Universidade de Brasília

Faculdade de Tecnologia

Modelagem e Construção de uma Base Pública de Relatos de Consumidores

Beatriz Mingorance Sarmento

PROJETO FINAL DE CURSO

ENGENHARIA DE REDES DE COMUNICAÇÃO

Brasília

2025

Universidade de Brasília

Faculdade de Tecnologia

Modelagem e Construção de uma Base Pública de Relatos de Consumidores

Beatriz Mingorance Sarmento

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheira de Redes de Comunicação.

Orientador: Prof. Dr. Daniel Guerreiro e Silva

Brasília

2025

Universidade de Brasília
Faculdade de Tecnologia

Modelagem e Construção de uma Base Pública de Relatos de Consumidores

Beatriz Mingorance Sarmento

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheira de Redes de Comunicação.

Trabalho aprovado. Brasília, 11 de fevereiro de 2025:

Prof. Dr. Daniel Guerreiro e Silva,
UnB/FT/ENE
Orientador

Prof. Dr. Fábio Lúcio Lopes de Mendonça,
UnB/FT/ENE
Examinador interno

Prof. Dra. Cláudia Jacy Barenco Abbas,
UnB/FT/ENE
Examinador interno

Este trabalho é dedicado a todas as pessoas que vivem
na era digital e reconhecem o valor dos dados como
um recurso essencial para transformar e compreender o mundo atual.

Agradecimentos

Gostaria de agradecer à minha família, que sempre me proporcionou todo o apoio e encorajamento necessários para que eu pudesse perseguir e realizar os meus sonhos. Agradeço também aos meus amigos, dos mais antigos aos mais novos, que estiveram ao meu lado nos momentos mais desafiadores, oferecendo palavras de incentivo, companhia e motivação, contribuindo de forma significativa para o alcance dos meus objetivos.

“Information is the oil of the 21st century,
and analytics is the combustion engine.”
(Peter Sondergaard)

Resumo

Este trabalho aborda o desenvolvimento de uma base pública de dados estruturados a partir de relatos de consumidores extraídos do site Consumidor.gov.br (Consumidor.gov.br,). A motivação principal está na necessidade de fornecer acesso fácil e automatizado a dados públicos que, muitas vezes, são disponibilizados de forma desestruturada e de difícil manipulação, limitando sua utilização por pesquisadores e profissionais. Para superar esses desafios, foi empregada uma abordagem baseada em técnicas de web scraping, associada a metodologias de modelagem de banco de dados, processos ETL (Extração, Transformação e Carga), e análise de dados com *machine learning*.

Na etapa metodológica, foram utilizadas ferramentas como Python, Selenium e Beautiful Soup para realizar a extração automatizada dos relatos e suas respectivas respostas. A coleta de dados enfrentou desafios técnicos, como a inconsistência no funcionamento dos filtros do site e limitações impostas pelo carregamento dinâmico das páginas. Esses obstáculos foram superados por meio de ajustes no código e na estratégia de extração, garantindo a integridade e a completude dos dados coletados. O resultado foi um banco de dados relacional criado no MySQL, organizado para facilitar o acesso e a análise.

Após a estruturação dos dados, foi realizada uma análise descritiva que destacou padrões e *insights* relevantes, como o desempenho das empresas na resolução de conflitos. Adicionalmente, foi explorado o potencial dos dados para aplicações de *machine learning*, implementando algoritmos clássicos para demonstração de viabilidade. Esses resultados mostram como a extração e organização de dados podem abrir novas oportunidades para pesquisas e para o aprimoramento de políticas públicas baseadas em evidências.

Por fim, este trabalho destaca a importância de iniciativas que promovam a transparência e a acessibilidade de dados públicos. A criação de uma base pública estruturada possibilita não apenas o avanço de estudos acadêmicos, mas também a inovação tecnológica e a maior eficiência na tomada de decisões por parte de gestores e pesquisadores. O projeto demonstra que, apesar das barreiras técnicas e legais, é possível transformar dados desorganizados em recursos valiosos para a sociedade.

Todos os links relevantes para esse trabalho podem ser encontrados aqui.

Palavras-chave: Dados. Web Scraping. Relatos. Banco de Dados.

Lista de figuras

Figura 1.1 Evolução da quantidade de empresas participantes e de reclamações finalizadas até 2022. 11
Figura 1.2 Evolução da quantidade de empresas participantes e de reclamações finalizadas até 2023. 11
Figura 2.1 Ilustração das funcionalidades de Beautiful Soup. 14
Figura 2.2 Conceitos de Modelagem ETL. 16
Figura 2.3 Aprendizado Supervisionado. 18
Figura 2.4 Um exemplo de Árvore de Decisão. 19
Figura 2.5 Máquina de Vetores de Suporte. 20
Figura 2.6 Previsão do Status de Diabetes com SVM. 21
Figura 2.7 Algoritmo K-Means. 21
Figura 3.1 Tela inicial do site Consumidor.gov.br. 23
Figura 3.2 Página que exibe cartões com relatos de consumidores. 23
Figura 3.3 Dados que compõem um cartão. 24
Figura 3.4 Filtro presente no site Consumidor.gov.br. 25
Figura 3.5 Ilustração do carregamento da filtragem. 26
Figura 3.6 Tela após *timeout*. 26
Figura 4.1 Banco de Dados MySQL. 29
Figura 4.2 Nuvem de palavras dos relatos. 30
Figura 4.3 Nuvem de palavras das respostas. 30
Figura 4.4 Nuvem de palavras dos comentários. 31
Figura 4.5 Status de todos os relatos. 31
Figura 4.6 Número de reclamações por mês. 32
Figura 4.7 As 5 empresas com mais reclamações em 2022. 33
Figura 4.8 As 5 empresas com mais reclamações em 2023. 33
Figura 4.9 As 5 empresas com mais reclamações em 2024. 34
Figura 4.10 Percentual de reclamações resolvidas pela Latam. 35
Figura 4.11 Percentual de reclamações resolvidas pelo Hurb - Hotel Urbano. 35
Figura 4.12 Percentual de reclamações resolvidas pela Serasa Experian. 36
Figura 4.13 Matriz de confusão do processamento com *machine learning*. 38

Sumário

1 Introdução 9
1.1 Plataforma Consumidor.gov 10
1.2 Proposta 11
1.3 Objetivos 11
1.4 Organização do Trabalho 12

2 Fundamentação 13
2.1 Web Scraping 13
2.1.1 Métodos e Tecnologias 13
2.1.2 Considerações Legais e Éticas 14
2.2 Modelagem de Banco de Dados 14
2.3 ETL (Extração, Transformação, Carga) 15
2.3.1 Extração 16
2.3.2 Transformação 16
2.3.3 Carga 17
2.4 Machine Learning 18
2.4.1 Aprendizado Supervisionado 18
2.4.2 Aprendizado Não Supervisionado 18
2.4.3 Algoritmos Comuns 19

3 Metodologia 22
3.1 Ferramentas 22
3.2 A plataforma de dados abertos do Consumidor.gov.br 23
3.2.1 Extração 24
3.2.2 Transformação e Armazenamento 27

4 Resultados 28
4.1 Extração e Disponibilização da Base de Dados 28
4.2 Análise Descritiva 29
4.3 Aplicação de Machine Learning 36

5 Conclusão 39

Referências 40

1 Introdução

Atualmente, pesquisadores, cientistas de dados e muitos outros profissionais que trabalham com dados enfrentam grandes desafios ao tentar coletar dados públicos, automaticamente, e em escala a partir da *internet*. Isso acontece pois os dados fornecidos *online* muitas vezes são disponibilizados em formatos complexos e não estruturados, como células mescladas e larguras variáveis, além de muitos documentos estarem em formatos não digitais ou não pesquisáveis, como PDFs escaneados. (Lauren Stone, 2024). A posse de um grande volume de dados tem se tornado cada vez mais crucial em diversas áreas de pesquisa, como análise de redes sociais, estudos de mercado e inteligência artificial, tendo em vista o crescimento acelerado do mercado. Embora a *Web* ofereça uma vasta quantidade de informações úteis, a coleta automatizada de dados, conhecida como *web scraping*, apresenta obstáculos técnicos, legais e éticos que complicam o processo (Bhatt *et al.*, 2023).

De acordo com a lei nº 12.527/2011 da Constituição Federal Brasileira, qualquer pessoa física ou jurídica pode solicitar e receber informações públicas produzidas ou custodiadas pelos órgãos e entidades públicas. A Lei de Acesso à Informação também garante o direito de acesso às informações produzidas ou custodiadas pelas entidades privadas sem fins lucrativos que recebam recurso público para a realização de ações de interesse público. (GOV.BR, 2023).

Infelizmente, alguns sites do governo brasileiro ainda não disponibilizam essas informações de forma completa ou de forma clara, com a possibilidade de exportação massiva de dados. O ComprasNet, um site web instituído pelo Ministério do Planejamento, Orçamento e Gestão - MP, que promete disponibilizar à sociedade informações referentes às licitações e contratações promovidas pelo Governo Federal, bem como permitir a realização de processos eletrônicos de aquisição, não fornece uma interface amigável para o usuário nem as informações de forma íntegra, por exemplo. Já o site SICAR (Sistema Nacional de Cadastro Ambiental Rural), embora ofereça informações ambientais de relevância pública, o sistema restringe a exportação massiva de dados para análises, dificultando o acesso amplo para estudos científicos e fiscalização. Esses são apenas alguns exemplos de como os sites do governo brasileiro poderiam melhorar a disponibilidade e acesso às suas informações de utilidade pública, pois tornando as informações mais transparentes e de fácil acesso, facilita-se a análise, fiscalização e tomada de decisões por parte da sociedade, pesquisadores e gestores públicos, promovendo maior eficiência, participação cidadã e responsabilidade na gestão dos recursos e na implementação de políticas públicas.

A Suécia, por exemplo, foi pioneira na promoção da transparência pública, sendo o primeiro país do mundo a implementar uma lei de acesso à informação, em 1766, chamada de Lei de Liberdade de Imprensa. Essa legislação inovadora, que surgiu em um contexto de reformas democráticas, estabeleceu o direito dos cidadãos de acessar documentos governamentais, um marco na história da transparência e da responsabilidade pública. A iniciativa foi baseada no princípio de que a abertura de informações é essencial para combater a corrupção e fortalecer a democracia. Desde então, a Suécia tem se mantido na vanguarda da transparência, desenvolvendo plataformas modernas como o Open Data Sweden, que centralizam dados sobre economia, transporte, meio ambiente e outros setores, tornando-os acessíveis à população e incentivando a participação cidadã e a inovação tecnológica. O modelo sueco inspirou diversos outros países a adotarem legislações semelhantes ao longo dos séculos (Sweden Institute, 2024).

Entre os desafios técnicos, a heterogeneidade dos sites é um dos principais obstáculos. Cada site pode ser estruturado de maneira diferente, o que exige soluções personalizadas para extrair os dados. Além disso, muitas páginas utilizam tecnologias dinâmicas como JavaScript para carregar conteúdo, o que pode dificultar ou impedir a extração de dados com métodos simples. Ferramentas mais avançadas, como *headless browsers* ou APIs específicas, são frequentemente necessárias, mas podem aumentar a complexidade do processo (Gulbahar Karatas, 2024).

Outro desafio significativo é o bloqueio de scraping por parte dos sites. Muitas plataformas implementam medidas de proteção, como CAPTCHAs, limitação de requisições por IP (*rate limiting*), e verificações de comportamento suspeito, o que dificulta a coleta contínua e automatizada em larga escala. Além disso, a coleta de dados precisa ser eficiente para lidar com grandes volumes de informações sem sobrecarregar os servidores, exigindo infraestrutura robusta e otimizada.

Do ponto de vista legal, o acesso a dados da web nem sempre é permitido. Diversos sites têm termos de uso que proíbem explicitamente o *web scraping*. O não cumprimento dessas regras pode levar a problemas legais, incluindo o bloqueio de IPs e possíveis ações judiciais. Além disso, em algumas regiões, leis de proteção de dados, como o GDPR na Europa (GDPR, 2016) e a LGPD no Brasil (LGPD, 2018), impõem restrições

rigorosas sobre a coleta e uso de dados pessoais, exigindo que pesquisadores obtenham consentimento explícito para processar tais informações.

Os desafios éticos também são críticos. O uso de dados pessoais e informações sensíveis sem consentimento pode levantar questões sobre privacidade e transparência. Mesmo quando as informações são publicamente acessíveis, como em redes sociais, os indivíduos que publicam esses dados podem não estar cientes de que suas informações serão coletadas e analisadas em larga escala, o que cria dilemas sobre a utilização desses dados em estudos acadêmicos ou em produtos comerciais (IEEE RAS UFCG, 2021).

A coleta e a disponibilização de dados governamentais desempenham um papel igualmente crucial na era digital, oferecendo uma fonte confiável e abrangente de informações para pesquisadores, cientistas de dados e profissionais de diversas áreas. Dados governamentais incluem informações sobre saúde pública, educação, infraestrutura, meio ambiente, segurança e economia, entre outros tópicos, que são coletados, processados e organizados por instituições públicas. Quando disponibilizados de forma aberta e acessível, esses dados permitem análises detalhadas que auxiliam no desenvolvimento de políticas públicas mais eficientes, promovem a transparência e fortalecem a relação entre governo e sociedade (Lorena Sobreiro, 2023) (Nexos, 2024).

Por exemplo, cientistas de dados podem utilizar essas informações para criar modelos preditivos que antecipem surtos de doenças, identifiquem áreas prioritárias para investimentos ou analisem o impacto de políticas socioeconômicas em diferentes populações. Além disso, a abertura de dados públicos incentiva a inovação, possibilitando que *startups*, ONGs e pesquisadores desenvolvam soluções baseadas em evidências para problemas sociais complexos. Na ausência de dados governamentais acessíveis, muitos projetos de impacto global ficariam comprometidos, destacando a importância de plataformas que centralizem e organizem essas informações para uso público e acadêmico.

1.1 Plataforma Consumidor.gov

O Consumidor.gov.br é uma plataforma pública que facilita a comunicação direta entre consumidores e empresas para resolver conflitos de consumo pela internet. Supervisionado pela Secretaria Nacional do Consumidor (Senacon) do Ministério da Justiça e Segurança Pública, juntamente com Procons, Defensorias, Ministérios Públicos e a sociedade em geral, este serviço agiliza a resolução de conflitos de consumo de forma descomplicada: atualmente, 78% das reclamações registradas são resolvidas pelas empresas, que respondem aos consumidores em média em apenas 6 dias, de acordo com a pesquisa (Consumidor.gov.br, 2024).

Uma das vantagens do Consumidor.gov.br é a sua praticidade e eficiência. Por ser um canal online, os consumidores podem registrar suas reclamações a qualquer hora do dia, sem a necessidade de deslocamento físico até órgãos de defesa do consumidor. Além disso, o site promove a transparência nas relações de consumo, uma vez que as empresas são avaliadas pelos consumidores com base na qualidade do atendimento e na resolução dos problemas.

O Consumidor.gov.br promove relações mais transparentes entre consumidores, fornecedores e o Estado, baseado nos seguintes princípios:

- Transparência e controle social são imprescindíveis à efetividade dos direitos dos consumidores;
- As informações apresentadas pelos cidadãos consumidores são estratégicas para gestão e execução de políticas públicas de defesa do consumidor;
- O acesso a informação potencializa o poder de escolha dos consumidores e contribui para o aprimoramento das relações de consumo.

Por ser um serviço público, a participação das empresas no Consumidor.gov.br só é possível mediante adesão formal, comprometendo-se a resolver os problemas apresentados. Os consumidores, por sua vez, devem fornecer informações completas e se identificar corretamente ao registrar suas reclamações.

A Senacon gerencia e opera o Consumidor.gov.br, em colaboração com outros órgãos do Sistema Nacional de Defesa do Consumidor, para alcançar seus objetivos. A criação dessa plataforma está em conformidade com a Lei 8.078/1990 e o Decreto 7.963/2013.

O Consumidor.gov.br tem contribuído significativamente para a melhoria das relações de consumo no Brasil, proporcionando um meio ágil e acessível para a resolução de conflitos entre consumidores e empresas.

De acordo com os dados que ilustram a Figura 1.1, retirados da pesquisa (Consumidor.gov.br, 2023), 1.293.096 reclamações foram finalizadas em 2022 e 127 novas empresas cadastradas. Enquanto isso, no gráfico

da Figura 1.2, 1.385.840 reclamações foram finalizadas e 100 novas empresas foram cadastradas de acordo com a pesquisa de 2023 (Consumidor.gov.br, 2024).

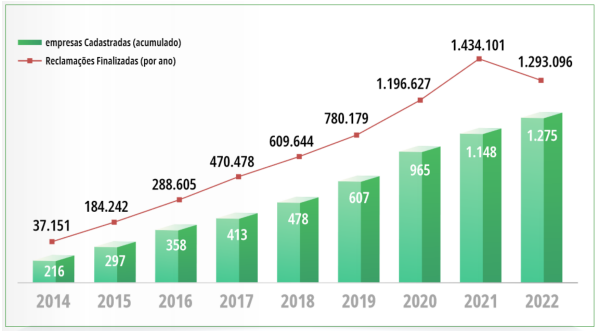
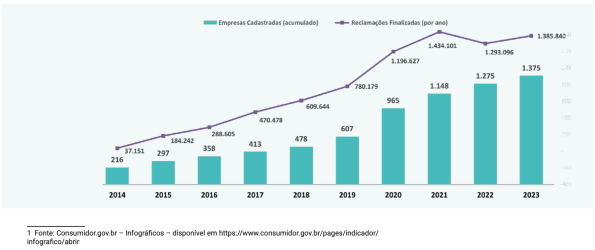


Figura 1.1 – Evolução da quantidade de empresas participantes e de reclamações finalizadas até 2022.



1 Fonte: Consumidor.gov.br – Infográficos – disponível em https://www.consumidor.gov.br/pages/indicador/infografico/abrir

Figura 1.2 – Evolução da quantidade de empresas participantes e de reclamações finalizadas até 2023.

Em suma, a alta taxa de resolução das reclamações e o tempo reduzido de resposta demonstram o impacto positivo da plataforma Consumidor.gov.br na melhoria das relações de consumo no Brasil, mas embora ela desempenhe esse papel essencial na facilitação de conflitos ao oferecer um canal eficiente e transparente para a comunicação direta entre consumidores e empresas, a plataforma deixa muito a desejar na parte da disponibilização das informações, pois os relatos dos consumidores não são disponibilizados para *download*, eles podem ser acessados apenas pelo site e, mesmo tendo um sistema de filtragem, os relatos antigos não conseguem ser recuperados pois o carregamento da página atinge *timeout*, o que dificulta o acesso e o processamento dessas informações públicas pelos interessados.

1.2 Proposta

Considerando o contexto dessa necessidade de extração de dados em escala, de forma automática para os pesquisadores e cientistas de dados, este trabalho visa realizar um *web scraping* no site https://consumidor.gov.br, que possui centenas de milhares de dados com muito potencial, para explorar uma possível maneira de realizar essa extração, a fim de facilitar o trabalho desses profissionais futuramente.

Para alcançar este objetivo, foram utilizados relatos de clientes e as suas respectivas respostas das empresas, que são disponibilizados ao público no https://consumidor.gov.br. Essas informações são disponibilizadas em páginas, sendo que cada página contém 10 cartões, mas esse formato que não é ideal, pois os pesquisadores precisariam clicar incessantemente no botão para carregar mais dados. Esses desafios serão melhor detalhados na seção 2, assim como a plataforma será melhor explicada na próxima seção.

1.3 Objetivos

Esse trabalho tem como objetivo implementar um processo de extração e processamento de relatos presentes no site https://consumidor.gov.br de forma automática, para que assim estejam disponíveis em um

banco de dados público, a fim de facilitar o acesso a qualquer um que deseje estudá-los. Também faz parte dos objetivos desse trabalho uma análise dos dados de forma descritiva, expondo as suas principais características, e processar os dados com um algoritmo clássico de *Machine Learning*, a fim demonstrar o potencial de aplicações para este banco de dados.

1.4 Organização do Trabalho

O restante desta monografia se organiza da seguinte forma: o Capítulo 2 provê uma visão global da teoria necessária para o entendimento do trabalho proposto, através de uma breve apresentação sobre *web scraping*, Modelagem de Banco de Dados, Extração, Transformação e Carga (ETL), e Machine Learning. O Capítulo 3 apresenta a metodologia usada nesse trabalho para a obtenção dos resultados, apresentados no Capítulo 4. Por fim, as conclusões e considerações para trabalhos futuros são elaboradas no Capítulo 5.

2 Fundamentação

Este capítulo apresenta os fundamentos teóricos que sustentam o desenvolvimento e a execução do presente trabalho, cujo objetivo é aplicar técnicas de *web scraping* para a extração de dados públicos disponíveis na internet. Inicialmente, são discutidos os conceitos básicos do *web scraping*, detalhando sua definição, métodos e ferramentas amplamente utilizadas. Em seguida, abordam-se os princípios de modelagem de dados e os processos de Extração, Transformação e Carga (ETL), fundamentais para o armazenamento e tratamento das informações extraídas. Por fim, será explorado o papel do machine learning no contexto deste trabalho, destacando seus principais algoritmos e como podem ser aplicados à análise e classificação dos dados coletados, permitindo a geração de insights a partir de grandes volumes de informações públicas.

A fundamentação teórica fornece, portanto, o embasamento necessário para compreender as etapas do projeto, bem como os desafios e limitações inerentes à prática de *web scraping* em ambientes governamentais. O foco está em contextualizar as tecnologias e práticas utilizadas, assegurando alinhamento com os objetivos propostos e com as diretrizes de transparência e ética no acesso a dados públicos.

2.1 Web Scraping

O *web scraping*, também chamado de web crawling, é uma técnica automatizada que utiliza softwares específicos para extrair dados de sites. Esse processo permite a coleta de informações diretamente do conteúdo HTML das páginas, facilitando o acesso a dados que, de outra forma, não estariam disponíveis em formatos legíveis por máquinas. Ao contrário da extração manual, que pode ser demorada e propensa a erros, o *web scraping* oferece uma solução eficiente, garantindo maior precisão, consistência e rapidez na coleta de grandes volumes de dados. Na era da informação, essa prática se torna cada vez mais essencial para empresas e pesquisadores, pois possibilita o acesso a informações valiosas de maneira automatizada e escalável, otimizando a análise de dados e a tomada de decisões. (Bhatt *et al.*, 2023)

O Web Scraping é utilizado em diversas áreas, como pesquisa de mercado, monitoramento de preços, coleta de dados para análise e até mesmo para treinamento de modelos de *machine learning*. Embora seja uma ferramenta poderosa para a coleta de informações, o web scraping pode ser complexo devido à variedade de estruturas de sites e à necessidade de respeitar as políticas de uso de cada página.

2.1.1 Métodos e Tecnologias

Existem diferentes métodos e tecnologias para realizar *web scraping*. Dentre os principais, destacam-se a raspagem com *scripts* de automação, em que se utiliza linguagens de programação como Python, com bibliotecas como Beautiful Soup (Pant *et al.*, 2024), que é uma biblioteca Python projetada para facilitar a extração de dados de arquivos HTML e XML. (Richardson, 2025) Ela fornece ferramentas simples e intuitivas para navegar e pesquisar no *parse tree* do documento, que é uma estrutura hierárquica que representa a organização sintática do documento, permitindo que os elementos, como *tags*, atributos e conteúdos, sejam acessados e manipulados de maneira estruturada. A biblioteca é especialmente útil quando se trata de manipular dados de forma pontual, permitindo que os usuários localizem elementos específicos no HTML por meio de seletores CSS ou métodos de busca baseados em *tags*. (Gunasekaran, 2023)

A Figura 2.1, retirada de (Gunasekaran, 2023), ilustra o funcionamento básico da biblioteca *Beautiful Soup* para manipulação de arquivos HTML. O processo começa com a entrada de um arquivo HTML, representado pelo `doc.html`. Esse arquivo é então processado pela biblioteca, em um estágio chamado *Soup Preparation*, que utiliza um analisador (como `'html.parser'`) para transformar o código HTML em uma estrutura navegável, conhecida como *parse tree*. Após essa preparação, o usuário pode realizar diferentes operações no documento, como localizar tags específicas (*Finding by tag names*), encontrar elementos por nomes de classes (*Finding by class names*), buscar texto completo (*Getting whole text*) ou extrair links através do atributo `href` (*Finding href*). Por fim, o código exemplifica a criação do objeto *Beautiful Soup* com o comando `soup = BeautifulSoup('doc.html', 'html.parser')`, destacando a simplicidade e eficiência dessa ferramenta para extração e manipulação de dados.

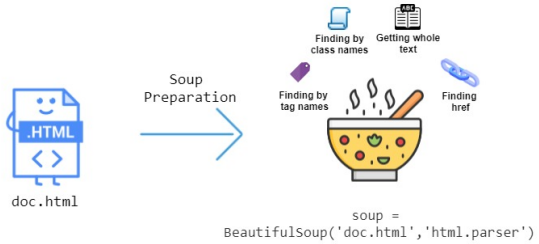


Figura 2.1 – Ilustração das funcionalidades de Beautiful Soup.

Existe, também, a biblioteca Scrapy (Kouzis-Loukas, 2016), que é um framework robusto e de alto desempenho para *web scraping* e crawling em Python. Ele é ideal para projetos que exigem a extração de grandes volumes de dados de múltiplas páginas da web. O Scrapy fornece uma estrutura completa, incluindo um sistema de gerenciamento de solicitações, suporte para pipelines de dados, e a capacidade de lidar com vários tipos de saída, como JSON ou CSV. A configuração do Scrapy é mais complexa em comparação com Beautiful Soup, mas, em troca, ele oferece recursos avançados, como a possibilidade de realizar scraping assíncrono, o que aumenta a eficiência na coleta de dados em sites que podem ter uma resposta lenta. Essas duas ferramentas permitem a navegação em sites, a análise do HTML e a extração de informações específicas, basta decidir qual usar em cada caso.

Além disso, para páginas dinâmicas que dependem de JavaScript para carregar seu conteúdo, ferramentas como o Selenium (Chapagain, 2019) desempenham um papel crucial, pois permitem que os usuários simulem interações humanas com um navegador. O Selenium possibilita a automação de tarefas, como clicar em botões, preencher formulários, filtrar pesquisas ou navegar por diferentes seções de um site, o que é fundamental para a extração de dados em ambientes onde o conteúdo não está imediatamente disponível ao carregar a página. Isso é especialmente útil em sites que exigem etapas interativas, como login, preenchimento de formulários ou navegação por menus dinâmicos, já que o Selenium pode reproduzir essas ações com precisão. Além de facilitar a extração de dados, essa ferramenta permite que os usuários lidem com desafios associados à renderização de conteúdo baseado em JavaScript, garantindo que as informações desejadas possam ser acessadas e extraídas com sucesso, mesmo em sites complexos e altamente interativos. Isso amplia significativamente as possibilidades de scraping, tornando possível coletar dados de um maior número de sites que dependem de funcionalidades interativas para exibir seus conteúdos. (Selenium; Raghavendra, 2021)

2.1.2 Considerações Legais e Éticas

Embora o *web scraping* ofereça grandes benefícios em termos de automação e coleta de dados, ele também levanta questões éticas e legais importantes (Khder, 2021). Muitas vezes, os sites estabelecem restrições claras em seus termos de uso, proibindo a extração automatizada de dados. Desrespeitar esses termos pode acarretar consequências legais, como processos por violação de direitos autorais ou de contrato. Além disso, é uma boa prática seguir as instruções do robots.txt, que é um arquivo de texto simples presente no diretório raiz de um site, orientando motores de busca e outros web crawlers sobre quais páginas ou seções do site eles podem ou não acessar e indexar segundo o autor (Jarmul, 2015). Este arquivo faz parte do Protocolo de Exclusão de Robôs (Robots Exclusion Protocol), um padrão que ajuda a gerenciar o comportamento de crawlers de forma amigável. Embora não seja um requisito legal, ignorar essas diretrizes pode resultar em sobrecarga de servidores e prejudicar a experiência dos usuários legítimos do site.

Outro ponto fundamental é o respeito às legislações de proteção de dados, como o GDPR (GDPR, 2016) na Europa e a LGPD (LGPD, 2018) no Brasil. Essas leis estabelecem limites claros sobre a coleta e o uso de dados pessoais, exigindo consentimento expresso dos indivíduos para o tratamento dessas informações. Coletar dados pessoais sem a devida autorização pode resultar em sanções severas, como multas significativas. (Capanema, 2020)

2.2 Modelagem de Banco de Dados

Os bancos de dados tornaram-se essenciais no cotidiano moderno, sendo utilizados em diversas atividades, como transações bancárias, reservas de viagens, consultas em bibliotecas e compras online. Essas

ações geralmente envolvem o acesso a bancos de dados para armazenar e recuperar informações textuais ou numéricas. Além disso, avanços tecnológicos recentes impulsionaram o desenvolvimento de aplicações inovadoras e diversificadas para sistemas de banco de dados (Elmasri *et al.*, 2005).

A modelagem de dados é o processo de criar uma representação visual que define como uma organização coleta, armazena e gerencia informações. Ela descreve os dados coletados, as relações entre eles e os métodos de armazenamento e análise. Essencial em um cenário onde grandes volumes de dados são coletados, a modelagem permite entender e organizar essas informações, orientando a escolha de tecnologias adequadas. Assim como um arquiteto projeta um esquema antes de construir, as organizações desenvolvem modelos de dados para reduzir erros, aumentar a eficiência na criação de bancos de dados, promover consistência na documentação e facilitar a comunicação entre equipes técnicas e de inteligência de negócios (AWS, 2024). A modelagem de banco de dados é fundamental para organizar e estruturar os dados coletados via *web scraping* de forma eficiente e acessível. Esse processo envolve a criação de um modelo lógico que define como os dados serão armazenados, relacionados e consultados (Khder, 2021).

Os bancos de dados relacionais organizam os dados em tabelas com linhas e colunas, como uma planilha na qual cada tabela representa uma entidade, como clientes, produtos etc. As relações entre essas entidades são definidas por chaves primárias e estrangeiras, que são elementos fundamentais na modelagem de banco de dados relacional. Uma chave primária é um identificador único para cada registro em uma tabela, garantindo que não haja duplicatas e permitindo que cada linha seja distinguida de forma exclusiva. Geralmente, ela é composta por uma ou mais colunas cujos valores são únicos e não nulos. Já uma chave estrangeira é um campo ou conjunto de campos em uma tabela que estabelece um vínculo com a chave primária de outra tabela, criando uma relação entre os dados armazenados. Essa associação permite integrar informações de diferentes tabelas, garantindo a consistência e integridade referencial do banco de dados. Por exemplo, em um sistema de uma clínica veterinária, a chave primária de uma tabela "Pacientes"pode ser referenciada como chave estrangeira na tabela "Tutores"para relacionar os animais aos seus tutores correspondentes (Silva, 2023).

Essa estrutura permite a realização de consultas complexas usando a Linguagem de Consulta Estruturada (SQL), garantindo que os dados sejam consistentes e integrados. Os bancos de dados relacionais seguem as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), o que significa que eles garantem a integridade dos dados mesmo em caso de falhas. Exemplos de bancos de dados relacionais incluem MySQL e Oracle (Jatana *et al.*, 2012).

Já os bancos de dados não relacionais, por outro lado, não utilizam a estrutura de tabelas tradicionais. Eles podem armazenar dados em formatos variados, como documentos, pares chave-valor, colunas ou grafos. Esses bancos de dados são frequentemente mais flexíveis, permitindo que os desenvolvedores armazenem dados sem um esquema rígido, o que é útil para aplicações que lidam com dados não estruturados ou em constante mudança. Eles também costumam ser escaláveis, permitindo que mais dados sejam adicionados facilmente sem a necessidade de reestruturar a base de dados. Como não seguem as propriedades ACID, mas sim as propriedades BASE (Basicamente Disponível, Estado Soft), eles podem ser mais rápidos e menos complexos em ambientes onde a velocidade de acesso e a flexibilidade são mais importantes que a consistência absoluta. O MongoDB é um exemplo de banco de dados não relacional (Jatana *et al.*, 2012).

No caso de dados extraídos da web, eles podem ser armazenados em bancos de dados relacionais, como MySQL ou Oracle, que utilizam tabelas e chaves para organizar os dados de forma estruturada, garantindo a integridade e a consistência das informações. Essa escolha é justificada por diversos motivos, como a capacidade dos bancos relacionais de lidar com grandes volumes de dados de forma eficiente, utilizando índices e consultas SQL para realizar buscas rápidas e precisas. Os bancos relacionais oferecem suporte maduro a transações, segurança robusta e são amplamente utilizados na indústria, o que proporciona uma grande quantidade de ferramentas e recursos de suporte para manipulação e análise dos dados extraídos da *web*.

2.3 ETL (Extração, Transformação, Carga)

Com o crescimento exponencial do volume de dados disponíveis em diferentes formatos e fontes, surge a necessidade de integrar e organizar essas informações de maneira eficiente para facilitar sua análise e utilização estratégica. Nesse contexto, o processo de ETL (*Extract, Transform, Load) desempenha um papel fundamental.* ETL é um processo de integração de dados que envolve três etapas principais: extração, onde os dados são obtidos de fontes externas, transformação, na qual os dados são adaptados para atender às necessidades do negócio, o que inclui sua limpeza e formatação de maneira uniforme, e carga, onde os dados transformados são inseridos em um *data warehouse*, que é um sistema de armazenamento de dados projetado para consolidar grandes volumes de informações provenientes de diversas fontes, organizando-os de maneira

estruturada para facilitar sua análise e consulta, para serem explorados em análises e relatórios. O processo ETL pode lidar com dados provenientes de diversas fontes, como aplicativos de *mainframe*, sistemas ERP, ferramentas de CRM, arquivos simples ou planilhas Excel. A tecnologia ETL resolve o problema da integração de dados ao garantir que os dados extraídos sejam preparados adequadamente antes de serem carregados no repositório final, permitindo que as empresas utilizem esses dados de forma eficiente para tomada de decisão e análises (Gour *et al.*, 2010).

O esquema geral dos processos de ETL é mostrado na Fig. 2.2, retirada de (El-Sappagh; Hendawi; Bastawissy, 2011). Os dados são extraídos de diferentes fontes e, em seguida, são propagados para o DSA (*Data Staging Area*, em português Área de Estágio de Dados), uma zona intermediária utilizada durante os processos de ETL para armazenar temporariamente os dados extraídos de diferentes fontes antes de serem transformados e carregados no *data warehouse* (DW). As fontes de origem, a área de estágio e os ambientes de destino podem ter diversos formatos de estrutura de dados, como arquivos simples, conjuntos de dados XML, tabelas relacionais, fontes não-relacionais, fontes de logs da web, sistemas legados e planilhas (El-Sappagh; Hendawi; Bastawissy, 2011).

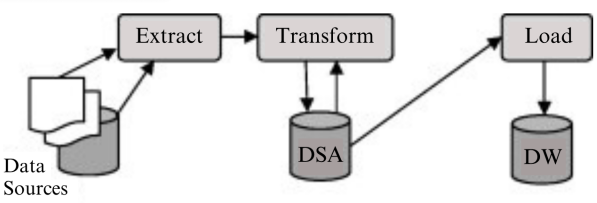


Figura 2.2 – Conceitos de Modelagem ETL.

2.3.1 Extração

A primeira etapa de qualquer cenário de ETL é a extração de dados. Esta fase é responsável por obter dados dos sistemas de origem. Cada fonte de dados possui características distintas que precisam ser gerenciadas para extrair os dados de forma eficaz. O processo deve integrar sistemas de diferentes plataformas, como diferentes sistemas de gerenciamento de banco de dados, sistemas operacionais e protocolos de comunicação. Durante a extração, a equipe de ETL deve estar ciente de como usar drivers ODBC/JDBC para se conectar às fontes de dados, entender a estrutura dos dados e lidar com fontes de diferentes naturezas, como mainframes. O processo de extração consiste em duas fases: a extração inicial e a captura de dados alterados. A extração inicial é realizada uma única vez, quando os dados são extraídos das fontes operacionais para serem carregados no data warehouse. A extração incremental, chamada de CDC (*Change Data Capture*, em português Captura de Dados Alterados), é um processo do ETL que identifica e captura as alterações realizadas nos dados de origem desde a última extração inicial. Essa fase atualiza o data warehouse com os dados modificados e adicionados, utilizando técnicas como colunas de auditoria, logs de banco de dados ou abordagens delta (El-Sappagh; Hendawi; Bastawissy, 2011).

2.3.2 Transformação

A segunda etapa do ETL é a transformação de dados. Esta fase envolve limpeza e conformidade dos dados recebidos para garantir que sejam precisos, completos, consistentes e sem ambiguidade. O processo inclui a limpeza dos dados, transformação e integração.

A transformação define o nível de detalhe das tabelas de fatos, tabelas de dimensão, esquemas do *data warehouse*, fatos derivados, dimensões que mudam lentamente e tabelas de fatos sem fatos.

As tabelas de fatos armazenam os dados quantitativos ou métricas do negócio. Elas contêm os "fatos"que são medidos e analisados, como vendas, lucros, ou quantidades, e são geralmente associadas a uma chave primária que faz referência às tabelas de dimensão, que contêm informações qualitativas ou descritivas sobre as entidades que estão associadas aos fatos. As dimensões são usadas para fornecer contexto às métricas armazenadas nas tabelas de fatos, como tempo, cliente, produto e região, por exemplo. Esquemas do *data warehouse* são a estrutura organizacional que define como os dados são armazenados e relacionados dentro do *data warehouse*. Os esquemas mais comuns incluem o esquema estrela (*star schema*), no qual uma tabela de fatos central é diretamente associada a várias tabelas de dimensão, e o esquema floco de neve (*snowflake*

schema), que normaliza as tabelas de dimensão para reduzir redundâncias, criando uma estrutura mais complexa e inter-relacionada entre as tabelas. Fatos derivados são calculados a partir de outros fatos ou dados. Em vez de serem coletados diretamente das fontes de dados, eles são derivados por meio de cálculos ou agregações, como médias, somas ou percentuais. Dimensões que mudam lentamente são dimensões cujos atributos podem mudar ao longo do tempo, mas com uma frequência baixa. Um exemplo seria a mudança de endereço de um cliente. Tabelas de fatos sem fatos, tabelas de fatos que não contêm métricas ou medidas numéricas diretamente, mas ainda assim têm um papel importante no processo de análise de dados. Elas podem ser usadas para registrar eventos ou transações, mas não possuem valores de medição, servindo mais como registros ou logs para referência futura (El-Sappagh; Hendawi; Bastawissy, 2011) (AWS, 2024).

A transformação dos dados extraídos via *web scraping* é um processo que vai além da simples coleta de informações, exigindo uma transformação dos dados para garantir que sejam armazenados de forma organizada, eficiente e acessível. Dados brutos extraídos de páginas web geralmente apresentam problemas como inconsistência de formatos, valores ausentes, duplicidade de informações ou falta de estrutura. Portanto, é necessário realizar uma série de etapas de processamento para preparar esses dados para armazenamento em um banco de dados (Aggarwal; Aggarwal, 2015).

É importante classificar os dados como quantitativos ou qualitativos. Dados quantitativos podem ser medidos e expressos em números. Eles se dividem em dois tipos: dados discretos, que assumem valores específicos e finitos, como o número de internações hospitalares de um paciente, e dados contínuos, que podem assumir qualquer valor dentro de uma faixa, como o peso de um paciente. Por outro lado, dados qualitativos não podem ser expressos numericamente. Eles são usados para classificar informações em categorias e podem ser divididos em dois tipos: nominais, quando as categorias não têm uma ordem natural, como a etnia, e ordinais, quando as categorias seguem uma sequência ordenada, como a gravidade da dor em uma escala. Cada valor de uma variável categórica é chamado de nível. A distinção entre dados quantitativos e qualitativos ajuda a determinar como os dados são tratados, analisados e interpretados em diferentes contextos (Pollard *et al.*, 2016).

Em seguida, os dados precisam ser limpos e padronizados. Isso inclui a remoção de caracteres indesejados, tratamento de campos em branco ou nulos, e unificação de formatos. Por exemplo, datas podem estar representadas em diferentes formatos (como "DD/MM/AAAA"ou "MM-DD-YYYY"), e a padronização desse tipo de dado é fundamental para garantir sua consistência e facilitar consultas futuras. Além disso, nomes de colunas, títulos ou chaves também devem ser uniformizados, assegurando que os dados sejam fáceis de identificar e manipular no banco (AWS, 2024).

Uma vez padronizados, os dados devem ser organizados de acordo com o modelo de banco de dados escolhido. Para bancos relacionais, isso significa distribuir os dados em tabelas inter-relacionadas, onde cada tabela representa uma entidade específica (por exemplo, "Usuários", "Produtos", "Transações"). Em contrapartida, em bancos de dados não relacionais como o MongoDB, os dados podem ser armazenados como documentos complexos que contêm coleções aninhadas de informações, oferecendo mais flexibilidade para dados semiestruturados.

Outro aspecto importante da estruturação é a definição de chaves primárias e estrangeiras (no caso de bancos relacionais), que são usadas para conectar diferentes tabelas e garantir a integridade referencial. Em bancos não-relacionais, essa estruturação pode envolver a definição de índices e agrupamento de dados que frequentemente são consultados juntos, a fim de melhorar a performance.

Além disso, é importante considerar como os dados serão atualizados e mantidos ao longo do tempo. No caso de *web scraping* contínuo ou recorrente, é necessário implementar mecanismos de verificação de duplicidade, para evitar armazenar repetidamente as mesmas informações. Isso pode ser feito através de verificações baseadas em identificadores únicos, como IDs de produtos ou URLs específicas. Em casos onde o scraping é feito periodicamente, os dados também podem precisar de uma estrutura que registre a data e hora da coleta, facilitando o rastreamento das mudanças ao longo do tempo.

2.3.3 Carga

A carga de dados na estrutura multidimensional alvo é a etapa final do ETL. Nessa fase, os dados extraídos e transformados são escritos nas estruturas dimensionais acessadas pelos usuários finais e sistemas de aplicação. A etapa de carga inclui tanto a carga das tabelas de dimensão quanto a carga das tabelas de fatos (El-Sappagh; Hendawi; Bastawissy, 2011).

2.4 Machine Learning

Machine learning (ML), ou aprendizado de máquina, é o estudo científico de algoritmos e modelos estatísticos que sistemas computacionais utilizam para executar uma tarefa específica sem serem explicitamente programados. Algoritmos de aprendizado são a base de muitas aplicações que usamos diariamente, como, por exemplo, toda vez que um motor de busca como o Google é usado para pesquisar na internet, um dos motivos pelo qual ele funciona tão bem é devido a um algoritmo de aprendizado que aprendeu como classificar páginas da web. Esses algoritmos são usados para várias finalidades, como mineração de dados, processamento de imagens, análise preditiva, entre outras. A principal vantagem de usar *machine learning* é que, uma vez que um algoritmo aprende o que fazer com os dados, ele pode realizar seu trabalho automaticamente. Os tipos de ML mais conhecidos são o Aprendizado Supervisionado e o Aprendizado Não Supervisionado, que serão abordados a seguir (Mahesh, 2020).

2.4.1 Aprendizado Supervisionado

Aprendizado supervisionado é a tarefa de *machine learning* que consiste em aprender uma função que mapeia uma entrada para uma saída com base em pares de entrada e saída fornecidos como exemplo. Ele cria uma função a partir de dados de treino rotulados, ou seja, com as respostas já definidas. Esses algoritmos precisam de ajuda externa para funcionar. Primeiro, os dados são divididos em dois conjuntos: um para treinar o modelo e outro para testar. No conjunto de treino, o modelo tem acesso à saída que deve ser prevista ou classificada. Os algoritmos aprendem padrões a partir desses dados e depois aplicam o que aprenderam nos dados de teste para se verificar o grau de acerto das previsões ou classificações (Mahesh, 2020).

O fluxo do aprendizado supervisionado é mostrado na Figura 2.3. Tudo começa com a coleta de dados. Esses dados são então organizados e divididos em dois conjuntos: dados para treino, que servem para ensinar o modelo a identificar padrões e relações, e dados para teste, reservados para avaliar como o modelo se comporta com informações que ele não viu durante o treinamento. O modelo é a representação matemática ou algorítmica que aprende a partir dos dados de treino, sendo treinado para minimizar erros e melhorar a precisão. Após o treinamento, o modelo é avaliado usando os dados de teste para medir sua capacidade de generalização. Caso o desempenho não seja satisfatório, são feitos ajustes, como otimização de hiperparâmetros, ajustes no algoritmo ou até melhorias nos dados. Quando o modelo atinge um desempenho aceitável, ele é implantado em produção, onde começa a ser usado em situações reais para fazer previsões ou tomar decisões. Mesmo após a implantação, ele pode ser monitorado e reavaliado para manter sua eficiência, principalmente quando novos dados são coletados.

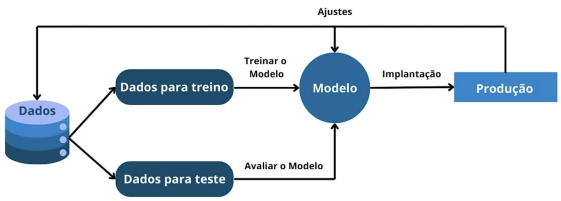


Figura 2.3 – Aprendizado Supervisionado.

2.4.2 Aprendizado Não Supervisionado

É chamado de aprendizado não supervisionado porque, ao contrário do aprendizado supervisionado, as respostas não são fornecidas durante o treinamento do modelo, ou seja, não há respostas corretas nem um "professor"guiando o processo. Os algoritmos são deixados "por conta própria"para descobrir e apresentar estruturas interessantes, do ponto de vista estatístico, nos dados. Os algoritmos de aprendizado não supervisionado buscam aprender algumas características estruturais dos dados. Quando novos dados são eventualmente introduzidos, eles usam as características aprendidas anteriormente para, por exemplo, reconhecer a classe dos dados. Esse tipo de aprendizado é principalmente utilizado para agrupamento (*clustering*) e redução de características (Mahesh, 2020).

As abordagens de aprendizado não supervisionado compartilham semelhanças em seus objetivos e estrutura com abordagens estatísticas que buscam identificar subgrupos não especificados com características semelhantes (por exemplo, variáveis ou classes "latentes"). Algoritmos de agrupamento, que agrupam observações com base em características de dados semelhantes (por exemplo, tanto laranjas quanto bolas de praia são redondas), são implementações comuns de aprendizado não supervisionado (Bi *et al.*, 2019). Esse tipo de aprendizado também pode ser útil para classificar os e-mails que recebemos na nossa caixa de entrada em spam, promoções, importantes etc.

2.4.3 Algoritmos Comuns

A seguir, serão explorados alguns dos algoritmos mais amplamente utilizados em *Machine Learning*, destacando suas características principais. Esses algoritmos são ferramentas fundamentais para resolver problemas como classificação, regressão, agrupamento e redução de dimensionalidade, sendo aplicados em diversas áreas, como reconhecimento de padrões, previsões financeiras, diagnósticos médicos e análise de dados.

Árvore de Decisão: Uma árvore de decisão é um grafo que representa escolhas e seus resultados na forma de uma árvore. Ela é composta por nós, que representam perguntas ou eventos, e ramificações (ou arestas), que indicam as regras ou condições que levam a decisões específicas. O nó raiz é o ponto inicial da árvore, geralmente uma pergunta que divide os dados em subconjuntos baseados em uma condição. Os nós internos representam atributos ou critérios usados para classificar os dados ou tomar decisões intermediárias, enquanto os nós folhas indicam os resultados finais ou as classes atribuídas após todas as condições serem avaliadas. Cada ramificação que conecta os nós representa um possível valor que o atributo pode assumir, guiando o processo de decisão até um resultado. Essa estrutura é intuitiva e eficiente, permitindo que decisões complexas sejam divididas em etapas menores e mais gerenciáveis. Além disso, a árvore de decisão é facilmente interpretável, tornando-se uma ferramenta valiosa tanto para prever resultados quanto para explicar o raciocínio por trás das decisões.

A Figura 2.4 mostra um exemplo de árvore de decisão. No caso deste exemplo, o objetivo é determinar se o convidado consumiu uma refeição vegetariana (*Veg*) ou não vegetariana (*Non-Veg*). O processo começa com a pergunta: "O convidado comeu frango?". Se a resposta for "Sim", ele é classificado como *Non-Veg*, caso a resposta seja "Não", a próxima pergunta é: "O convidado comeu carne de carneiro (*mutton*)?". Se a resposta for "Sim", ele também é classificado como *Non-Veg*, se a resposta for "Não", outra pergunta é feita: "O convidado comeu frutos do mar?". Se a resposta for "Sim", o convidado é classificado como *Non-Veg*, caso contrário, ele é classificado como *Veg*. Esse tipo de árvore reflete o funcionamento lógico de decisões baseadas em condições, onde cada nó interno representa uma pergunta, e os ramos levam a diferentes decisões ou classificações (Mahesh, 2020).

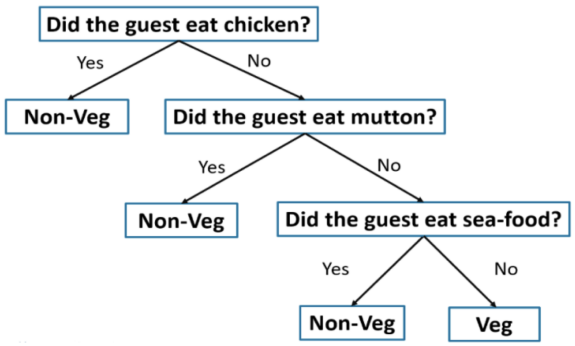


Figura 2.4 – Um exemplo de Árvore de Decisão.

Random Forest: O modelo Random Forest, desenvolvido por (Breiman, 2001), é um algoritmo altamente eficaz para tarefas de classificação e regressão, que combina várias árvores de decisão randomizadas. Cada árvore é treinada em uma amostra diferente dos dados e, em seguida, suas previsões são agregadas por média (em regressão) ou votação (em classificação). O método se destaca em situações em que o número de variáveis é muito maior que o número de observações e é altamente versátil, podendo ser aplicado a grandes conjuntos de dados, além de ser fácil de adaptar a diferentes tarefas de aprendizado. Entre seus principais

benefícios estão a capacidade de lidar com dados de alta dimensionalidade, amostras pequenas e a habilidade de medir a importância das variáveis, o que o torna uma ferramenta poderosa e amplamente utilizada em áreas como bioinformática, ecologia, e reconhecimento de objetos 3D. Sua implementação em pacotes como o ‘randomForest‘ no R e o Partial Decision Forests no Apache Mahout permite lidar com grandes volumes de dados, aproveitando o paralelismo para aumentar a eficiência (Biau; Scornet, 2016).

Máquina de Vetores de Suporte: Outra técnica de *machine learning* amplamente usada é a Máquina de Vetores de Suporte (SVM). Elas são modelos de aprendizado supervisionado que aplicam algoritmos para analisar dados e realizar tarefas de classificação e regressão, utilizando a maximização da margem de separação entre classes para melhorar a precisão do modelo. Além de realizar a classificação linear, as SVMs podem executar classificações não lineares de forma eficiente usando o chamado truque do kernel, que mapeia implicitamente as entradas em espaços de características de alta dimensão. Basicamente, elas traçam margens entre as classes. Essas margens são desenhadas de modo que a distância entre as margens e as classes seja a máxima possível, minimizando, assim, o erro esperado de classificação.

A Figura 2.5 de (Mahesh, 2020) ilustra o conceito de Máquinas de Vetores de Suporte. O objetivo do SVM é encontrar o hiperplano ótimo (a linha ou plano que divide o espaço em duas regiões) que separa as duas classes de dados (neste caso, representadas pelos círculos vermelhos e estrelas verdes) com a maior margem possível. Esse hiperplano é a linha central que divide as classes, enquanto as linhas paralelas próximas a ele definem a margem de separação, marcada pelo "Gap"na figura. Os pontos localizados mais próximos dessas margens, chamados de vetores de suporte, são cruciais para determinar a posição do hiperplano. Neste exemplo, os vetores de suporte estão destacados como pontos vermelhos e uma estrela verde mais próximos das margens. A SVM busca maximizar a distância entre essas margens, garantindo uma separação clara e robusta entre as classes, o que contribui para uma maior generalização ao lidar com novos dados (Mahesh, 2020).

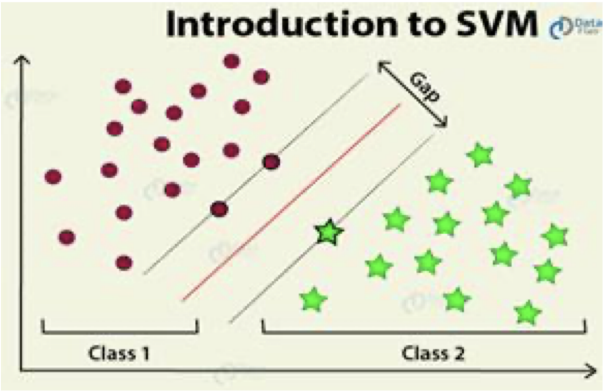


Figura 2.5 – Máquina de Vetores de Suporte.

Um outro exemplo de SVM é apresentado na Figura 2.6 de (Bi *et al.*, 2019) que mostra uma ilustração da transformação de dados com uma máquina de vetores de suporte para prever o status de diabetes. No gráfico A temos a distribuição hipotética de idade e índice de massa corporal (IMC; peso (kg)/altura (m)^2) de pacientes diabéticos (pontos pretos) e não diabéticos (pontos cinzas) em um espaço bidimensional. No gráfico A, a e b são parâmetros fixos estimados a partir dos dados e pode-se perceber que, neste mesmo gráfico, é visualmente difícil separar os pontos pretos dos pontos cinzas. Já no gráfico B, que mostra a situação em um plano tridimensional após a transformação, esses pontos/pacientes, que não são linearmente separáveis no espaço bidimensional, tornam-se linearmente separáveis em um espaço tridimensional. Um hiperplano no espaço tridimensional é mostrado como uma superfície horizontal (Bi *et al.*, 2019).

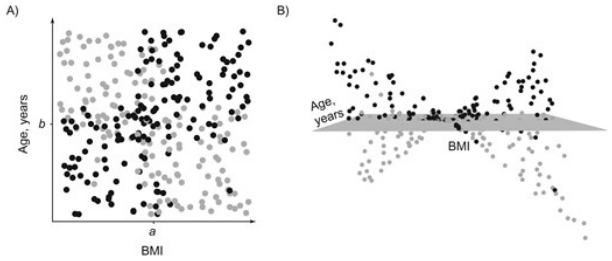


Figura 2.6 – Previsão do Status de Diabetes com SVM.

Agrupamento K-Means: K-means é um dos algoritmos de aprendizado não supervisionado mais simples, utilizado para resolver o conhecido problema de agrupamento (*clustering*), que consiste em organizar as informações em grupos, de maneira em que os dados dentro de cada grupo sejam mais semelhantes entre si do que com os dados de outros grupos. O procedimento começa com a definição de k centros, um para cada *cluster*, que é um grupo formado por elementos semelhantes dentro de um conjunto de dados, baseados em características ou proximidade no espaço multidimensional. Esses centros, chamados de centróides, são inicialmente posicionados de maneira estratégica, pois sua localização inicial pode influenciar os resultados finais. A ideia é escolher posições que estejam o mais distantes possível umas das outras, buscando cobrir diferentes regiões do espaço de dados. Em seguida, o algoritmo associa cada ponto de dados ao centróide mais próximo, formando os grupos iniciais. Após essa etapa, os centróides são recalculados como a média dos pontos em seus respectivos *clusters*, e o processo de associação e recalculação é repetido até que os centróides se estabilizem, ou seja, não mudem mais de posição. Essa abordagem iterativa permite agrupar os dados em conjuntos semelhantes de forma eficiente e organizada (Mahesh, 2020).

A Figura 2.7 de (Mahesh, 2020) mostra o antes e o depois do algoritmo K-Means. À esquerda, em "*Before K-Means*", observa-se um conjunto de pontos dispersos em um espaço bidimensional, representando dados não classificados ou agrupados. À direita, em "*After K-Means*", os mesmos pontos foram divididos em três grupos distintos, cada um destacado por uma elipse colorida, indicando os *clusters* formados. O algoritmo funciona iterativamente, atribuindo os pontos a *clusters* com base na proximidade ao centróide mais próximo e ajustando os centróides até que a separação dos grupos seja otimizada. Este processo organiza os dados em grupos semelhantes, facilitando a análise e interpretação.

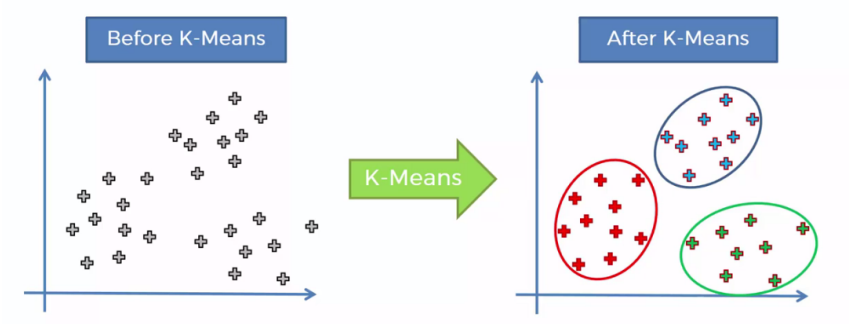


Figura 2.7 – Algoritmo K-Means.

3 Metodologia

Como explicado na introdução, esta monografia tem o objetivo de mostrar uma possível maneira, dentre muitas, de realizar uma extração automatizada e em escala de dados públicos disponíveis na internet. Esse projeto focou na extração das reclamações do site Consumidor.gov.br e, neste capítulo, a metodologia será abordada e explicada em detalhes, a fim de esclarecer quais passos foram seguidos até a obtenção dos resultados, que serão apresentados no capítulo 4. Este capítulo será dividido da seguinte forma: A seção 3.1 introduzirá as ferramentas utilizadas, a seção 3.2 contextualizará o site utilizado, a seção 3.3 discorrerá sobre a extração e os desafios enfrentados e a seção 3.4 falará sobre o processo de armazenamento.

3.1 Ferramentas

Algumas ferramentas foram utilizadas neste trabalho para possibilitar a extração dos dados. Nesta seção, elas serão apresentadas.

O código foi escrito com a linguagem Python (Python, 2024), por ser uma excelente opção para o tratamento de dados. Ela é uma linguagem simples e versátil, como explicado anteriormente no capítulo 2, que possui várias bibliotecas para o tratamento de dados e para a criação de gráficos, sem mencionar as bibliotecas para *Machine Learning*, que já vem com algoritmos prontos. Algumas das bibliotecas mais conhecidas são: Pandas (Pandas, 2025), NumPy (NumPy, 2025) e Matplotlib (Matplotlib, 2025).

Neste trabalho, a biblioteca usada para o ato da extração foi a *Beautiful Soup* (Richardson, 2025), que também foi introduzida no capítulo 2. Ela é uma ferramenta popular para extrair dados de arquivos HTML e XML em Python (Python, 2024), e é amplamente utilizada para web scraping, que é o processo de coletar dados de páginas da web. A *Beautiful Soup* cria um "*parse tree*", que é uma estrutura hierárquica que representa a organização sintática do documento, permitindo que os elementos, como *tags*, atributos e conteúdos, sejam acessados e manipulados de maneira estruturada, a partir de uma página HTML ou XML, facilitando a navegação, busca e modificação de dados. Ela é uma ferramenta adequada para iniciantes, por ser bastante intuitiva.

A ferramenta Selenium (Selenium, 2025) é uma ferramenta de automação de testes amplamente utilizada para aplicações web, e foi usada para permitir a simulação de interações de usuários em navegadores. Essa ferramenta permite que desenvolvedores e testadores escrevam *scripts* que imitam ações humanas, como clicar em botões, preencher formulários e navegar entre páginas. A ferramenta é composta por várias partes, incluindo o *Selenium WebDriver*, que fornece uma interface para controlar navegadores específicos, e o *Selenium Grid*, que permite a execução de testes em múltiplos ambientes simultaneamente. Suas características de flexibilidade, suporte a diversas linguagens de programação, como Python, Java e C, e sua compatibilidade com diversos navegadores e sistemas operacionais fazem do Selenium uma escolha popular para automação de testes.
O Selenium é uma ferramenta imprescindível para lidar com sites que utilizam JavaScript para gerar conteúdo dinâmico, pois ele permite a automação de interações e a extração de dados que seriam difíceis ou impossíveis de obter apenas com scraping tradicional, tornando-o essencial em muitas situações.

O MySQL (Oracle, 2025) foi adotado como sistema de gerenciamento de banco de dados. Ele é um sistema relacional, conforme explicado no capítulo 2, de código aberto, amplamente utilizado para armazenar e gerenciar dados em aplicações web e softwares de grande escala. MySQL utiliza a linguagem SQL (*Structured Query Language*) para a manipulação e consulta de dados, permitindo que usuários e desenvolvedores criem, leiam, atualizem e excluam informações de forma eficiente.

Uma das principais características do MySQL é sua escalabilidade, permitindo que ele suporte desde pequenas aplicações até grandes sistemas corporativos. Ele é conhecido por sua alta performance, confiabilidade e facilidade de uso, além de oferecer suporte a várias plataformas, incluindo Windows, Linux e macOS. O MySQL também suporta transações, o que garante a integridade dos dados, e permite a definição de diferentes níveis de acesso e permissões para usuários, tornando-o uma opção segura para o gerenciamento de informações sensíveis. E foi pela sua versatilidade que ele foi escolhido para esse projeto.

3.2 A plataforma de dados abertos do Consumidor.gov.br

Nesta seção, os passos para a extração e para o armazenamento dos dados do site serão explorados com maior profundidade. O primeiro passo para extrair as informações do site Consumidor.gov.br foi entender o seu funcionamento. A página inicial do site pode ser vista na imagem da Figura 3.1. A imagem mostra a tela inicial da plataforma que contém um campo para buscar empresas cadastradas, algumas informações sobre o site e alguns botões, dentre eles o botão "Últimas Reclamações", que é o botão que leva para a página que está na imagem da Figura 3.2, a página que mostra os cartões com os relatos dos consumidores e as respostas das empresas, sendo que um cartão é uma unidade que contém o nome da empresa, a data e o local da reclamação, o status da reclamação, o relato, a resposta da empresa e uma avaliação que é composta por uma nota, que varia de 1 a 5 e por um comentário opcional, como pode ser visto na Figura 3.3.

Figura 3.1 – Tela inicial do site Consumidor.gov.br.

Figura 3.2 – Página que exibe cartões com relatos de consumidores.

Figura 3.3 – Dados que compõem um cartão.

A seção do site de qual serão extraídas as informações se chama "Últimas Reclamações"e pode ser encontrada em (Reclamações, 2025). A página começa exibindo 10 cartões, e cada vez que o botão é clicado, no fim da página, mais 10 cartões são carregados.

Após o entendimento da estrutura dos cartões e do funcionamento do site, foi a vez de iniciar o processo de extração, e posteriormente, o armazenamento no banco de dados, que serão explicados mais detalhadamente a seguir.

3.2.1 Extração

Sites que geram páginas dinamicamente frequentemente utilizam tecnologias como JavaScript, AJAX e frameworks modernos, como React, Angular ou Vue.js, para criar conteúdos que são carregados no navegador em tempo real, após a página inicial ser renderizada. Isso significa que o HTML exibido ao usuário pode ser gerado e modificado após o carregamento inicial da página, dificultando o uso de ferramentas tradicionais de web scraping, como o *Beautiful Soup* (Richardson, 2025), que dependem do acesso ao HTML estático. Como essas ferramentas geralmente capturam o código HTML imediatamente após o carregamento da página, elas não conseguem acessar os dados que são carregados dinamicamente. Para contornar essa limitação, é necessário recorrer a técnicas mais avançadas, como a automação de navegadores com ferramentas como Selenium (Selenium, 2025), que simula a interação do usuário, permitindo a coleta de informações que são atualizadas após o carregamento inicial da página.

Foi exatamente essa situação que aconteceu quando esse projeto começou a ser desenvolvido. O primeiro obstáculo encontrado foi a falha da ferramenta Beautiful Soup ao tentar extrair informações do site. Após alguns testes e pesquisas, o Selenium entrou em cena e foi utilizado com sucesso, juntamente com o Beautiful Soup, para realizar as primeiras extrações. Em seguida, outro obstáculo foi encontrado na hora de utilizar o filtro presente no site, o qual pode ser visto na Figura 3.4, que mostra as opções de filtragem que a plataforma oferece, dentre elas a filtragem por período, que contém dois campos: "Data Inicial"e "Data Final".

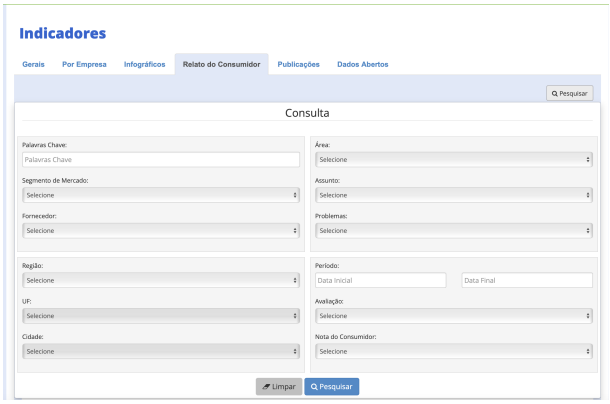


Figura 3.4 – Filtro presente no site Consumidor.gov.br.

Para que a extração pudesse ser realizada de forma automática e sem falhas, o filtro precisaria ser usado para que os cartões pudessem ser obtidos a partir das datas das suas publicações, o problema é que se constatou que o sistema do filtro não funcionava corretamente. Tal mecanismo deveria operar da seguinte forma: ao preencher o campo de data inicial e data final, a página carregaria 10 cartões em ordem decrescente de data, ou seja, começando pelos mais recentes e descendo até os mais antigos. No entanto, alguns problemas foram observados:

- Primeiramente o campo de "Data Inicial"não funciona como esperado, a pesquisa ocorre somente nos cartões pela data final. Então suponha que um pesquisador queira buscar as ocorrências que aconteceram durante as olimpíadas de Paris, entre os dias 24 de julho e 11 de agosto de 2024. Se ele colocar essas datas no filtro, este vai retornar 10 cartões começando pela data final (11 de agosto) e o pesquisador terá que clicar em um botão que carrega mais 10 cartões no fim da página até que chegue na sua data inicial (24 de julho). O problema é que a busca não se limita a essa data, ele consegue continuar descendo pela página e até mesmo consultar os cartões do dia 20 de julho, se ele continuar clicando no botão, por exemplo.
- O segundo problema aparece quando a pesquisa é realizada somente pela data final. Quando a pesquisa é lançada, surgem 10 cartões começando por essa data, mas existe a chance de alguns cartões, que também foram publicados nessa data, ficarem omitidos na página anterior, pois o filtro se preocupa apenas em apresentar uma página que comece com um cartão publicado na data final.

  A Figura 3.5 ilustra uma situação em que o usuário utiliza o filtro com a data final 20/10/24. Nessa imagem, os cartões que o usuário consegue ver estão representados pela cor branca e o cartão que não carrega está representado pela cor rosa. O que acontece é que os cartões apresentados ao usuário começam de fato pela data desejada (cartão branco superior), mas o usuário não consegue ver que havia mais cartões com essa data que não foram carregados porque ficaram na página anterior (cartão rosa).

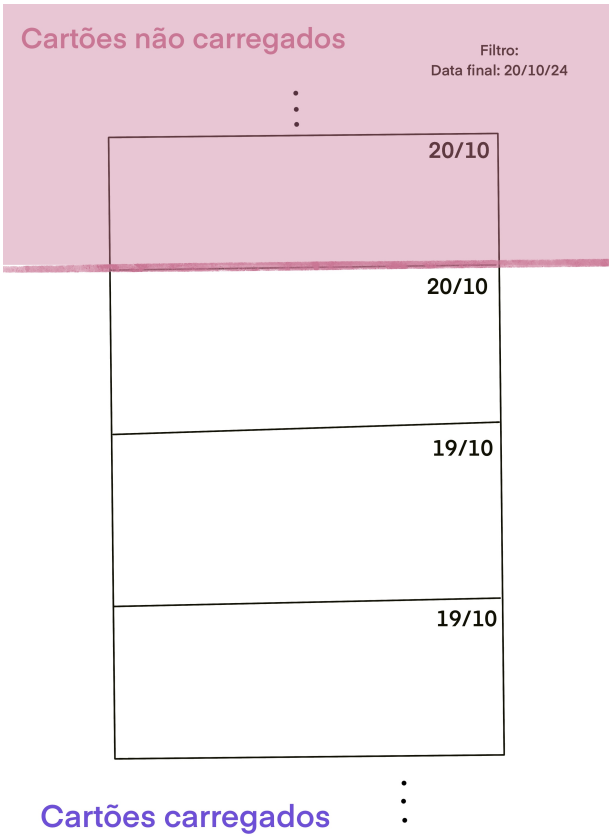


Figura 3.5 – Ilustração do carregamento da filtragem.

- O terceiro problema encontrado foi a velocidade de carregamento dos cartões após a filtragem. Quanto mais antiga a data final do filtro, mais tempo leva para carregar os resultados, podendo chegar ao ponto de nem sequer carregar, o *timeout* leva aproximadamente 5 minutos e a tela após o *timeout* pode ser vista na Figura 3.6. No teste feito para provocar o *timeout*, a data final inserida foi 03/04/2021.

Figura 3.6 – Tela após *timeout*.

Para tratar o segundo problema, o dos cartões com a data final que não carregarem e ficam omitidos (Figura 3.5), o código simplesmente coloca a data seguinte à data inserida pelo usuário para garantir que nenhum cartão fique para trás, ignorando os que contém a data D+1, para manter a coleta limpa, sem dados adicionais.

As informações puderam ser extraídas dos cartões de forma automática, através de uma implementação no código que pode ser encontrada no repositório deste projeto na plataforma GitHub [1].

[1] https://github.com/biamsarmento/relatos-consumidores

3.2.2 Transformação e Armazenamento

No código, os dados extraídos do site passam por várias etapas de formatação para garantir que estejam organizados e estruturados adequadamente antes de serem armazenados. Inicialmente, os dados são extraídos da página HTML utilizando o Selenium (Selenium, 2025) e o *Beautiful Soup* (Richardson, 2025). A cada iteração, o código acessa os elementos de interesse, como data, local, status, relato, resposta, nota e comentário de cada cartão, e realiza a manipulação dessas informações. Durante esse processo, as datas são convertidas para o formato *datetime* (DD/MM/YYYY), o que facilita a comparação com o intervalo de datas desejado. As informações extraídas, como os relatos e respostas, são reunidas em listas e convertidas em texto, removendo excessos e formatação desnecessária e alguns dados são separados uns dos outros, como a data do local, e a nota do comentário do consumidor.

Uma vez feita a rotina de formatação dos relatos, passamos para os *scripts* de armazenamento dos dados. O código foi escrito em Python (Python, 2024) e usa as bibliotecas mysql.connector (Corporation, 2025) para conexão com o banco de dados MySQL (Oracle, 2025), pandas (Pandas, 2025) para manipulação de dados CSV, e glob (Python Software Foundation,) para acesso e listagem de arquivos de um diretório específico.

O código responsável pelo web scraping separa os dados extraídos em semanas, ou seja, gera um arquivo CSV para cada semana dentro do intervalo especificado, facilitando a organização e o controle do pesquisador sobre os dados.

Quando os arquivos CSV estão todos dentro de uma pasta especificada pelo usuário, o *script* de armazenamento os adiciona ao banco de dados MySQL (Oracle, 2025), um banco relacional. A modelagem do banco foi feita com a criação da tabela, que contém os seguintes campos: id (chave primária auto-incrementada), empresa (nome da empresa), data (data da reclamação), local (local de origem da reclamação), status (status da reclamação), relato (texto do relato do consumidor), resposta (resposta da empresa), nota (nota atribuída ao atendimento) e comentario (comentário adicional do consumidor), com cada linha representando um cartão de reclamação. Além disso, foi implementada uma restrição de unicidade sobre os campos empresa, data, local e status, garantindo que não haja registros duplicados para o mesmo conjunto de dados.

É importante lembrar que os dados devem ser limpos e padronizados antes de serem colocados nos arquivos CSV. Nesse projeto alguns dados precisaram ser separados, como por exemplo a nota do comentário do consumidor. Essa parte é importante pois garante que os dados estejam em um formato adequado para uso. Também vale lembrar que a implementação de uma verificação de duplicidade de dados é fundamental, uma vez que dados duplicados podem reduzir a qualidade e a precisão da análise e também podem aumentar o custo e o tempo de processamento.

4 Resultados

Neste capítulo serão apresentados os resultados obtidos após as etapas explicadas na metodologia no capítulo 3 deste projeto. Na seção 4.1 o resultado do web scraping será apresentado, na seção 4.2 será apresentada uma análise descritiva dos dados obtidos, enaltecendo as suas principais utilidades e funções, e na seção 4.3, serão apresentados os resultados do processamento, no qual o algoritmo *Random Forest* foi utilizado em conjunto com técnicas de *machine learning* baseadas em aprendizado supervisionado, com o objetivo de treinar o modelo para prever a nota que um consumidor daria a um atendimento, considerando o status e o comentário do cartão.

4.1 Extração e Disponibilização da Base de Dados

Nesta seção serão apresentados os dados obtidos da extração automatizada que foi realizada no site Consumidor.gov.br.

Como dito anteriormente na seção 3.2.2, o programa recebe o intervalo de datas fornecido pelo usuário e o divide em semanas para facilitar o controle do pesquisador. Então, se o usuário quiser os cartões do mês de janeiro de 2023, por exemplo, o programa vai dividir esse intervalo em semanas e os arquivos CSV que o pesquisador receberá serão nomeados da seguinte maneira: dados_01-01-2023_a_07-01-2023, cada um com o intervalo especificado.

O código Python foi implementado e executado no Visual Studio Code (VS Code) (Microsoft, 2025) utilizando o interpretador Python 3.12. Também foi usada a extensão Python do VS Code para facilitar o desenvolvimento e depuração. No código, foram utilizados *timeouts* para gerenciar as interações com a página. Após a execução do filtro, há um *timeout* de 300 segundos para garantir que o botão "Pesquisar"desapareça, indicando que a pesquisa foi processada e durante o carregamento dos cartões, o botão "Mais Resultados"tem um *timeout* de 300 segundos para estar disponível e clicável, permitindo a exibição de mais registros. Esses *timeouts* garantem que o programa lide adequadamente com os tempos de resposta da página.

A duração do processo de *scraping* varia conforme o intervalo de dados desejado. Para intervalos dentro do último ano, a extração de dados equivalentes a uma semana leva cerca de 5 minutos, totalizando aproximadamente 25 minutos para um mês completo. No entanto, para períodos anteriores a um ano, o tempo de extração pode aumentar proporcionalmente à antiguidade dos dados, até o ponto de *timeout*, como explicado na seçao 3.2.1.

Antes de serem salvos no banco de dados, os dados passam por um processo de transformação e padronização para garantir que estejam limpos e no formato correto. Nessa etapa, alguns campos são separados, como "data e local", onde a data é convertida para o formato DD/MM/AAAA. O campo de avaliação também é dividido em dois: "nota", que varia de 1 a 5, e "comentário", que é opcional. Se o consumidor não inserir um comentário, a mensagem «não há comentários do consumidor>"será exibida no lugar.

Uma vez que todos os arquivos CSV desejados estiverem prontos, o pesquisador pode salvá-los no banco de dados. A Figura 4.1 ilustra o banco de dados e o número de cartões extraídos nesse projeto. No total foram extraídos 204.031 cartões, referentes a reclamações dos meses de julho a dezembro de 2022, e dos anos completos de 2023 e 2024. No total a base tem 335,9 MB de dados.

A base que contém todos os dados extraídos durante a execução desse projeto está disponível para *download* e pode ser acessada na plataforma Kaggle [1].

[1] https://www.kaggle.com/datasets/beatrizmsarmento/relatos-de-consumidores-do-site-consumidor-gov-br

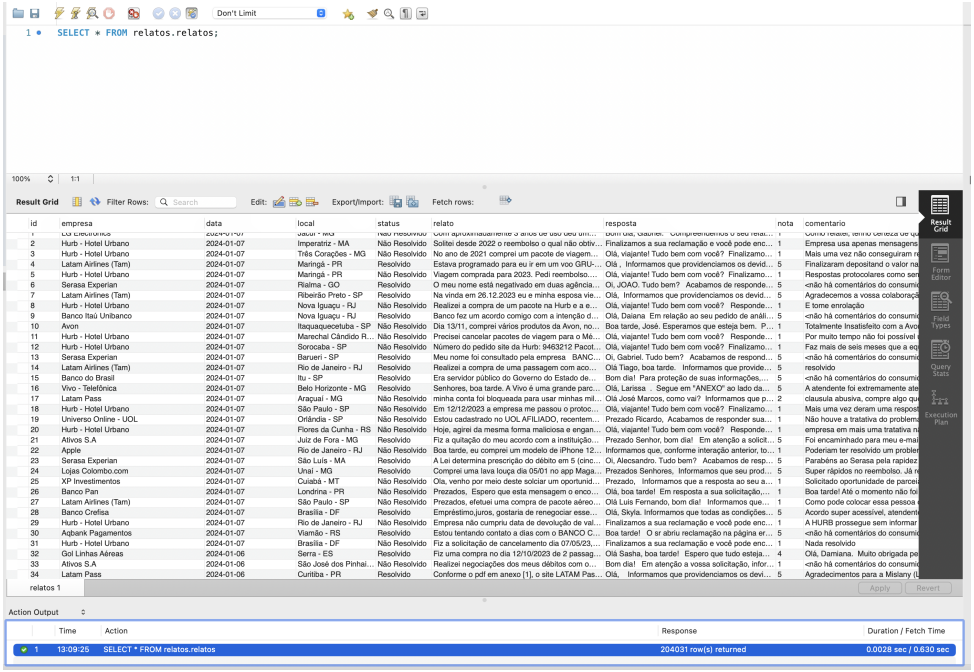


Figura 4.1 – Banco de Dados MySQL.

4.2 Análise Descritiva

Nesta seção os resultados das análises dos dados extraídos vão ser apresentados, dados esses que foram utilizados para gerar algumas estatísticas e análises interessantes.

Em primeiro lugar, serão apresentadas nuvens de palavras, que são as palavras que apareceram com maior frequência nos cartões estudados, dos relatos, respostas e dos comentários. A Figura 4.2 apresenta a nuvem de palavras dos relatos dos consumidores, a Figura 4.3 apresenta a nuvem das respostas das empresas participantes, e a Figura 4.4 traz a nuvem dos comentários dos consumidores, após terem recebido a resposta da empresa.

A análise da nuvem de palavras dos relatos na Figura 4.2 revela as principais palavras utilizadas nos relatos dos consumidores. Termos como "valor", "conta", "cartão", "compra"e "produto"se destacam devido ao seu tamanho e centralidade, indicando alta frequência de ocorrência. Isso sugere que grande parte das reclamações está relacionada a questões financeiras, produtos adquiridos e serviços de compra. Outras palavras como "dias", "sem", "problema", "reembolso"e "atendente"indicam preocupações comuns com prazos, atendimento e devoluções. Termos como "cancelamento", "prazo"e "recebido"reforçam que muitos relatos podem estar associados a atrasos, problemas com entregas ou dificuldades em cancelar pedidos ou serviços. Além disso, a análise das 10 palavras mais frequentes nos relatos revela que o termo **"valor"** aparece com 108.829 ocorrências, seguido de **"contato"** (78.165), **"compra"** (57.377), **"conta"** (53.991), **"cartão"** (43.115), **"banco"** (41.501), **"cancelamento"** (39.040), **"pedido"** (38.642), **"site"** (37.992), e **"recebi"** (37.982). Esses termos indicam que os consumidores frequentemente mencionam aspectos financeiros, processos de compra e desafios com os serviços oferecidos pelas empresas. Essa nuvem reflete um padrão de insatisfação comum em plataformas de reclamações, especialmente no que diz respeito a transações financeiras, suporte ao cliente e prazos. Para as empresas, esse tipo de análise é essencial para identificar áreas críticas a serem melhoradas.

A análise da nuvem de palavras das respostas das empresas na Figura 4.3 revela um padrão de comunicação institucional voltado para o atendimento ao cliente e o esclarecimento de dúvidas. Termos como "atendimento", "resposta", "atenção"e "disposição"aparecem em destaque, indicando que as empresas buscam demonstrar disponibilidade e preocupação com o consumidor. Outros termos relevantes, como "opção", "reclamação"e "interação", destacam a ênfase em oferecer alternativas ou reforçar a ideia de que as reclamações estão sendo tratadas. A presença de **"consumidor"** e **"cliente"** reforça o foco no relacionamento com o público. Além disso, a análise das 10 palavras mais frequentes nas respostas destaca o compromisso das empresas com a interação com o consumidor: **"atendimento"** (149.330), **"atenciosamente"** (120.234), **"resposta"** (95.577), **"equipe"** (75.553), **"opção"** (69.987), **"informamos"** (56.310), **"interação"** (55.067), **"disposição"** (53.876),

"nota" (50.274), e **"reclamação"** (49.222). Essas palavras indicam que as empresas focam em oferecer soluções rápidas e proativas, mas a repetição de termos como "atendimento"e "disposição"pode ser percebida como pouco personalizada, o que poderia ser melhorado.

A nuvem de palavras dos comentários dos consumidores na Figura 4.4 revela os principais aspectos abordados pelos consumidores em seus comentários sobre empresas e serviços. Um dos temas mais destacados é a busca por soluções, evidenciada por termos como "problema", "resolvido", "resolução"e "resolver". Isso demonstra que uma parte significativa das interações entre consumidores e empresas está relacionada à tentativa de solucionar dificuldades enfrentadas durante o uso de produtos ou serviços. Outro ponto de destaque é a relevância do atendimento. Palavras como "atendimento", "atendente"e "contato"refletem a importância que os consumidores atribuem à qualidade das interações com as equipes responsáveis. A análise das 10 palavras mais frequentes nos comentários também revela um foco significativo em aspectos de resolução de problemas e satisfação com o atendimento: **"atendimento"** (18.419), **"problema"** (17.942), **"resposta"** (14.752), **"valor"** (10.835), **"resolvido"** (9.949), **"agradeço"** (9.799), **"obrigado"** (9.430), **"contato"** (9.398), **"resolver"** (8.898), e **"reclamação"** (8.355). Essas palavras indicam que os consumidores estão interessados em soluções eficazes para seus problemas, e a qualidade do atendimento continua sendo uma prioridade. No geral, a nuvem de palavras aponta a importância de um atendimento ágil e empático, capaz de resolver problemas de forma eficiente e reforçar a confiança dos consumidores.

Essas análises, combinadas com as palavras mais frequentes de cada conjunto, oferecem *insights* importantes sobre a natureza das interações entre consumidores e empresas.

Figura 4.2 – Nuvem de palavras dos relatos.

Figura 4.3 – Nuvem de palavras das respostas.

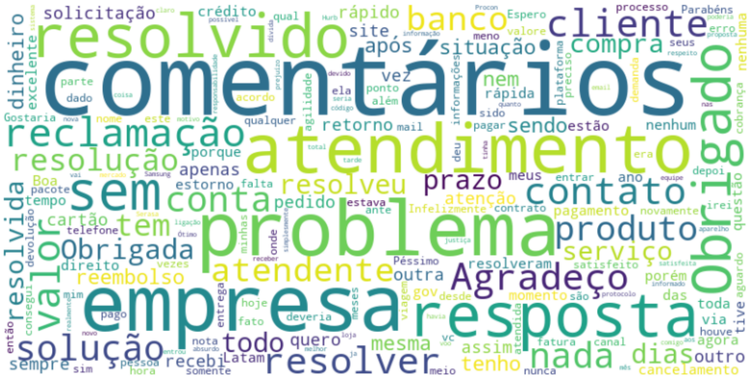


Figura 4.4 – Nuvem de palavras dos comentários.

Quando o atendimento é encerrado, o consumidor atribui para a reclamação o *status* "Resolvido"ou "Não Resolvido". Uma análise foi feita com esses dados para descobrir qual é a porcentagem exata de reclamações que tiveram o conflito solucionado. De acordo com a Figura 4.5, que ilustra essa informação, 120.599 reclamações foram de fato resolvidas pelas empresas, totalizando aproximadamente 59%, ou seja, um pouco mais da metade.

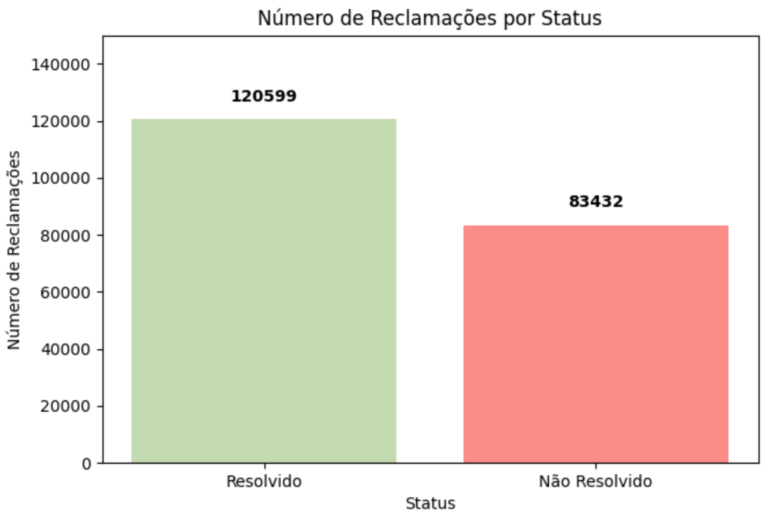


Figura 4.5 – Status de todos os relatos.

Também foi feita a análise do total de reclamações por mês de cada ano. A média de reclamações por mês em 2022 foi de 6.767,83, a de 2023 foi de 6.851,50 e a de 2024 foi de 6.767,17 reclamações, com um aumento de 1,24% de 2022 para 2023 e uma diminuição de 1,23% de 2023 para 2024. O gráfico da Figura 4.6 mostra o número de reclamações por mês para cada ano.

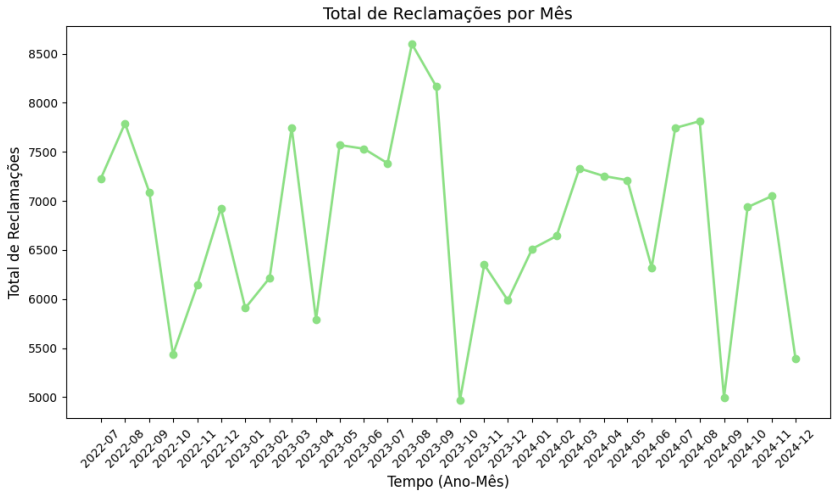


Figura 4.6 – Número de reclamações por mês.

Será analisado, agora, o número de reclamações que as 5 empresas com mais reclamações receberam por ano. Como pode ser visto na Figura 4.7, em 2022 as cinco empresas que mais receberam reclamações são majoritariamente do setor de serviços essenciais, como aviação e telecomunicações, com destaque para a Latam Airlines (Tam), que lidera com 2295 reclamações. Isso reflete uma possível insatisfação dos consumidores com os serviços oferecidos, seja por problemas operacionais, como cancelamentos e atrasos de voos, ou por questões relacionadas ao atendimento ao cliente. A Tim, com 1948 reclamações, também é um exemplo de uma operadora de telecomunicações que, frequentemente, enfrenta críticas sobre qualidade de serviço, cobertura e tarifas. Já as Linhas Aéreas Gol e Azul, com 1662 e 1452 reclamações, respectivamente, mostram que o setor aéreo continua sendo um dos mais suscetíveis a queixas, especialmente em relação à pontualidade e ao atendimento. Por fim, a Serasa Experian, com 1687 reclamações, destaca-se como uma empresa do setor financeiro, possivelmente associada a problemas de crédito e cobrança indevida. Esses dados revelam a alta demanda por melhorias no atendimento ao consumidor e na qualidade dos serviços, principalmente nas áreas de aviação e telecomunicações.

Na Figura 4.8 podemos ver que em 2023 as cinco empresas que mais receberam reclamações são lideradas novamente por empresas do setor de aviação e serviços financeiros, com um aumento significativo no número de queixas em comparação ao ano anterior. A Latam Airlines (Tam), com 5615 reclamações, registrou um aumento expressivo, evidenciando uma possível deterioração na qualidade do serviço ou uma maior insatisfação dos consumidores. A Hurb - Hotel Urbano, com 5535 reclamações, surge como um novo *player* no ranking, destacando-se no setor de turismo e viagens, o que pode indicar problemas com reservas, cancelamentos ou atendimento ao cliente. A Serasa Experian, com 3731 reclamações, segue em destaque, possivelmente refletindo questões no setor financeiro, como negativação indevida e dificuldades em resolver pendências. A Tim, com 2716 reclamações, continua sendo uma das principais operadoras de telecomunicações a ser criticada, possivelmente por questões de qualidade de serviço e cobrança. Por fim, a Azul Linhas Aéreas, com 2249 reclamações, também enfrenta uma quantidade considerável de queixas, embora em número menor que as demais. Esses dados apontam para uma tendência de crescimento das reclamações em vários setores, particularmente em aviação e serviços financeiros, destacando a necessidade de melhorias substanciais no atendimento e na qualidade dos serviços oferecidos.

A Figura 4.9 mostra que em 2024 as empresas que mais receberam reclamações mantiveram-se em sua maioria no setor de aviação e serviços financeiros, com algumas mudanças notáveis no ranking. A Latam Airlines (Tam) segue no topo, com 5510 reclamações, embora tenha apresentado uma leve queda em comparação com 2023, indicando que a empresa ainda enfrenta problemas recorrentes no atendimento ou na experiência do cliente. A Hurb - Hotel Urbano, com 5118 reclamações, continua a ser uma das mais reclamadas, refletindo possíveis dificuldades no setor de turismo, como problemas com reservas ou cancelamentos. A Serasa Experian, com 4155 reclamações, permanece como uma das líderes em queixas no setor financeiro. A Gol Linhas Aéreas, com 2186 reclamações, aparece em uma posição mais baixa em 2024, sugerindo que, apesar de ainda enfrentar críticas, a empresa pode ter implementado melhorias em seu serviço. Por fim, a Vivo - Telefônica, com 1764 reclamações, surge como uma nova empresa no top 5, refletindo as insatisfações dos consumidores com serviços de telecomunicação. Esses números continuam a evidenciar a necessidade

de um foco maior em melhorar o atendimento ao cliente, especialmente nos setores de aviação, turismo e telecomunicações.

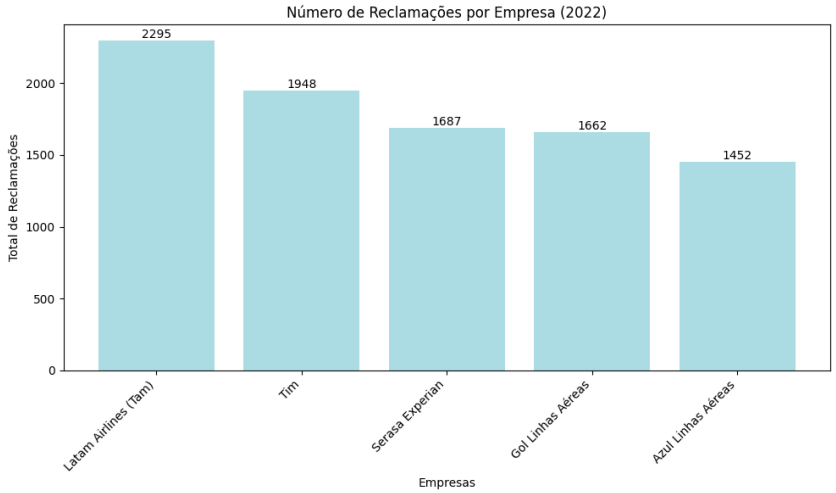


Figura 4.7 – As 5 empresas com mais reclamações em 2022.

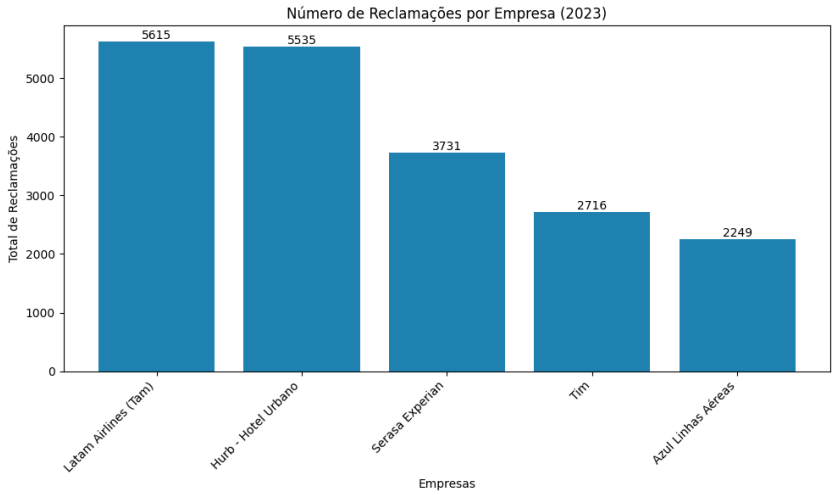


Figura 4.8 – As 5 empresas com mais reclamações em 2023.

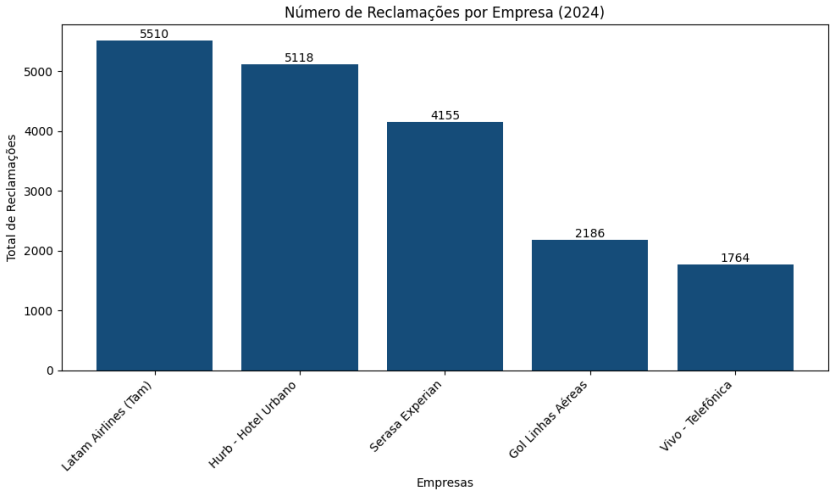


Figura 4.9 – As 5 empresas com mais reclamações em 2024.

Agora será analisado o percentual de reclamações que estão com o *status* 'Resolvido' dentre 3 empresas, a Latam Airlines (Tam) que está representada na Figura 4.10, o Hurb - Hotel Urbano que está representado na Figura 4.11 e a Serasa Experian que está representada na Figura 4.12. A análise dos percentuais de reclamações resolvidas das três empresas apresenta uma variação significativa nos índices de resolução. A Latam Airlines (Tam), embora seja a líder em número absoluto de reclamações, com 13.420 no total, destaca-se positivamente ao resolver 83,12% delas, o que indica um esforço considerável para atender às demandas dos consumidores. Em contraste, o Hurb - Hotel Urbano, com 11.756 reclamações, apresenta um desempenho preocupante, resolvendo apenas 5,92% das queixas. Esse baixo índice de resolução levanta questões sobre a capacidade da empresa em oferecer soluções eficazes para os problemas dos seus clientes. Já a Serasa Experian, com 9.573 reclamações, conseguiu resolver 59,79% delas, um percentual intermediário que, embora melhor do que o do Hurb, ainda deixa espaço para melhorias no atendimento e na resolução das demandas. Em resumo, enquanto algumas empresas, como a Latam, têm mostrado um compromisso considerável com a resolução das reclamações, outras, como o Hurb, ainda enfrentam sérios desafios nesse aspecto, o que impacta diretamente na satisfação do consumidor.

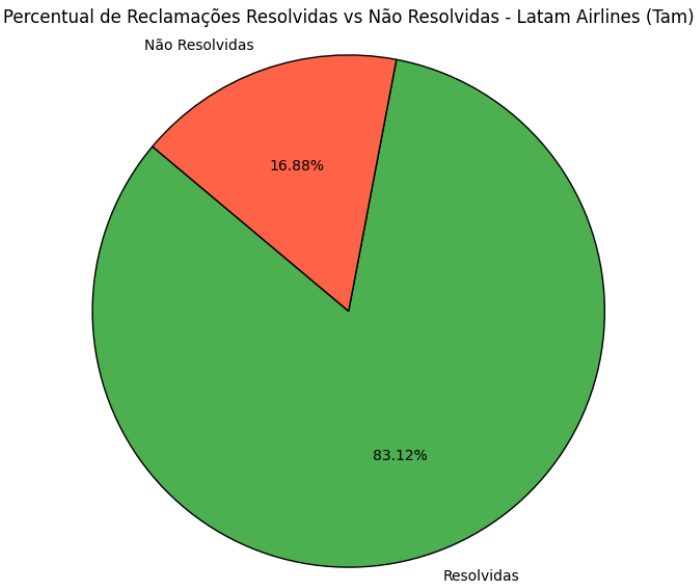


Figura 4.10 – Percentual de reclamações resolvidas pela Latam.

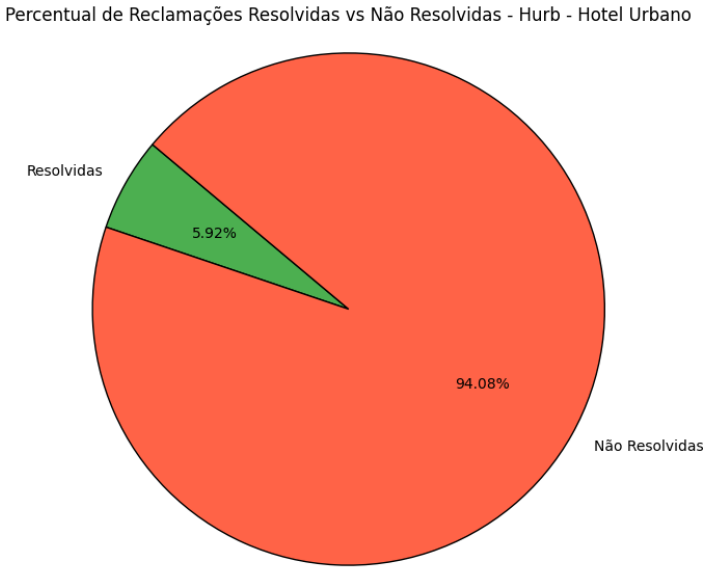


Figura 4.11 – Percentual de reclamações resolvidas pelo Hurb - Hotel Urbano.

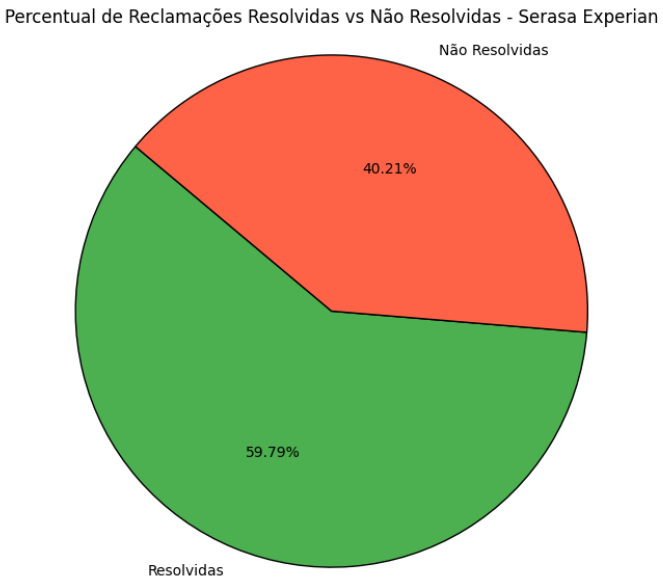


Figura 4.12 – Percentual de reclamações resolvidas pela Serasa Experian.

4.3 Aplicação de Machine Learning

Nesta seção será apresentado o resultado de uma aplicação de *machine learning*, utilizando a biblioteca scikit-learn (Pedregosa *et al.*, 2011) e o algoritmo Random Forest, com o objetivo de demonstrar mais um possível uso desse projeto.

Primeiramente deve-se separar as reclamações em dois conjuntos, 80% para treinar o modelo e 20% para testá-lo. O modelo será treinado para usar as colunas de Comentário, que contém um texto, e Status, que pode ser "Resolvido"ou "Não Resolvido", para tentar classificar qual será a nota que o consumidor vai dar para o atendimento, essa podendo variar de 1 a 5.

Os dados são carregados de um arquivo JSON e então pré-processados pelo *script*: o Status resolvido/-Não resolvido é convertido os valores 1/0, respectivamente, e as notas para inteiros, facilitando a manipulação dos dados. O *script*, que pode ser encontrado no repositório deste projeto na plataforma GitHub [2], também lida com a situação em que o consumidor não deixa um comentário. Nesse caso, aparece uma mensagem exatamente assim: '<não há comentários do consumidor>', e essas mensagens são ignoradas.

Stopwords também são usadas para remover palavras comuns e irrelevantes na análise de texto, como "de", "a", "o". Elas foram aplicadas no TF-IDF (Term Frequency-Inverse Document Frequency), que será explicado no próximo parágrafo, para filtrar palavras antes da transformação dos comentários em uma matriz numérica, reduzindo ruído e melhorando a classificação. A lista de *stopwords* contém 221 palavras.

Também é feita a vetorização dos dados, que é o processo de converter textos em dados numéricos para que possam ser processados por modelos de aprendizado de máquina. No código, a técnica TF-IDF (Term Frequency-Inverse Document Frequency), que calcula a relevância de cada palavra em um texto com base na frequência de sua ocorrência, é utilizada. O TF (Term Frequency) mede quantas vezes uma palavra aparece em um documento, enquanto o IDF (Inverse Document Frequency) reduz o peso de palavras muito comuns no conjunto de textos e destaca as menos frequentes. O resultado é uma matriz numérica onde cada linha representa um comentário, cada coluna representa uma palavra única, e os valores indicam a importância relativa das palavras. Essa matriz é usada como entrada para o modelo de previsão, junto com a variável *status* binarizada.

[2] https://github.com/biamsarmento/relatos-consumidores

O modelo de classificação Random Forest é utilizado, então, para aprender a prever as notas dos consumidores. A seleção dos hiperparâmetros no código, tanto o número de árvores quanto a profundidade máxima das árvores, é feita por meio de uma busca em grade (*GridSearch*). A grade de parâmetros testados inclui valores de número de árvores de 25, 50, 100 e 200, e valores de profundidade máxima de 5, 10, 20, 40 e *None* (sem limite de profundidade). O processo utiliza validação cruzada com 5 divisões e tem como métrica de avaliação a acurácia. Durante a busca, a busca em grade treina um modelo *RandomForestClassifier* para cada combinação de parâmetros e seleciona a melhor configuração com base no desempenho nos dados de validação. Ao final, os melhores hiperparâmetros encontrados são armazenados, e o modelo otimizado é utilizado para realizar previsões no conjunto de teste. Além disso, são identificadas as palavras e características que mais influenciam nas previsões, ajudando a entender como o modelo toma suas decisões.

Após o processamento, o *script* identificou que o texto padrão '<não há comentários do consumidor>' apareceu 58.729 vezes no conjunto de dados, aproximadamente 28,79% do total, o que é um número considerável. A análise revelou uma distribuição desbalanceada das notas, com a nota 1 sendo a mais frequente (44,39%), seguida pela nota 5 (40,38%). Notas intermediárias foram menos comuns: a nota 4 representou apenas 6,26% dos dados, enquanto as notas 3 e 2 tiveram proporções ainda menores, de 5,56% e 3,41%, respectivamente. Os dados foram divididos em conjuntos de treino e teste, contendo 116.117 amostras para treino e 29.030 amostras para teste, e abrangeram cinco classes distintas, que correspondem às notas que os consumidores atribuíram ao atendimento, que podem variar de 1 a 5.

Uma busca de hiperparâmetros foi realizada para otimizar o modelo Random Forest, que apresentou como melhores parâmetros profundidade máxima como 40 e número de estimadores como 200. O modelo resultante alcançou uma acurácia geral de 80,79%, mas a acurácia balanceada foi significativamente menor (38,23%), evidenciando dificuldades em lidar com o desbalanceamento das classes. A análise das *features* mais importantes destacou a variável 'status', com um peso de 42,85%, como a mais relevante, seguida por várias características do vetor TF-IDF, como as colunas 295 (4,49%), 25 (2,64%), e 60 (2,35%), que correspondem às palavras 'obrigado', 'agradeço' e 'atendimento', respectivamente.

A matriz gerada pelo TfidfVectorizer é uma representação numérica dos textos contidos nos comentários dos consumidores, onde cada linha corresponde a um comentário e cada coluna representa a importância de um termo específico. No caso desse projeto, foi definido no *script* que a matriz teria 500 colunas, limitando o número de características extraídas a esse número de palavras mais relevantes. O número de linhas da matriz é determinado pelo número de comentários no conjunto de dados de treino, que possui 116.117 amostras. Assim, a matriz gerada tem 116.117 linhas (um comentário por linha) e 500 colunas (um termo por coluna). Cada valor na matriz é o peso associado ao termo para aquele comentário específico, e a importância de cada termo é calculada levando em conta sua frequência no documento e a raridade no conjunto de documentos. Essa matriz é utilizada como entrada para o modelo de aprendizado de máquina, que pode então analisar a relevância de cada termo para classificar os dados, como a nota dada pelos consumidores nesse caso.

A matriz de confusão apresentada na Figura 4.13 mostra a distribuição das previsões do modelo em relação às classes reais. As linhas representam as classes reais dos dados, enquanto as colunas indicam as classes previstas pelo modelo. Os valores na diagonal principal correspondem às previsões corretas, enquanto os valores fora da diagonal indicam erros de classificação. Observa-se que as classes 1 e 5 apresentam a maior quantidade de acertos, enquanto as classes intermediárias (2, 3 e 4) possuem um número significativo de erros, sendo frequentemente confundidas com a classe 5. Isso sugere que o modelo pode ter dificuldades em distinguir algumas categorias, possivelmente devido a um desbalanceamento nos dados ou semelhanças entre essas classes. Melhorias podem ser feitas utilizando técnicas como balanceamento dos dados ou modelos mais robustos para reduzir os erros de classificação.

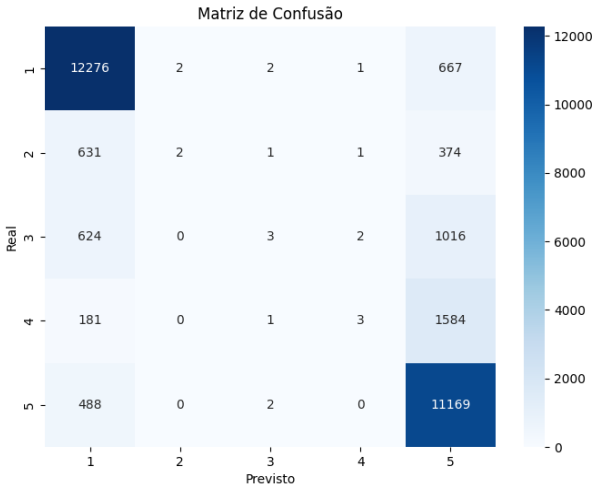


Figura 4.13 – Matriz de confusão do processamento com *machine learning*.

A execução do código levou aproximadamente 9,6 minutos. Essa análise mostra tanto a eficiência do modelo em prever classes majoritárias quanto os desafios impostos pelo desbalanceamento das classes.

Esse é apenas um dos possíveis usos desse projeto, e ilustra a grande importância da coleta de dados para o mundo atual.

5 Conclusão

Na era da informação, a quantidade de dados disponíveis online cresce exponencialmente, tornando essencial o desenvolvimento de ferramentas capazes de coletar, organizar e analisar essas informações de forma eficiente. Nesse contexto, o *web scraping* surge como uma solução prática e poderosa para extrair dados estruturados a partir de fontes não estruturadas disponíveis na web. Este trabalho explora o uso dessa técnica aplicada ao portal Consumidor.gov.br, uma plataforma pública voltada à resolução de conflitos entre consumidores e empresas que, apesar de seu potencial, não disponibiliza os dados de maneira acessível para download massivo.

O problema central investigado foi como automatizar a coleta, o processamento e a análise de grandes volumes de dados provenientes de reclamações registradas por consumidores, com o objetivo de gerar *insights* úteis que possam beneficiar empresas na melhoria de seus serviços, governos na formulação de políticas públicas e os próprios consumidores em suas decisões de consumo.

Os objetivos estabelecidos incluíram, primeiramente, a implementação de um sistema de extração automatizada de dados. O desafio consistiu em contornar as barreiras impostas pela estrutura dinâmica do site, o que exigiu o uso de tecnologias como o Selenium (Selenium, 2025) e o Beautiful Soup (Richardson, 2025) para acessar e capturar os dados de maneira contínua e escalável. Como resultado, o sistema foi capaz de extrair mais de 200 mil registros de reclamações, consolidando dados de múltiplas empresas e consumidores em diferentes períodos, provando ser eficaz na coleta automatizada em larga escala.

O segundo objetivo foi o armazenamento desses dados em um banco de dados estruturado. A escolha do MySQL (Oracle, 2025), um banco de dados relacional, permitiu organizar as informações extraídas de maneira eficiente, utilizando tabelas que representavam entidades essenciais, como "id", "empresa", "data", "local", "status", "relato", "resposta", "nota"e "comentario". Essa estruturação facilitou tanto o acesso quanto a análise posterior dos dados, proporcionando uma base sólida para a realização de consultas e visualizações, além de manter a integridade e consistência das informações.

Por fim, o terceiro objetivo foi a aplicação de técnicas de *machine learning* para explorar as informações contidas nos relatos dos usuários. Para isso, foram implementados modelos de aprendizado supervisionado com o intuito de "ensinar a máquina"a classificar as notas dos atendimentos das empresas com base nas características dos status e dos comentários. O algoritmo utilizado, o Random Forest, mostrou-se eficaz em identificar padrões relevantes nos dados, permitindo a classificação automática de novos registros. No entanto, um desafio significativo foi o desbalanceamento das classes, que são as notas de 1 a 5 que o consumidor escolhe para avaliar o atendimento, o que prejudicou a precisão do modelo em certos casos.

Portanto, com base nos resultados, é possível afirmar que os objetivos propostos foram, em grande parte, alcançados. A extração e o armazenamento automatizados funcionaram conforme esperado, e as análises com *machine learning* revelaram *insights* importantes, apesar das limitações encontradas no balanceamento de classes. Esses desafios abrem espaço para futuros aprimoramentos no tratamento dos dados e no desempenho dos modelos, talvez explorando técnicas de balanceamento de classes ou modelos mais avançados, como redes neurais.

Além disso, um possível aprimoramento seria a substituição da biblioteca Beautiful Soup pela biblioteca Scrapy, que pode proporcionar um desempenho superior na extração de dados, especialmente em projetos mais complexos. Outra linha interessante para estudos futuros seria a análise da probabilidade de que a aviação aérea continue a ser o setor líder de reclamações no ano de 2025.

Por fim, este trabalho abre portas para o desenvolvimento de sistemas mais robustos que utilizem web scraping para monitoramento e análise de dados, contribuindo tanto para a melhoria dos serviços prestados por empresas quanto para o fortalecimento de políticas públicas voltadas ao consumidor. Além disso, uma futura migração para o MongoDB, no lugar do MySQL, poderia proporcionar uma maior flexibilidade no armazenamento e processamento de grandes volumes de dados, especialmente em um cenário de crescimento contínuo da base de dados.

Referências

AGGARWAL, C. C.; AGGARWAL, C. C. Data preparation. **Data Mining: The Textbook**, Springer, 2015. Citado na p. 17.

AWS. **O que é ETL (extrair, transformar e carregar)?** 2024. Disponível em: https://aws.amazon.com/pt/what-is/etl/?nc1=h_ls. Citado na p. 17.

AWS. **O que é modelagem de dados?** 2024. Disponível em: https://aws.amazon.com/pt/what-is/data-modeling/. Citado na p. 15.

BHATT, C.; BISHT, A.; CHAUHAN, R.; VISHVAKARMA, A.; KUMAR, M.; SHARMA, S. Web scraping techniques and its applications: A review. **2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT)**, 2023. Citado nas pp. 9 e 13.

BI, Q.; GOODMAN, K. E.; KAMINSKY, J.; LESSLER, J. What is machine learning? a primer for the epidemiologist. **American journal of epidemiology**, Oxford University Press, v. 188, 2019. Disponível em: https://sph.umsha.ac.ir/uploads/18/2023/May/22/kwz189.pdf. Citado nas pp. 19 e 20.

BIAU, G.; SCORNET, E. A random forest guided tour. **Test**, Springer, 2016. Citado na p. 20.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, 2001. Citado na p. 19.

CAPANEMA, W. A. A responsabilidade civil na lei geral de proteção de dados. **Cadernos Jurídicos, São Paulo, ano**, 2020. Citado na p. 14.

CHAPAGAIN, A. **Hands-On Web Scraping with Python: Perform advanced scraping operations using various Python libraries and tools such as Selenium, Regex, and others**. [*S.l.*]: Packt Publishing Ltd, 2019. Citado na p. 14.

CONSUMIDOR.GOV.BR. Consumidor.gov.br. Disponível em: https://consumidor.gov.br. Citado na p. 6.

CONSUMIDOR.GOV.BR. Boletim consumidor.gov.br 2022. p. 23, 2023. Disponível em: https://www.gov.br/mj/pt-br/assuntos/noticias/dia-do-consumidor-senacon-lanca-boletins-com-os-dados-de-reclamacoes-recebidas-em-2
15-03-2023-boletim_consumidor-gov-br_2022_v6.pdf/view. Citado na p. 10.

CONSUMIDOR.GOV.BR. Boletim consumidor.gov.br 2022. p. 23, 2024. Disponível em: https://www.gov.br/mj/pt-br/assuntos/noticias/11.03.2024PDFBoletimConsumidor.gov.br2023_final_compressed3.pdf. Citado nas pp. 10 e 11.

CORPORATION, O. **MySQL Connector: The official MySQL Database Connector**. 2025. Disponível em: https://www.mysql.com/products/connector/. Citado na p. 27.

EL-SAPPAGH, S. H. A.; HENDAWI, A. M. A.; BASTAWISSY, A. H. E. A proposed model for data warehouse etl processes. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, v. 23, 2011. Citado nas pp. 16 e 17.

ELMASRI, R.; NAVATHE, S. B.; PINHEIRO, M. G. *et al.* Sistemas de banco de dados. Pearson Addison Wesley São Paulo, 2005. Citado na p. 15.

GDPR. **General Data Protection Regulation (GDPR)**. 2016. https://gdpr-info.eu. Citado nas pp. 9 e 14.

GOUR, V.; SARANGDEVOT, S.; TANWAR, G. S.; SHARMA, A. Improve performance of extract, transform and load (etl) in data warehouse. **Int. Journal on Comp. Sci. and Eng**, v. 2, 2010. Citado na p. 16.

GOV.BR. **Aspectos Gerais - Lei de Acesso à Informação**. 2023. Disponível em: https://www.gov.br/acessoainformacao/pt-br/perguntas-frequentes/aspectos-gerais. Citado na p. 9.

Gulbahar Karatas. **What Is a Headless Browser and Its Applications?** 2024. Disponível em: https://research.aimultiple.com/headless-browser/. Citado na p. 9.

GUNASEKARAN, S. S. Stack abuse, guide to parsing html with beautifulsoup. 2023. Disponível em: https://stackabuse.com/guide-to-parsing-html-with-beautifulsoup-in-python/. Citado na p. 13.

IEEE RAS UFCG. **Privacidade na Internet: Como nossos dados são coletados?** 2021. Disponível em: https://edu.ieee.org/br-ufcgras/privacidade-na-internet-como-nossos-dados-sao-coletados/. Citado na p. 10.

JARMUL, R. L. K. **Python Web Scraping**. [*S.l.*]: PACKT, 2015. Citado na p. 14.

JATANA, N.; PURI, S.; AHUJA, M.; KATHURIA, I.; GOSAIN, D. A survey and comparison of relational and non-relational database. **International Journal of Engineering Research & Technology**, v. 1, 2012. Citado na p. 15.

KHDER, M. Web scraping or web crawling: State of art, techniques, approaches and application. **International Journal of Advances in Soft Computing and its Applications**, 2021. Citado nas pp. 14 e 15.

KOUZIS-LOUKAS, D. **Learning scrapy**. [*S.l.*]: Packt Publishing Livery Place, 2016. Citado na p. 14.

Lauren Stone. **The Challenges of Table Data Extraction**. 2024. Disponível em: https://www.nrx.com/challenges-table-data-extraction/. Citado na p. 9.

LGPD. **Lei Geral de Proteção de Dados Pessoais (LGPD) - Lei nº 13.709, de 14 de agosto de 2018**. 2018. https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Citado nas pp. 9 e 14.

Lorena Sobreiro. **A IMPORTÂNCIA DA TRANSPARÊNCIA PÚBLICA PARA O CONTROLE SOCIAL**. 2023. Disponível em: https://repositorio.ifes.edu.br/bitstream/handle/123456789/4261/Plano%20de%20intervenÃğÃčo%20A%20ImportÃćncia%20da%20TransparÃłncia%20PÃžblica%20para%20o%20Controle%20Social%20-%20P%20PUBLICAÃĞÃČO.pdf?sequence=1&isAllowed=y. Citado na p. 10.

MAHESH, B. Machine learning algorithms-a review. **International Journal of Science and Research (IJSR).[Internet]**, v. 9, 2020. Citado nas pp. 18, 19, 20 e 21.

MATPLOTLIB, T. D. T. **Matplotlib: Visualization with Python**. 2025. Disponível em: https://matplotlib.org. Citado na p. 22.

MICROSOFT. **Visual Studio Code**. 2025. Disponível em: https://code.visualstudio.com, Acesso em: 15 jan. 2025. Citado na p. 28.

Nexos. **A importância da Transparência Pública**. 2024. Disponível em: https://nexosgov.com.br/blog/a-importancia-da-transparencia-publica/. Citado na p. 10.

NUMPY, T. D. **NumPy: Fundamental package for array computing in Python**. 2025. Disponível em: https://numpy.org. Citado na p. 22.

ORACLE, C. **MySQL: The world's most popular open source database**. 2025. Versão acessada em janeiro de 2025. Disponível em: https://www.mysql.com. Citado nas pp. 22, 27 e 39.

PANDAS, T. D. T. **Pandas: Python Data Analysis Library**. 2025. Disponível em: https://pandas.pydata.org. Citado nas pp. 22 e 27.

PANT, S.; YADAV, E. N.; MILAN; SHARMA, M.; BEDI, Y.; RATURI, A. Web scraping using beautiful soup. **2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS)**, 2024. Citado na p. 13.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, É. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, 2011. Citado na p. 36.

POLLARD, T.; DERNONCOURT, F.; FINLAYSON, S.; VELASQUEZ, A. Data preparation. **Secondary Analysis of Electronic Health Records**, Springer, 2016. Citado na p. 17.

PYTHON, S. F. **Python Programming Language**. 2024. Disponível em: https://www.python.org. Citado nas pp. 22 e 27.

Python Software Foundation. **glob — Unix style pathname pattern expansion**. [*S.l.*]. Disponível em: https://docs.python.org/3/library/glob.html. Citado na p. 27.

RECLAMAçõES, C. **Relatos de Consumidores - Indicador de Reclamações**. 2025. Disponível em: https://consumidor.gov.br/pages/indicador/relatos/abrir. Citado na p. 24.

RICHARDSON, L. **Beautiful Soup: Library for web scraping in Python**. 2025. Disponível em: https://pypi.org/project/beautifulsoup4/. Citado nas pp. 13, 22, 24, 27 e 39.

SELENIUM; RAGHAVENDRA, S. **Python Testing with Selenium**. [*S.l.*]: Springer, 2021. Citado na p. 14.

SELENIUM, P. **Selenium: Browser Automation**. 2025. Disponível em: https://www.selenium.dev. Citado nas pp. 22, 24, 27 e 39.

SILVA, R. **MySQL Crash Course: A Hands-on Introduction to Database Development**. [*S.l.*]: No Starch Press, 2023. Citado na p. 15.

Sweden Institute. **Openness in Sweden**. 2024. Disponível em: https://sweden.se/life/democracy/openness-in-sweden. Citado na p. 9.