | 3719 | text | 28 | 3719 | 0 |
| 1,59027E+12 | 3700 | text | 28 | 3700 | 0 |
| 1,59027E+12 | 3712 | text | 28 | 3711 | 0 |
| 1,59027E+12 | 3721 | text | 28 | 3720 | 0 |

**V3**

| timeStamp | elapsed | dataType | bytes | Latency | Connect |
|---|---|---|---|---|---|
| 1,59027E+12 | 59152 | text | 28 | 59135 | 663 |
| 1,59027E+12 | 2375 | text | 28 | 2374 | 0 |
| 1,59027E+12 | 2116 | text | 28 | 2116 | 0 |
| 1,59027E+12 | 2072 | text | 28 | 2072 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 1,59027E+12 | 2243 | text | 28 | 2243 | 0 |
| | 1,59027E+12 | 2220 | text | 28 | 2220 | 0 |
| | 1,59027E+12 | 2063 | text | 28 | 2063 | 0 |
| | 1,59027E+12 | 2040 | text | 28 | 2040 | 0 |
| | 1,59027E+12 | 2128 | text | 28 | 2128 | 0 |
| | 1,59027E+12 | 2104 | text | 28 | 2104 | 0 |

**TESTE #3**

| **V2** | **timeStamp** | **elapsed** | **dataType** | **bytes** | **Latency** | **Connect** |
|---|---|---|---|---|---|---|
| | 1,59027E+12 | 143488 | text | 1193 | 143465 | 642 |
| | 1,59027E+12 | 3840 | text | 1193 | 3836 | 0 |
| | 1,59027E+12 | 4074 | text | 1193 | 4071 | 0 |
| | 1,59027E+12 | 3847 | text | 1193 | 3844 | 0 |
| | 1,59027E+12 | 3881 | text | 1193 | 3880 | 0 |
| | 1,59027E+12 | 3762 | text | 1193 | 3761 | 0 |
| | 1,59027E+12 | 3901 | text | 1193 | 3900 | 0 |
| | 1,59027E+12 | 3972 | text | 1193 | 3972 | 0 |
| | 1,59027E+12 | 3856 | text | 1193 | 3855 | 0 |
| | 1,59027E+12 | 3889 | text | 1193 | 3888 | 0 |

| **V3** | **timeStamp** | **elapsed** | **dataType** | **bytes** | **Latency** | **Connect** |
|---|---|---|---|---|---|---|
| | 1,59027E+12 | 59500 | text | 1191 | 59484 | 697 |
| | 1,59027E+12 | 2398 | text | 1191 | 2396 | 0 |
| | 1,59027E+12 | 2076 | text | 1191 | 2073 | 0 |
| | 1,59027E+12 | 2117 | text | 1191 | 2116 | 0 |
| | 1,59027E+12 | 2156 | text | 1191 | 2156 | 0 |
| | 1,59027E+12 | 2139 | text | 1191 | 2139 | 0 |
| | 1,59027E+12 | 2161 | text | 1191 | 2160 | 0 |
| | 1,59027E+12 | 2131 | text | 1191 | 2130 | 0 |
| | 1,59027E+12 | 2077 | text | 1191 | 2077 | 0 |
| | 1,59027E+12 | 2219 | text | 1191 | 2218 | 0 |

**TESTE #4**

| **V2** | **timeStamp** | **elapsed** | **dataType** | **bytes** | **Latency** | **Connect** |
|---|---|---|---|---|---|---|
| | 1,59027E+12 | 32926 | text | 1786 | 32913 | 605 |
| | 1,59027E+12 | 2924 | text | 1786 | 2924 | 0 |
| | 1,59027E+12 | 2904 | text | 1786 | 2903 | 0 |
| | 1,59027E+12 | 2914 | text | 1786 | 2914 | 0 |
| | 1,59027E+12 | 2909 | text | 1786 | 2909 | 0 |
| | 1,59027E+12 | 2895 | text | 1786 | 2895 | 0 |
| | 1,59027E+12 | 2899 | text | 1786 | 2899 | 0 |
| | 1,59027E+12 | 2909 | text | 1786 | 2909 | 0 |
| | 1,59027E+12 | 2877 | text | 1786 | 2877 | 0 |
| | 1,59027E+12 | 2909 | text | 1786 | 2909 | 0 |

**V3**

| timeStamp | elapsed | dataType | bytes | Latency | Connect |
|---|---|---|---|---|---|
| 1,59027E+12 | 7472 | text | 1786 | 7466 | 762 |
| 1,59027E+12 | 27 | text | 1786 | 27 | 0 |
| 1,59027E+12 | 22 | text | 1786 | 22 | 0 |
| 1,59027E+12 | 22 | text | 1786 | 22 | 0 |
| 1,59027E+12 | 23 | text | 1786 | 22 | 0 |
| 1,59027E+12 | 23 | text | 1786 | 23 | 0 |
| 1,59027E+12 | 22 | text | 1786 | 22 | 0 |
| 1,59027E+12 | 22 | text | 1786 | 22 | 0 |
| 1,59027E+12 | 24 | text | 1786 | 23 | 0 |
| 1,59027E+12 | 23 | text | 1786 | 23 | 0 |

**TESTE #5**

**V2**

| timeStamp | elapsed | dataType | bytes | Latency | Connect |
|---|---|---|---|---|---|
| 1,59027E+12 | 57404 | text | 38 | 57384 | 595 |
| 1,59027E+12 | 221 | text | 38 | 221 | 0 |
| 1,59027E+12 | 205 | text | 38 | 205 | 0 |
| 1,59027E+12 | 206 | text | 38 | 206 | 0 |
| 1,59027E+12 | 218 | text | 38 | 218 | 0 |
| 1,59027E+12 | 207 | text | 38 | 207 | 0 |
| 1,59027E+12 | 212 | text | 38 | 212 | 0 |
| 1,59027E+12 | 215 | text | 38 | 215 | 0 |
| 1,59027E+12 | 210 | text | 38 | 209 | 0 |
| 1,59027E+12 | 209 | text | 38 | 209 | 0 |

**V3**

| timeStamp | elapsed | dataType | bytes | Latency | Connect |
|---|---|---|---|---|---|
| 1,59027E+12 | 32737 | text | 27 | 32718 | 694 |
| 1,59027E+12 | 79 | text | 27 | 79 | 0 |
| 1,59027E+12 | 61 | text | 27 | 61 | 0 |
| 1,59027E+12 | 62 | text | 27 | 62 | 0 |
| 1,59027E+12 | 61 | text | 27 | 61 | 0 |
| 1,59027E+12 | 62 | text | 27 | 62 | 0 |
| 1,59027E+12 | 62 | text | 27 | 62 | 0 |
| 1,59027E+12 | 61 | text | 27 | 61 | 0 |
| 1,59027E+12 | 64 | text | 27 | 64 | 0 |
| 1,59027E+12 | 62 | text | 27 | 62 | 0 |