

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

VITOR DOS SANTOS CAVALHEIRO

**Bem-Falado: Um Sistema de Coleta e
Análise de Avaliações de Produtos com
Técnicas de Web Scraping**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia da Computação

Orientador: Prof. Dr.
Marcelo Soares Pimenta

Porto Alegre
2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof^ª. Patricia Helena Lucas Pranke

Pró-Reitora de Graduação: Prof^ª. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^ª. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Engenharia de Computação: Prof. Claudio Machado Diniz

Bibliotecário-chefe do Instituto de Informática: Alexander Borges Ribeiro

*“If I have seen farther than others,
it is because I stood on the shoulders of giants.”*

— SIR ISAAC NEWTON

AGRADECIMENTOS

Em primeiro lugar agradeço aos meus pais, Rogerio e Rosana, por todo o suporte e exemplo ao longo dessa caminhada. Agradeço à minha irmã por sempre estar ao meu lado e me despertar o amor pelas exatas. Aos meus familiares Adriana, Andreia e Laudir por todo incentivo e apoio. A todos os amigos e colegas que tornaram esta passagem mais leve e prazerosa. Aos excelentes professores que se dedicaram diariamente a esta nobre função. E por fim ao estado brasileiro por fornecer acesso a uma educação pública e de qualidade.

RESUMO

Este trabalho apresenta a definição e o desenvolvimento de um sistema de web scraping que visa simplificar a escolha de compra de notebooks e celulares por meio de avaliações feitas por outros consumidores. Inicialmente, são apresentadas suas motivações e objetivos, destacando a importância de consultar as avaliações em decisões informadas. A seguir, descreve-se a arquitetura do sistema, a implementação de scripts de web scraping para coletar dados, a configuração do servidor para comunicação em tempo real e a construção da interface do sistema. Apesar dos desafios enfrentados, o projeto busca oferecer aos consumidores informações transparentes e úteis para embasar suas escolhas, com potencial para expansão e aprimoramentos futuros.

Palavras-chave: Web Scraping. Avaliações de produtos. Python. Scrapy. Node.js. Vue.js. API. Server-Sent Events (SSE). Comércio Eletrônico.

Bem-Falado: A Product Review Collection and Analysis System Using Web Scraping Techniques

ABSTRACT

This paper presents the definition and development of a web scraping system aimed at simplifying the purchase decision of laptops and cell phones through reviews made by real users. Initially, its motivations and objectives are presented, highlighting the importance of consulting reviews in informed decisions. Next, the system architecture is described, the implementation of web scraping scripts to collect data, the server configuration for real-time communication, and the construction of the system's interface. Despite the challenges faced, the project aims to provide consumers with transparent and useful information to support their choices, with potential for expansion and future improvements.

Keywords: Web Scraping, Product Reviews, Python, Scrapy, Node.js, Vue.js, API, Server-Sent Events (SSE), E-Commerce.

LISTA DE ABREVIATURAS E SIGLAS

SPA Single Page Application

API Application Programming Interface

SSE Server-Sent Events

HTTP Hypertext Transfer Protocol

URL Uniform Resource Locator

JSON JavaScript Object Notation

LISTA DE FIGURAS

Figura 2.1 Exemplo de sistema de avaliações utilizado por diversos sites de E-commerce.....	17
Figura 2.2 Avaliações para características específicas.....	17
Figura 2.3 Avaliação classificada como útil.....	18
Figura 3.1 Arquitetura do sistema.....	20
Figura 3.2 Objeto JSON com informações de uma avaliação.....	24
Figura 4.1 Card com uma avaliação e suas informações associadas.	28
Figura 4.2 Card com informações gerais do produto.....	29
Figura 4.3 Tabs de filtragem ativada para comentários que mencionem câmera.....	30
Figura 4.4 Componente exibindo o histórico de buscas do usuário.....	30
Figura 4.5 Interface inicial ao entrar na aplicação.	31
Figura 4.6 Tela de carregamento durante a coleta de avaliações.	32
Figura 4.7 Interface após o recebimento das avaliações.	33
Figura 5.1 Distribuição de Gênero e Escolaridade.....	35
Figura 5.2 Distribuição de Idade.....	36

LISTA DE TABELAS

Tabela 5.1	Resultado da aplicação do formulário de Sistem Usability Scale	37
------------	--	----

SUMÁRIO

1 INTRODUÇÃO	12
1.1 Contextualização	12
1.1.1 Processo de Busca de Avaliações	13
1.1.2 Plataformas Existentes	13
1.2 Objetivos do sistema	14
1.3 Estrutura do Trabalho	15
2 CONCEITOS E TECNOLOGIAS UTILIZADAS	16
2.1 As Avaliações de Produtos na Era do E-Commerce	16
2.2 Tipos de Informações nas Avaliações de Produtos	16
2.2.1 Sistema de Classificação por Estrelas	16
2.2.2 Avaliações Detalhadas	17
2.2.3 Comentários Mais Úteis	17
2.2.4 Comentários e Respostas	18
2.2.5 Utilização das Informações de Avaliações	18
2.3 Web Scraping	19
2.3.1 Conceito e definição	19
2.3.2 Ferramentas utilizadas para a coleta de informações	19
3 DESCRIÇÃO E DEFINIÇÃO GERAL	20
3.1 Arquitetura do sistema	20
3.2 Scripts de Web Scraping	21
3.2.1 Análise e Respeito à Estrutura das Páginas	21
3.2.1.1 Adaptabilidade a Diferentes Tipos de Páginas	21
3.2.1.2 Web Scraping Ético e Responsável	21
3.2.2 Configurações Iniciais	22
3.2.3 Identificação do Produto	22
3.2.4 Coleta de Avaliações	22
3.2.5 Coleta de Informações Gerais	23
3.2.6 Processamento das Informações coletadas	23
3.2.7 Encapsulamento dos Dados	24
3.3 Configuração do Servidor para Server-Sent Events (SSE)	25
3.3.1 Utilização de Node.js e Express	25
3.3.2 Implementação de Server-Sent Events (SSE) na Rota	25
3.4 Execução de Scripts Python	25
3.5 Consumindo Server-Sent Events (SSE) no Front-end	26
4 INTERFACE DO APLICATIVO	27
4.1 Construção e Estrutura da Interface	27
4.1.1 Componente App.vue	27
4.1.2 Componente de Avaliações	28
4.1.3 Componente de Informações	28
4.1.4 Filtragem dos Resultados	29
4.1.5 Componente de Buscas Passadas	30
4.2 Experiência do Usuário	31
5 AVALIAÇÃO DA INTERFACE	34
5.1 Metodologia e System Usability Scale (SUS)	34
5.2 Perfil dos Participantes	35
5.3 Resultados	36
6 PROCESSO DE DESENVOLVIMENTO E CONCLUSÃO	38
6.1 Dificuldades Encontradas no Desenvolvimento	38

6.2 Limitações do Sistema e Próximos Pontos de Desenvolvimento.....	38
6.3 Conclusão.....	39
REFERÊNCIAS.....	40

1 INTRODUÇÃO

Este trabalho foca na concepção de uma plataforma centralizadora de avaliações de notebooks e celulares com o objetivo de simplificar a decisão de compra do consumidor. A ideia baseia-se na percepção da dificuldade em encontrar um grande número de avaliações confiáveis, que retratem a experiência real dos usuários, no mercado brasileiro.

1.1 Contextualização

A aquisição de um bem durável, como celular ou notebook, representa um investimento significativo para a maioria dos consumidores. Espera-se que o produto escolhido satisfaça todas as expectativas e necessidades do usuário. Portanto, para minimizar esse risco, os consumidores buscam, na medida do possível, informações que os ajudem a tomar uma decisão informada. Avaliações de produtos são uma das fontes de informação mais úteis neste contexto, pois podem oferecer *insights* valiosos sobre a experiência real de outros usuários. No entanto, encontrar e analisar essas avaliações pode ser uma tarefa desafiadora e demorada, e muitas vezes as avaliações disponíveis podem não ser confiáveis ou imparciais.

No momento que o consumidor define as configurações necessárias para o seu uso, ele se depara com diversas opções no mercado. Cada uma com suas características, como tamanho de tela, design, sistema operacional e recursos adicionais, teclado retroiluminado ou leitor de digitais.

Para fazer a melhor escolha alguns fatores devem ser levados em conta: preço, avaliações de especialistas, indicações, avaliações de outros consumidores. Obviamente o preço sempre é um dos principais fatores levados em consideração. Porém quando falamos de itens para serem usados a longo prazo, uma pequena variação no preço pode não significar tanto. Outros aspectos também devem ser levados em conta. Avaliações de especialistas por outro lado, buscam vincular as configurações que a máquina possui, com a expectativa de uso do consumidor. Porém não representam a opinião de alguém que fez o uso de fato do equipamento.

Observar avaliações de outros usuários é essencial para tomar uma decisão mais assertiva e evitar possíveis problemas. No entanto, encontrar avaliações confiáveis pode ser um processo difícil e demorado. Além disto, muitas vezes as avaliações disponíveis podem ser manipuladas por empresas ou indivíduos com interesses de denegrir ou melho-

rar a imagem de um produto.

1.1.1 Processo de Busca de Avaliações

Segundo um estudo feito pela Capterra em 2019, 85% dos consumidores consideram importante ou muito importante observar *reviews* e 56% afirmam que sempre leem avaliações online antes de realizar uma compra. A partir do momento que o consumidor deseja pesquisar avaliações sobre o produto, é iniciado um pequeno processo. Nem todos comentários são completos, alguns não são confiáveis, não expressando uma opinião imparcial. Por vezes expõem uma situação atípica em relação ao produto. Por isto, para se chegar a uma conclusão mais assertiva, é necessário coletar um vasto número de avaliações.

Além disso, os consumidores também procuram por vídeos na internet de avaliações do produto. Porém, frequentemente essas avaliações são vinculadas a plataformas de venda e a intenção dos avaliadores é realizar uma propaganda sobre o produto e não representam a avaliação de alguém que de fato utilizou o produto por tempo suficiente para poder dar um *feedback* confiável e imparcial.

No mercado brasileiro não possuímos uma ferramenta que centralize essas informações. Com isto a coleta de avaliações, exige uma busca em variadas fontes como: Blogs, sites das marcas, lojas de eletrônicos, fóruns. Muitas dessas plataformas apresentam informações incompletas e com diferentes critérios de avaliação. Este processo resulta com que muitas vezes os consumidores não concluem o processo da melhor forma possível ou nem comecem ele.

1.1.2 Plataformas Existentes

Existem soluções no mercado que coletam e centralizam informações sobre os produtos, mas relacionado a preços e configurações apenas. "Zoom" e "Buscapé" são dois exemplos. Elas realizam consultas em diversas lojas, visando comparar os preços e entregar aos seus usuários as melhores ofertas. No "Zoom" encontramos uma seção de comentários, mas com pouquíssima informação. Apenas avaliações adicionadas diretamente na plataforma, não havendo qualquer incentivo a adicionarem avaliações.

Também temos plataformas voltadas aos consumidores fornecerem *feedbacks* como

o "Reclame Aqui" e "Consumidor.gov". Porém o uso das plataformas é muito voltado para os clientes reportarem experiências ruins em relação às lojas. Expondo as lojas mais problemáticas no processo de compra. Sejam entregas com atraso, produtos com defeito, mau suporte ao cliente.

As plataformas onde mais encontramos *reviews* são os *e-commerces* que possuem o produto em questão. Como os sites dos fabricantes ou lojas de eletrônicos: Magazine Luiza, Casas Bahia, Amazon, Mercado Livre, Americanas e entre outras. Com raras exceções, essas plataformas possuem seções de avaliações, cada uma com seus diferentes critérios adotados. Alguns mais sucintos, contendo apenas uma avaliação simples e espaço para texto, outras com múltiplos critérios de avaliação, considerando características específicas quando se tratando de notebooks, como bateria, desempenho, acabamento.

1.2 Objetivos do sistema

Considerando o cenário apresentado, o problema de pesquisa que este trabalho se propõe a resolver pode ser resumido da seguinte maneira: Como criar uma plataforma confiável que centralize avaliações de notebooks e celulares de diversas fontes, tornando a pesquisa de produtos mais eficiente e confiável para os consumidores?

O sistema desenvolvido tem como objetivo permitir que os usuários pesquisem avaliações de produtos de várias fontes na web. Isso é realizado através de uma aplicação web de página única (SPA) criada com Vue.js.

Os usuários interagem com a interface do sistema inserindo um produto que desejam pesquisar. O sistema então retorna uma lista de avaliações para esse produto, que pode ser filtrada e ordenada de acordo com as preferências do usuário. Além disso, o sistema fornece um resumo das avaliações para fornecer uma visão geral rápida da percepção do produto pelo mercado.

O sistema é alimentado por dados coletados por *spiders* (*scripts* de coleta de dados) que são escritos em Python e utilizam a biblioteca Scrapy para a realização de *web scraping*. Esses *scripts* são responsáveis por buscar avaliações de produtos de diversas fontes na internet. Essas avaliações são então enviadas para a interface do usuário por meio de uma implementação de SSE (*Server-Sent Events*) em Node.js com Express.

1.3 Estrutura do Trabalho

O presente trabalho está estruturado em seis capítulos. O Capítulo 2 se aprofunda nos conceitos e tecnologias utilizadas, descrevendo a importância das avaliações de produtos no atual cenário de e-commerce e introduzindo a técnica de web scraping. No Capítulo 3, é detalhada a descrição e definição geral do sistema, passando por tópicos como a arquitetura do sistema, scripts de web scraping, configuração do servidor e outros aspectos técnicos. O Capítulo 4 é dedicado à interface do aplicativo, apresentando sua construção, estrutura, componentes e focando na experiência do usuário. Seguido por uma avaliação da usabilidade da plataforma no capítulo 5. No Capítulo 6, é realizada uma análise do sistema e do processo de desenvolvimento, abordando as dificuldades, limitações e pontos para desenvolvimentos futuros. Por fim, o Capítulo 7 traz a conclusão do trabalho, consolidando os pontos abordados e refletindo sobre o impacto e relevância do projeto, associado a uma seção que explora as perspectivas e intenções de trabalhos futuros.

2 CONCEITOS E TECNOLOGIAS UTILIZADAS

Nesta seção, serão explorados os diferentes tipos de informações presentes nas avaliações de produtos, desde sistemas de classificação por estrelas até avaliações detalhadas e comentários úteis. Bem como explicar as técnicas utilizadas para a coleta desses dados.

2.1 As Avaliações de Produtos na Era do E-Commerce

Com o advento da internet e a consequente explosão do comércio eletrônico, a maneira como os consumidores interagem e se engajam com produtos mudou drasticamente. Antes da era digital, os compradores dependiam, em grande parte, de publicidade, experiências de amigos e familiares ou conselhos de vendedores em lojas físicas. Hoje, no ambiente digital, as avaliações de produtos online representam uma poderosa ferramenta para ajudar os consumidores a tomar decisões informadas.

As avaliações de produtos online tornaram-se um aspecto essencial dos sites de comércio eletrônico, fornecendo uma plataforma para que os clientes compartilhem suas experiências e opiniões sobre produtos específicos. Essas avaliações desempenham um papel crucial na decisão de compra de novos clientes, fornecendo *insights* valiosos sobre a qualidade, funcionalidade e eficácia de um produto.

2.2 Tipos de Informações nas Avaliações de Produtos

Nas plataformas de *e-commerce*, diferentes tipos de informações são utilizados nas avaliações de produtos para proporcionar aos clientes uma visão abrangente dos produtos. Essas variações, desde avaliações quantitativas até qualitativas, ajudam os consumidores a formar uma imagem completa do produto antes da compra.

2.2.1 Sistema de Classificação por Estrelas

A maioria das plataformas *online* implementa um sistema de classificação por estrelas, onde os usuários podem avaliar um produto em uma escala de 1 a 5 estrelas. Essa avaliação quantitativa fornece um indicador rápido do nível de satisfação do cliente

com o produto.

Figura 2.1: Exemplo de sistema de avaliações utilizado por diversos sites de E-commerce.

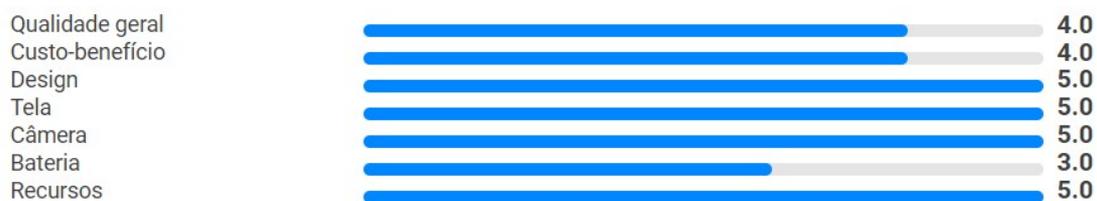


Fonte: <https://www.magazineluiza.com.br/>

2.2.2 Avaliações Detalhadas

Alguns sites vão além, permitindo que os clientes avaliem características específicas do produto, como qualidade, eficiência, estética, câmera (para produtos eletrônicos), bateria, etc. Isso permite uma análise mais detalhada e segmentada, ajudando outros consumidores a entender os pontos fortes e fracos do produto em diferentes aspectos.

Figura 2.2: Avaliações para características específicas.



Fonte: <https://www.magazineluiza.com.br/>

2.2.3 Comentários Mais Úteis

Muitos *e-commerces* destacam avaliações que foram marcadas como úteis por outros usuários. Essas avaliações "mais úteis" são muitas vezes ricas em detalhes e oferecem uma visão mais equilibrada do produto.

Figura 2.3: Avaliação classificada como útil.



Fonte: <https://www.amazon.com.br>

2.2.4 Comentários e Respostas

Além de fornecer avaliações, muitos sites também permitem que os usuários comentem as avaliações, criando um fórum de discussão. Isso pode incluir perguntas e respostas adicionais, esclarecimentos e até mesmo interações diretas com os fabricantes ou vendedores.

2.2.5 Utilização das Informações de Avaliações

Ao oferecer uma variedade de informações, desde classificações por estrelas até análises detalhadas de características específicas, essas avaliações desempenham um papel vital na construção da confiança e na promoção de uma experiência de compra transparente e informada.

Este projeto visa utilizar essas informações públicas para fornecer ao usuário uma vasta quantidade de avaliações de produtos online. Além de captar uma diversidade de dados disponíveis, desde a classificação por estrelas até comentários detalhados e análise de características específicas, através de técnicas de *Web Scraping*

2.3 Web Scraping

A prática de *web scraping* tem crescido em relevância à medida que a quantidade de informações disponíveis na web aumenta exponencialmente. Esta técnica possibilita que empresas e indivíduos acessem, analisem e utilizem dados que de outra forma estariam apenas visualmente disponíveis para leitores humanos. Esta seção discute os fundamentos da técnica e se aprofunda nas ferramentas específicas utilizadas neste trabalho.

2.3.1 Conceito e definição

Web scraping refere-se à técnica utilizada para extrair informações de sites na internet. Este processo pode ser realizado manualmente por um usuário navegando através de um site e copiando dados. No entanto, na maioria dos casos, *web scraping* é realizado automaticamente por um robô. Essa técnica é fundamental na arquitetura proposta, permitindo a coleta automatizada de avaliações em diversos sites.

2.3.2 Ferramentas utilizadas para a coleta de informações

Nesta subseção, exploraremos as ferramentas específicas adotadas neste trabalho para realizar o *web scraping*. Python, por ser uma linguagem de programação flexível e poderosa, é a escolha feita pela maioria dos desenvolvedores que atuam com *web scraping*, graças à sua vasta gama de bibliotecas, forte comunidade e facilidade de uso.

Por sua vez, Scrapy, uma biblioteca Python, foi projetada especificamente para *web scraping*. Ela não apenas simplifica o processo de extração de informações, mas também fornece recursos robustos para processar e armazenar os dados obtidos. Para as necessidades deste projeto, o Scrapy se mostrou essencial, coletando os dados com ótima eficiência e precisão, além de fornecer uma gama de ferramentas de depuração que facilitam o desenvolvimento.

Outras bibliotecas, como Selenium e BeautifulSoup, também foram consideradas na decisão. No entanto, foi optado pela Scrapy devido ao seu menor tempo de coleta de informações, proporcionando assim uma resposta mais ágil ao usuário.

3 DESCRIÇÃO E DEFINIÇÃO GERAL

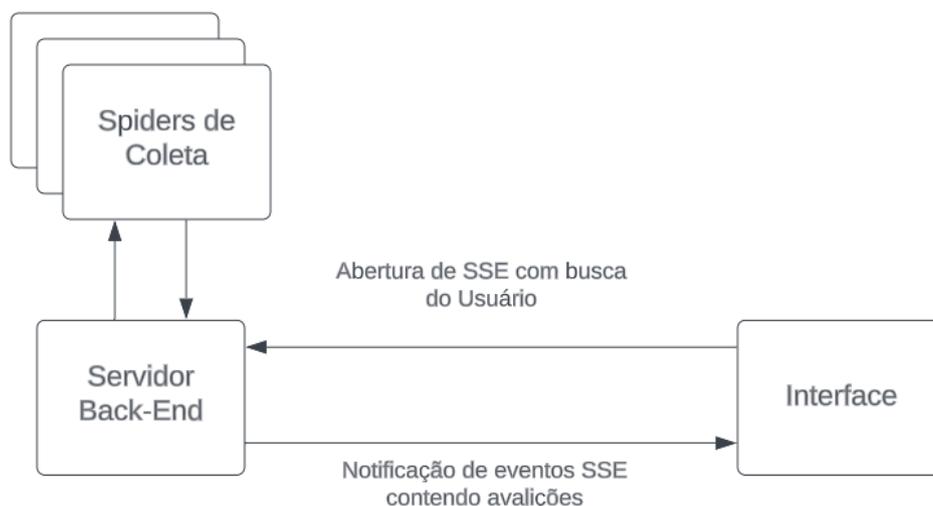
3.1 Arquitetura do sistema

A arquitetura do sistema é dividida em três componentes principais: a interface do usuário (*front-end*), o servidor (*back-end*) e os *scripts* de coleta de dados.

O *front-end* é construído como uma SPA (*Single Page Application*) usando Vue.js. Esta abordagem foi escolhida por sua capacidade de fornecer uma experiência de usuário simples e com transições rápidas entre os estados da aplicação sem a necessidade de recarregar a página. O Vue.js também fornece uma estrutura clara para a organização do código, o que facilita a manutenção e o desenvolvimento contínuo do sistema.

O *back-end* do sistema é construído usando Node.js com Express devido a sua simplicidade para expor as rotas para a interface. A implementação realiza a execução em paralelo dos *scripts* e é responsável por transmitir suas coletas para a interface através de eventos SSE.

Figura 3.1: Arquitetura do sistema.



Fonte: Captura de Tela da Aplicação

Por fim, os *scripts* de coleta de dados são escritos em Python e utilizam a biblioteca Scrapy para realizar *web scraping* nas páginas de avaliação de produtos. A Scrapy foi escolhida por sua robustez e velocidade, permitindo a coleta de dados de uma ampla variedade de fontes na web com rápido tempo de execução. Esses *scripts* também reali-

zam *requests* adicionais para obter mais comentários, garantindo uma grande quantidade de avaliações para cada produto.

3.2 Scripts de Web Scraping

Esta seção descreverá o processo de elaboração dos *scripts* de coleta utilizados no projeto.

3.2.1 Análise e Respeito à Estrutura das Páginas

A coleta de dados via *web scraping* requer uma compreensão aprofundada da estrutura HTML das páginas web alvo. Esta etapa é crítica para localizar de forma precisa os elementos que contêm as informações desejadas. No entanto, é importante notar que a eficácia dessa metodologia pode variar dependendo do tipo de página web.

3.2.1.1 Adaptabilidade a Diferentes Tipos de Páginas

Enquanto essa abordagem funciona bem para páginas estáticas que têm uma estrutura HTML bem definida, ela pode não ser adequada para páginas que carregam conteúdo dinamicamente usando JavaScript. Para esses casos, bibliotecas como Selenium podem ser utilizadas para simular interações do usuário e capturar informações carregadas dinamicamente. Além disso, páginas com atualizações em tempo real via WebSockets ou AJAX podem exigir estratégias de scraping especializadas.

3.2.1.2 Web Scraping Ético e Responsável

O projeto também buscou realizar web scraping de forma ética e responsável. Isso envolveu uma verificação rigorosa dos arquivos *robots.txt* dos sites alvo, que especificam quais partes do site podem ou não ser acessadas por *crawlers*. Em respeito às diretrizes fornecidas pelos administradores dos sites e para evitar qualquer acesso não autorizado, todas as rotas explicitamente marcadas como proibidas nos arquivos *robots.txt* foram evitadas.

3.2.2 Configurações Iniciais

Os *scripts* de raspagem começam com a definição da classe *Spider* e suas configurações iniciais. Ali, além das propriedades do *framework*, são definidas as *URLs* iniciais a partir da pesquisa do usuário. Essas *URLs* são construídas observando a requisição HTTP disparada pelos campos de pesquisa dos sites. Além disso são definidos os cabeçalhos do agente do usuário para uma solicitação de navegador. Essa configuração inicial permite a preparação para a raspagem, definindo os parâmetros e as *URLs* iniciais para começar a coleta.

3.2.3 Identificação do Produto

A primeira fase do processo de raspagem envolve a localização do produto, após a chegada da resposta HTTP da requisição inicial de busca. Nesta etapa, os sites apresentam os resultados mais próximos da busca desejada. Por esse motivo, na interface os usuários são encorajados a adicionarem as informações de componentes de hardware dos produtos, como memória RAM e processador. Com esses resultados, o *script* acessa a página do produto melhor elencado. Em alguns casos, durante esse processo a *spider* faz a coleta do Identificador do produto para ser utilizado futuramente em requisições de API.

3.2.4 Coleta de Avaliações

A coleta de avaliações é uma etapa vital na raspagem de dados, e é realizada de maneira diferenciada, a depender das características do site-alvo. No contexto deste trabalho, os sites foram categorizados em dois grupos: aqueles que fornecem rotas para acesso direto às avaliações e aqueles que necessitam de coleta estática por meio de requisições de novas páginas HTML.

No caso dos sites que disponibilizam uma API, os comentários são coletados por meio de chamadas específicas que permitem a extração de um grande número de avaliações. Foi utilizado o limite de 50 avaliações por requisição e até 5 requisições realizadas serialmente, garantindo um equilíbrio entre a quantidade de dados coletados e o respeito aos servidores dos sites.

Já nos sites que não possuem uma API, a coleta de avaliações é realizada direta-

mente das páginas dos produtos, sendo raspados sequencialmente. Esse processo é mais trabalhoso e exige a navegação e coleta de dados página por página. Mesmo que essas plataformas possuam uma grande quantidade de avaliações em suas bases de dados, dificilmente serão coletadas mais de 50. Apesar de não contribuírem tão significativamente para produtos muito populares, em produtos com menos avaliações disponíveis eles se fazem necessários.

Essa abordagem dual permitiu uma coleta eficiente e adaptada às diferentes estruturas e políticas dos sites, resultando em um conjunto diversificado e rico de avaliações para análise subsequente. Independe do processo de coleta, cada *crawler* armazena todas essas avaliações em uma lista padrão para todos os buscadores.

3.2.5 Coleta de Informações Gerais

Além dos comentários, informações gerais sobre avaliações e características específicas do produto são coletadas.

Todos os sites costumam apresentar algum tipo de resumo sobre o produto, o trabalho buscou compilar essas variedades de informações para apresentar elas conjuntamente.

Essas informações coletadas são vitais para observar características mais específicas do produto, como bateria, processamento e outros aspectos relevantes.

O processo de coleta dessas informações segue o padrão das avaliações. Algumas foram coletadas estaticamente, outras coletadas via requisições.

3.2.6 Processamento das Informações coletadas

A partir das avaliações coletadas são construídas entidades utilizadas por todas as Spiders, contendo propriedades básicas de uma avaliação: identificador, autor, data, fonte, comentário e nota.

Essas avaliações são ordenadas por tamanho do comentário. Partindo do pressuposto que avaliações maiores, apresentam mais informações sobre o produto, logo pode mais informações relevantes. Essa lista é percorrida posteriormente observando a presença de palavras específicas, a fim de filtrar as avaliações por assuntos como bateria, câmera ou processamento.

As informações gerais enviadas em uma lista, porém com uma estrutura específica para cada tipo de informação coletada. Ficando a cargo da interface lidar com uma implementação visual para cada tipo de informação.

3.2.7 Encapsulamento dos Dados

Após todo esse processo de coleta e análise de dados, eles são encapsulados em um objeto do formato JSON para serem utilizados pelo servidor. Esse processo garantiu uma fácil utilização e manipulação dos dados por parte do *back-end*. Ficando apenas responsável por gerenciar a execução dos *scripts* sem se preocupar com a execução interna dos mesmos.

Na figura abaixo podemos ver um objeto contendo as informações de uma avaliação. Ele é retornado pelos *scripts* de coleta nesse formato e repassado pelo servidor à interface.

Figura 3.2: Objeto JSON com informações de uma avaliação.

```
"reviews": [  
  {  
    "id": "PF:15409767",  
    "content": "Aparelho bom, porém atendimento da ponto do shopping Boulevard de Campos dos goytacazes horrível, uma funcionária grávida que parece a dona da ponto entra e sai sem uniforme, parece até que o cliente não existe; tomei um chá de cadeira esperando a retirada \"rapida\" que de rápido não tem nada. Uma vergonha para a Ponto; Péssimo atendimento, da próxima pago mais caro mas compro na concorrente, e não indico a ponto mais.",  
    "source": "Ponto Frio",  
    "sourceUrl": "https://www.pontofrio.com.br/Android/b?gclid=Cj0KCQjw6KunBhDxARIsAKFUGs-YV6aTum8KFsAPE7Bii0r1HY0xXSExiusa2664Qcunk41JATiK-8aA1BkEALw_wCB&gclidsrc=aw.ds",  
    "rating": 4,  
    "name": "Ronaldo",  
    "date": "10/07/23",  
    "bBattery": false,  
    "bCamera": false,  
    "bQuality": false,  
    "bEfficiency": true,  
    "bAesthetic": false  
  },  
  {  
    "id": "15648645"
```

Fonte: Captura de Tela do Ambiente de Desenvolvimento

3.3 Configuração do Servidor para Server-Sent Events (SSE)

3.3.1 Utilização de Node.js e Express

O ambiente de execução Node.js e o framework web Express foram a base para a criação e disponibilização da rota que fornece para a interface os dados desejados. O Node.js fornece um ambiente eficiente e escalável, enquanto o Express facilita a criação e manutenção de rotas.

3.3.2 Implementação de Server-Sent Events (SSE) na Rota

A implementação de SSE foi feita na rota `/reviewsSearchSSE` e envolveu algumas etapas. Inicialmente, foram configurados os cabeçalhos da resposta para habilitar o envio de eventos do servidor. Esses cabeçalhos asseguram a criação de uma conexão persistente, essencial para o fluxo constante de informações.

Em seguida, uma função é chamada para cada *script* de *scraping* ser executado em paralelo, iniciando um processo Python para executar o *script* correspondente. Os dados da saída padrão são capturados e enviados para o cliente como eventos SSE.

O servidor é configurado para tratar eventos em tempo real, capturando e enviando dados de forma incremental para o cliente, mantendo assim um fluxo constante de informações até o final da execução de todos os *scripts*.

3.4 Execução de Scripts Python

O servidor utiliza a função `runScriptSSE` para chamar os *scripts* de *scraping* para cada termo de pesquisa. Esta função recebe o nome do *script* e o termo de pesquisa como argumentos, além do objeto de resposta SSE.

Cada chamada para `runScriptSSE` inicia um novo processo Python que executa o *script* de *scraping* com o termo de pesquisa como argumento. Os dados de saída padrão (`stdout`) e de erro padrão (`stderr`) do processo Python são capturados pelo servidor e adicionados às variáveis de saída e erro, respectivamente. No processamento da saída de dados, é validado que de fato foram recebidas avaliações. Caso os dados tenham sido coletados, uma *flag* é ativada para sinalizar o recebimento. Algumas vezes os sites não

apresentam erros, mas não conseguem de fato realizar a coleta.

Quando o processo Python é encerrado (ou seja, quando a *spider* terminou de coletar os dados), o servidor verifica o código de saída do processo e a *flag* de recebimento de dados. Se o código de saída é 0 (o que significa que o processo foi concluído com sucesso) e a *flag* está ativada, o servidor analisa a saída como um objeto JSON e envia o resultado para o cliente como um evento SSE.

Se algum processo retorna um erro de execução ou a *flag* está desativada. Ele é colocado em uma fila para mais uma tentativa de execução após 1 segundo. Este processo é limitado em até 3 tentativas por *script*.

3.5 Consumindo Server-Sent Events (SSE) no Front-end

No lado do cliente, o fluxo de eventos SSE do servidor é consumido usando a API EventSource embutida na maioria dos navegadores modernos. Esta API permite criar uma conexão persistente com o servidor e receber atualizações em tempo real.

Dois manipuladores de eventos são definidos para a instância EventSource: `onmessage` e `onerror`. O manipulador `onmessage` é chamado sempre que o servidor envia um novo evento. Cada evento contém os dados de avaliação recém-coletados como uma string JSON, que é então analisada em um objeto JavaScript e adicionada ao estado do aplicativo.

Se os dados recebidos contêm um erro, a interface do usuário é notificada e o estado de carregamento é atualizado para `false`. Se não houver erros, os dados são adicionados às listas de avaliações e informações, e os sinais de recebimento são ativados.

O manipulador de erro `onerror` é chamado se ocorrer um erro com a conexão SSE. Neste caso, a função registra o erro, atualiza o estado de carregamento para `false` e fecha a conexão SSE.

Desta forma, a função `sendRequestSSE` garante que a interface do usuário seja atualizada em tempo real conforme os dados de avaliações são coletados pelo servidor, oferecendo uma experiência fluida e interativa ao usuário.

4 INTERFACE DO APLICATIVO

Neste capítulo serão explicadas as ferramentas utilizadas, a estrutura do *front-end* e o funcionamento dos componentes que o integram.

4.1 Construção e Estrutura da Interface

A construção da interface do usuário para este aplicativo é fundamentada no uso do *framework* Vue.js, uma biblioteca JavaScript voltada para facilitar a criação de interfaces interativas, lidando com reatividade de dados e mudanças de estado. O propósito é assegurar uma experiência de uso simples e eficaz, exibindo as informações de maneira clara e acessível. Na interface, os usuários podem compreender rapidamente o propósito do aplicativo e conduzir suas pesquisas para obter as informações relevantes sobre os produtos que desejam adquirir.

A escolha do Vue.js oferece uma estruturação eficaz do código em componentes, tornando a interface mais modular e de fácil manutenção. Na próxima seção serão abordados os componentes que compõem a interface.

4.1.1 Componente App.vue

O componente central, App.vue, assume o papel de raiz do aplicativo. Ele abarca o layout principal, bem como a lógica necessária para solicitar dados do servidor e administrar os estados da aplicação.

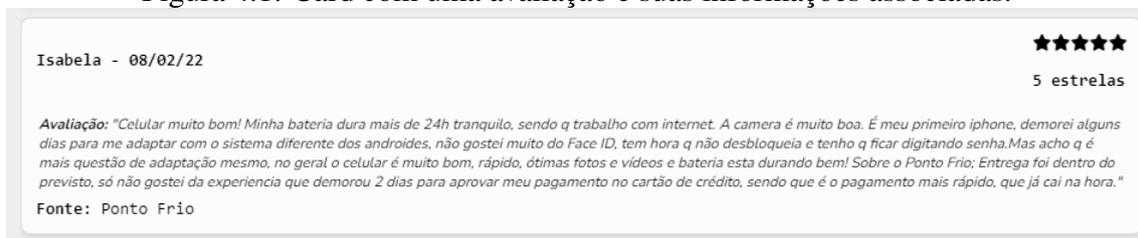
Possuímos um estado inicial, onde o usuário recebe informações sobre a aplicação e como utilizá-la. Após a pesquisa, a interface apresenta uma mensagem informando que a pesquisa esta sendo feita pelo lado do servidor e este processo por levar alguns segundos. A medida que os dados começam a chegar no *front-end*, um contêiner é mostrado contendo informações de resumo (representadas pelo componente `InfoCard`), as avaliações obtidas (implementadas pelo componente `ReviewCard`) e opções de filtragem (disponibilizadas pelo componente `WordFilters`). Além disso, opções de ordenação das avaliações, com base no comprimento ou na nota, também são oferecidas.

4.1.2 Componente de Avaliações

O componente ReviewCard é responsável por exibir os dados de uma única avaliação. São eles: Autor, data, nota, comentário e fonte, sendo ela um link direto para o site que a avaliação foi coletada. Ele recebe esses dados como um objeto e os exibe em um formato de card. A medida que novas avaliações vão chegando elas vão sendo inseridas ao final da fila que exibe todas as *cards* de avaliações.

Os objetos já vêm ordenados do servidor pelo tamanho do comentário, porém o usuário tem a possibilidade de reordená-los por data e nota também.

Figura 4.1: Card com uma avaliação e suas informações associadas.



Fonte: Captura de Tela da Aplicação

4.1.3 Componente de Informações

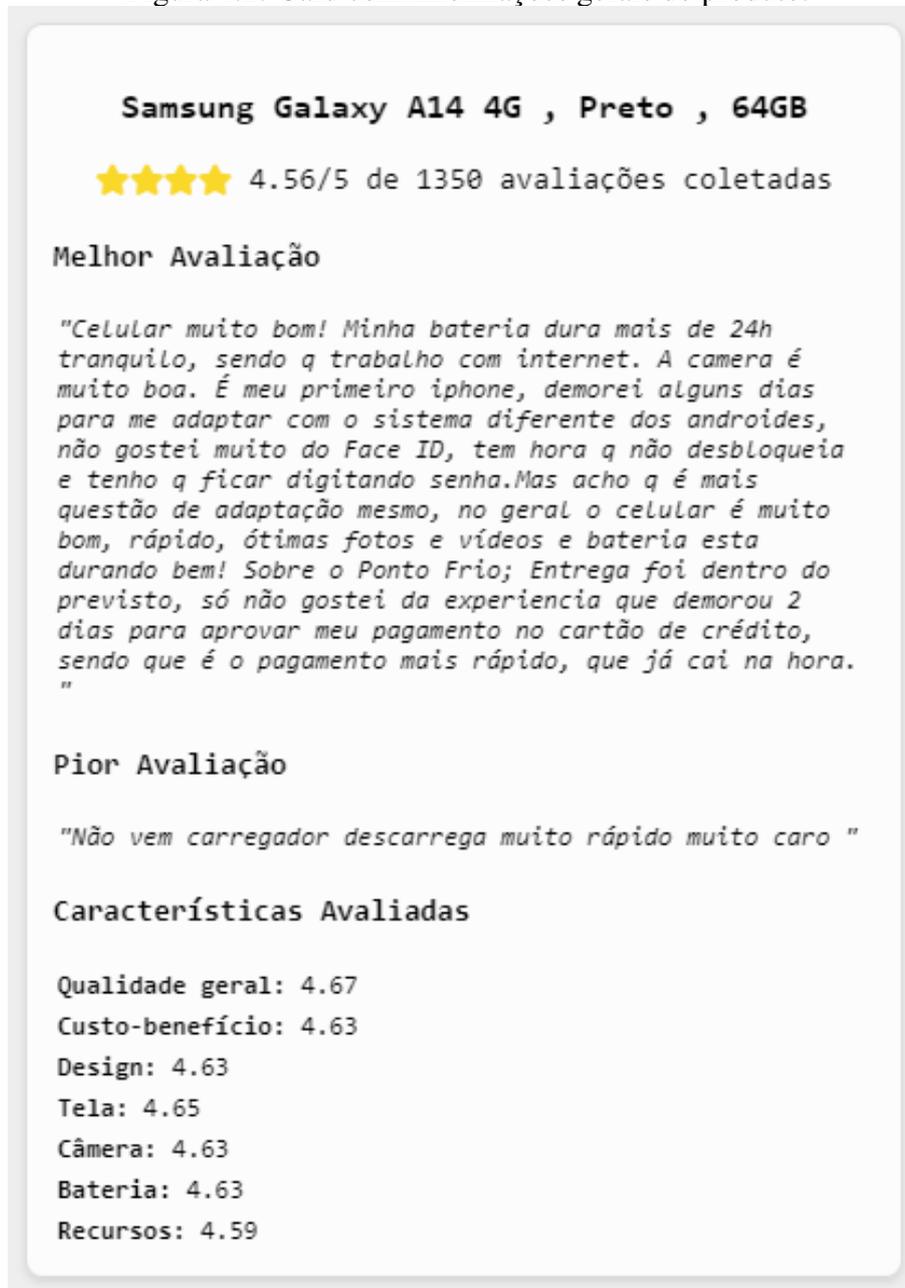
O componente InfoCard representa um agregado de diferentes informações coletadas nas plataformas. Algumas sinalizam melhores ou piores comentários e outras apresentam notas para características específicas dos produtos. A interface foi construída para processar esses dados de forma incremental. Caso algum site não envie as suas informações específicas, a interface apenas não renderizará o componente,

O ponto em comum é a representação da nota média e a quantidade de avaliações. As plataformas costumam ter um número muito grande de avaliações sem comentário (apenas nota) em relação as avaliações com texto. O sistema se valeu de coletar esses números a fim de construir uma média mais próxima da realidade. Com isso a média apresentada no componente de informações, não possui uma relação direta com a média das avaliações exibidas na aplicação.

As melhores ou piores avaliações, são definidas por sites que apresentam essa forma de exibição. Essas aplicações geralmente consideram votos de utilidade fornecidos por usuários que acessam as plataformas.

As avaliações de características específicas, são fornecidas em apenas uma aplicação coletada por enquanto. Por isso a média geral, apresentada na parte superior do componente, não possui com a média das características.

Figura 4.2: Card com informações gerais do produto.



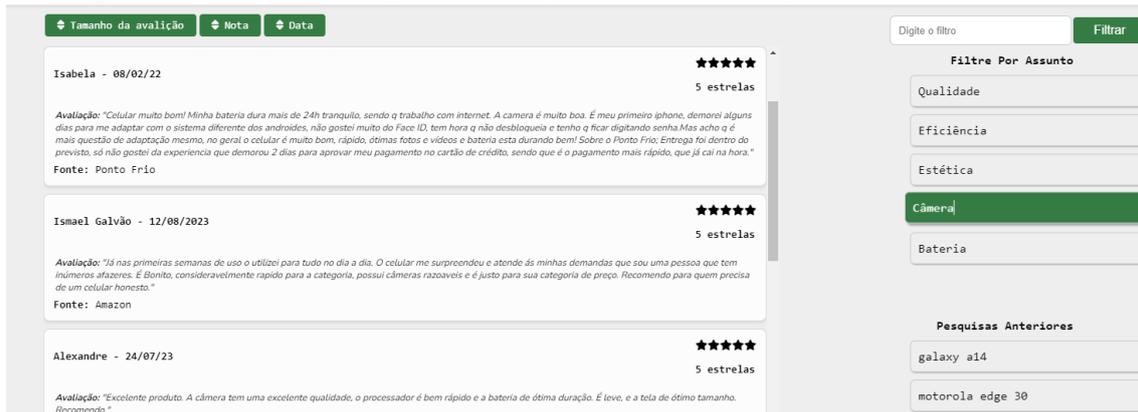
Fonte: Captura de Tela da Aplicação

4.1.4 Filtragem dos Resultados

O componente WordFilters permite ao usuário filtrar as avaliações por palavras-chave específicas. E possui uma lista de filtros, quando um filtro é selecionado, ele filtra

a lista de avaliações para mostrar apenas comentários que mencionem os assuntos relacionados. São eles: Qualidade, Eficiência, Estética, Câmera ou Bateria. Cada objeto representante de uma avaliação, vem do *back-end* com uma *flag*, habilitada ou não, para cada filtro. Esse processo de ativação de *flags* é realizado pelos *scripts* de raspagem após a coleta dos dados, como já explicado anteriormente.

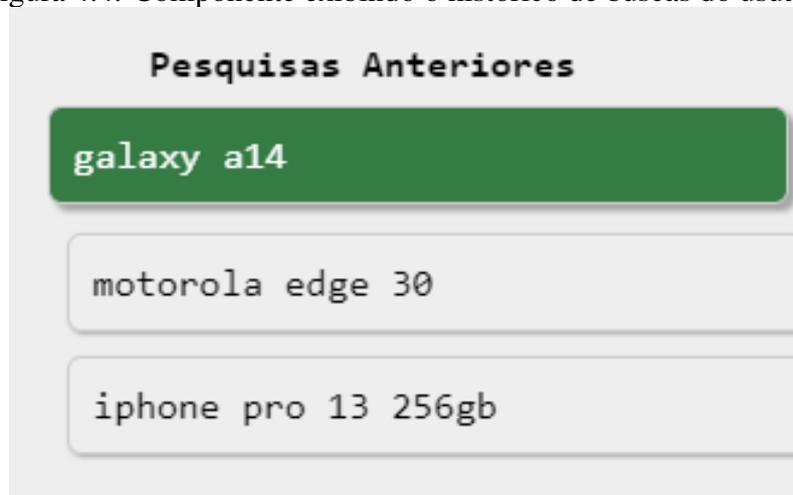
Figura 4.3: Tabs de filtragem ativada para comentários que mencionem câmera.



4.1.5 Componente de Buscas Passadas

O componente de PastSearches oferece ao usuário uma visualização das últimas consultas realizadas, permitindo um acesso rápido a resultados anteriores. Cada termo de busca é apresentado como um item clicável que, ao ser selecionado, recarrega as informações relacionadas a essa consulta anterior.

Figura 4.4: Componente exibindo o histórico de buscas do usuário.



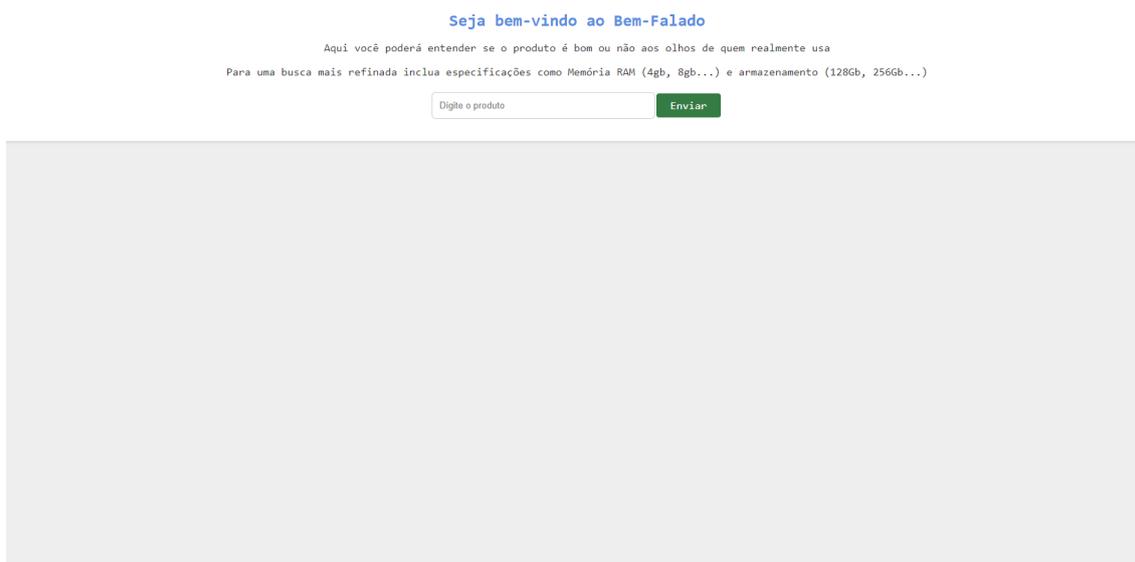
Fonte: Captura de Tela da Aplicação

No nível de implementação, a lista armazena até 10 produtos pesquisados anteriormente. Ao selecionar um produto da lista, o sistema recupera os *reviews* e informações relacionados, sendo armazenados no máximo 20 avaliações localmente, enquanto o *back-end* pode realizar novas requisições para adquirir mais informações se necessário.

4.2 Experiência do Usuário

Ao acessar o site, o usuário é recebido com uma breve explicação sobre a proposta da plataforma e como utilizar a funcionalidade de pesquisa. Essa introdução visa orientar o usuário sobre como aproveitar ao máximo a aplicação para obter informações confiáveis sobre notebooks e celulares.

Figura 4.5: Interface inicial ao entrar na aplicação.



Fonte: Captura de Tela da Aplicação

Uma vez que o usuário decide realizar uma pesquisa, ele insere o termo de busca desejado no campo designado e aciona o botão de pesquisa. Imediatamente, a interface entra no modo de carregamento, exibindo uma mensagem informativa que destaca o processo de coleta de dados em andamento. O usuário é informado de que a coleta de informações pode levar um tempo, garantindo que ele esteja ciente do processo enquanto aguarda os resultados.

Figura 4.6: Tela de carregamento durante a coleta de avaliações.

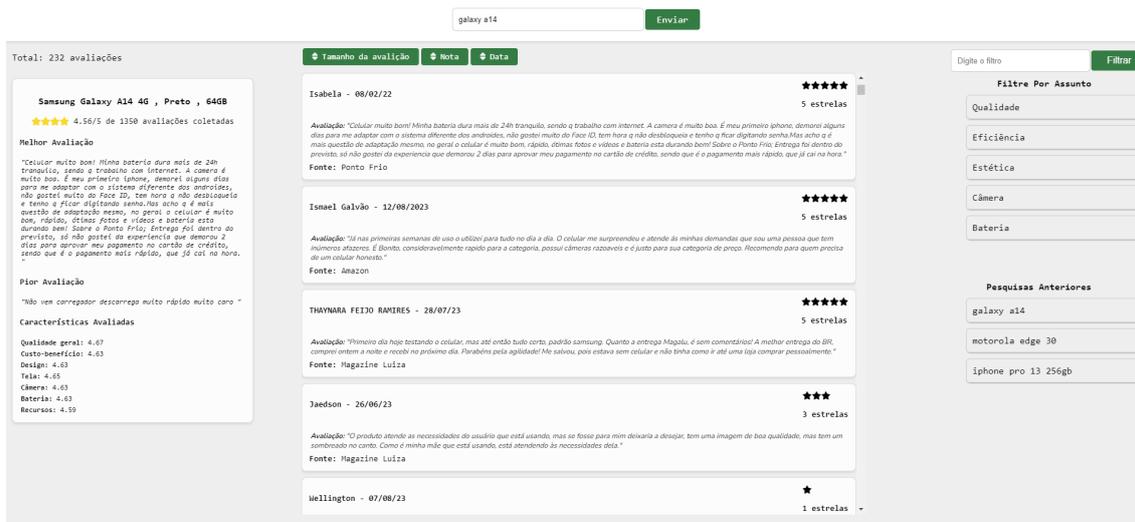


Fonte: Captura de Tela da Aplicação

Assim que os resultados da coleta de dados são obtidos, o usuário é apresentado a um layout estruturado. À esquerda, um painel de visão geral exibe informações importantes sobre o produto, como sua nota média e informações gerais. No centro da interface, as avaliações propriamente ditas são apresentadas, ordenadas pelo tamanho dos comentários. O usuário tem a liberdade de selecionar a ordem das avaliações, escolhendo entre a ordenação por tamanho, data ou nota, a fim de atender às suas preferências.

Na parte direita da interface, os filtros estão disponíveis para permitir que o usuário refine as avaliações de acordo com suas necessidades. Com esses filtros, é possível focar nas avaliações que mencionam aspectos específicos do produto, como bateria, câmera ou outras características relevantes. Isso proporciona uma maneira eficiente de encontrar as informações desejadas em meio a uma variedade de avaliações.

Figura 4.7: Interface após o recebimento das avaliações.



Fonte: Captura de Tela da Aplicação

Após a sua análise, caso o usuário deseje realizar novas pesquisas, a aplicação vai mantendo as pesquisas recentes no canto inferior direito. A interface guarda até 10 pesquisas e armazena as informações gerais e 20 avaliações por produto. Desta maneira, caso o usuário deseje voltar para o produto anterior ele já possui os dados em tela e não precisa aguardar novamente o tempo de busca sem ter nenhuma informação. Mas, da mesma forma o servidor dispara as requisições e notifica o *front-end* quando obtém mais avaliações.

Em suma, a experiência do usuário foi projetada para fornecer uma jornada intuitiva e informada. Desde a introdução à interface até a visualização de avaliações detalhadas e filtradas.

5 AVALIAÇÃO DA INTERFACE

Uma avaliação preliminar do sistema foi realizada convidando 11 indivíduos a utilizar a aplicação e posteriormente preencher um questionário sobre a mesma, além de serem convidados a fornecerem críticas e sugestões. O caso de uso sugerido foi realizar a comparação entre dois celulares (Samsung Galaxy A14 e Motorola Moto G23) e dois notebooks (Dell inspiron 15 e Acer aspire 5), porém os deixando livres para utilizar como bem entenderem.

5.1 Metodologia e System Usability Scale (SUS)

A avaliação da usabilidade é um aspecto vital para determinar a eficácia de um sistema. No contexto deste trabalho, utilizamos o *System Usability Scale (SUS)* para quantificar a usabilidade de nossa aplicação. Desenvolvido por John Brooke em 1995, o SUS é uma escala renomada que apresenta uma visão global das avaliações subjetivas de usabilidade. Ele é composto por 10 perguntas com 5 alternativas entre Discordo Totalmente ou Concordo Totalmente.

1. Eu acredito que eu gostaria de usar este sistema frequentemente.
2. Eu achei o sistema desnecessariamente complexo.
3. Eu achei o sistema fácil de usar.
4. Eu acredito que eu precisaria do suporte de uma pessoa técnica para conseguir usar este sistema.
5. Eu achei que as várias funções neste sistema estavam bem integradas.
6. Eu achei que havia muita inconsistência neste sistema.
7. Eu imagino que a maioria das pessoas aprenderia a usar este sistema muito rapidamente.
8. Eu achei o sistema muito pesado para usar.
9. Eu me senti muita confiança usando o sistema.
10. Eu precisei aprender muitas coisas antes de poder começar a usar este sistema.

O SUS é estruturado como uma escala Likert e seu cálculo resulta em uma pontuação de 0 a 100. Para calcular a pontuação SUS, primeiro subtraia 1 das respostas das perguntas ímpares e faça o processo inverso para perguntas pares, onde a opção de "Dis-

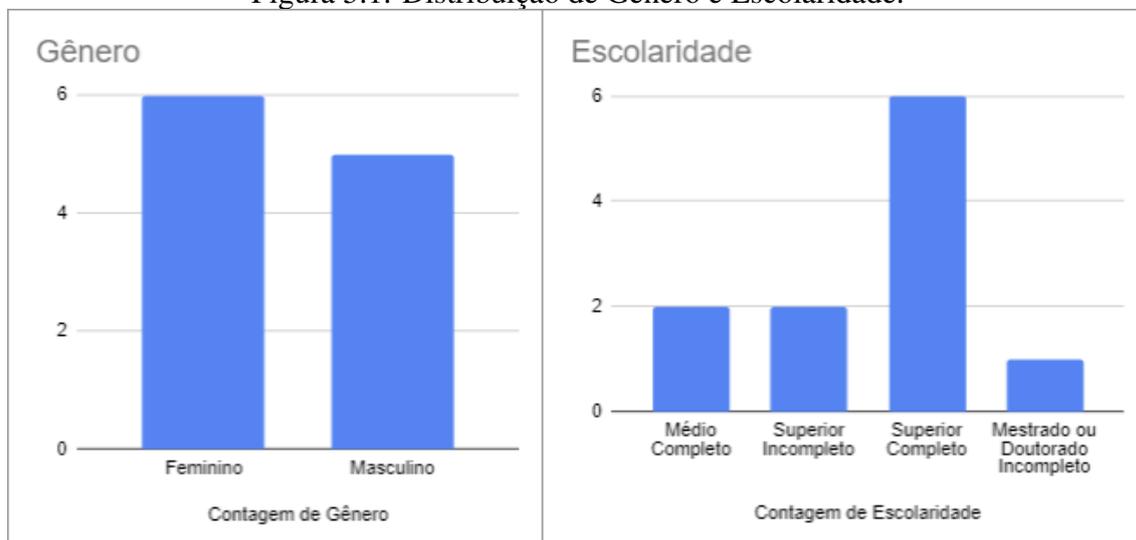
cordo Totalmente" contribui com 4 pontos. Isso converte as pontuações para uma escala de 0 a 4. Em seguida, multiplique a soma total por 2,5 para obter uma pontuação entre 0 e 100. Vale ressaltar que, embora a pontuação seja de 0 a 100, esta não é uma porcentagem e deve ser considerada apenas como uma escala de usabilidade.

5.2 Perfil dos Participantes

Para obter uma avaliação completa e abrangente da interface, é fundamental envolver uma diversidade de perfis de participantes. No entanto, é importante mencionar que, devido a limitações de tempo, não foi possível incluir indivíduos com níveis de escolaridade mais baixos ou faixas etárias mais avançadas no teste.

Quanto ao gênero dos participantes, foi observada uma distribuição de 6 indivíduos do sexo masculino e 5 do sexo feminino. Em relação à formação educacional, a maioria dos participantes possuía formação superior completa (7 participantes). Além disso, houve 2 participantes com ensino superior incompleto, 2 com ensino médio completo e 1 com mestrado ou doutorado incompleto.

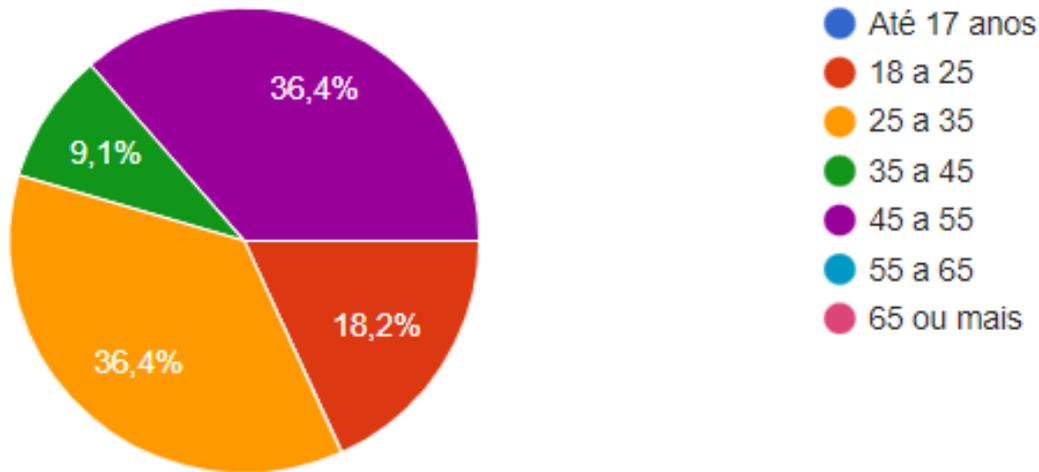
Figura 5.1: Distribuição de Gênero e Escolaridade.



Fonte: Dados coletados no formulário de avaliação.

A distribuição por faixa etária dos participantes foi a seguinte: 2 participantes tinham entre 18 e 25 anos, 4 estavam na faixa de 25 a 35 anos, 1 tinha entre 35 e 45 anos e 4 estavam na faixa de 45 a 55 anos.

Figura 5.2: Distribuição de Idade.



Fonte: Dados coletados no formulário de avaliação.

5.3 Resultados

A avaliação resultou em uma pontuação média do System Usability Scale (SUS) de 89.77, o que indica uma usabilidade bastante positiva. Essa avaliação positiva pode ser atribuída à simplicidade e objetividade do design do nosso sistema. A estrutura com apenas dois estados principais - o estado inicial de pesquisa e o estado de exibição das avaliações - foi elogiada por sua abordagem minimalista e intuitiva, facilitando a navegação dos usuários.

Na tabela 5.1 podemos ver as pontuações obtidas para cada pergunta. Onde 44 seria a pontuação máxima permitida.

Além disso, os comentários e sugestões fornecidos pelos participantes foram de grande valor. Eles destacaram a importância das seguintes melhorias:

Sugestões de Produtos e Preços: Foi sugerido que fossem incluídas sugestões de outros produtos relacionados e seus preços. Isso pode enriquecer a experiência do usuário, permitindo comparações mais eficazes.

Sistema de Likes e Classificação de Comentários: A ideia de coletar "likes" e classificar os comentários por relevância foi considerada útil para os usuários que desejam

Tabela 5.1: Resultado da aplicação do formulário de System Usability Scale

Pergunta	Valor
Eu acredito que eu gostaria de usar este sistema frequentemente.	42
Eu achei o sistema desnecessariamente complexo.	40
Eu achei o sistema fácil de usar.	41
Eu acredito que eu precisaria do suporte de uma pessoa técnica para conseguir usar este sistema.	38
Eu achei que as várias funções neste sistema estavam bem integradas.	36
Eu achei que havia muita inconsistência neste sistema.	41
Eu imagino que a maioria das pessoas aprenderia a usar este sistema muito rapidamente.	39
Eu achei o sistema muito pesado para usar.	39
Eu me senti muita confiança usando o sistema.	39
Eu precisei aprender muitas coisas antes de poder começar a usar este sistema.	40

Fonte: Dados coletados no formulário de avaliação.

avaliações mais significativas.

Inclusão de Imagens dos Produtos: A sugestão de adicionar imagens dos produtos foi destacada como uma maneira de influenciar a tomada de decisão do usuário.

Design Visual Colorido: Alguns participantes sugeriram tornar o aplicativo mais colorido, acreditando que isso poderia aumentar sua popularidade.

Os *feedbacks* gerais foram positivos, ressaltando a facilidade de uso e a confiabilidade percebida do sistema. Com base nessas avaliações, é possível observar a eficácia do sistema, mas também identificar áreas onde melhorias podem ser implementadas para aprimorar ainda mais a experiência dos usuários.

6 PROCESSO DE DESENVOLVIMENTO E CONCLUSÃO

6.1 Dificuldades Encontradas no Desenvolvimento

Durante o processo de desenvolvimento deste sistema, algumas dificuldades foram enfrentadas, impactando o progresso e as escolhas de implementação. A maior dificuldade surgiu no início, ao trabalhar com o *framework* Scrapy. A curva de aprendizado inicial foi bem elevada devido ao ambiente de desenvolvimento do scrapy não ser tão intuitivo, o que inclusive ocasionou a mudança para o framework Selenium para web scraping. No entanto, essa escolha resultou em um tempo de inicialização cerca de 8 vezes mais lento, atrasando muito a entrega inicial dos dados e prejudicando a experiência do usuário.

Ao perceber a desvantagem do Selenium, voltei ao Scrapy e, após superar essa barreira inicial, foram enfrentados desafios na integração com sites que possuíam APIs para coleta de dados. A busca por uma quantidade significativa de avaliações exigiu um tempo considerável de aprendizado e adaptação às particularidades de cada rota.

Após superar as dificuldades nas coletas dos dados, o desenvolvimento do *back-end* e *front-end* seguiu de forma mais tranquila. Porém, a implementação da comunicação contínua entre o servidor e o cliente por meio de *Server-Sent Events* (SSE) foi iniciada tardiamente. No início, estava sendo implementando comunicação via *web sockets*, até observar que SSE era uma solução mais adequada para a necessidade de comunicação contínua do servidor.

6.2 Limitações do Sistema e Próximos Pontos de Desenvolvimento

O sistema atual foi desenvolvido para ser uma prova de conceito e possui limitações em relação ao número de sites cadastrados. Abrangendo apenas três sites: dois que fornecem grandes quantidades de avaliações via *requests* paginados e um estático.

A próxima funcionalidade seria a criação de um banco de dados que faça a persistência de avaliações mais relevantes. Permitindo que os usuários interajam com a plataforma marcando os comentários como úteis ou não. Possibilitando assim, a entrega de dados relevantes com muito mais velocidade. Sendo necessário apenas uma consulta ao banco para fornecer resposta ao usuário.

Outra melhoria possível é dividir o processo de coleta e envio em partes meno-

res, acelerando o envio das avaliações. A arquitetura atual não prioriza a escalabilidade, tornando-se um foco para futuros desenvolvimentos. Atualmente será necessário implementar limitações de execução que protejam o servidor para o uso de muitos usuários ao mesmo tempo, pelo menos em estágios iniciais de *deploy*.

Outro ponto de melhoria é oferecer ao usuário sugestões de produtos durante a digitação a fim de que ele selecione o produto em específico, diminuindo assim a chance das *spiders* coletarem informações do produto errado.

Além de trabalhar nas sugestões propostas nos *feedbacks*. Que vão desde novas funcionalidades até melhorias de layout.

6.3 Conclusão

Este projeto buscou desenvolver uma plataforma que auxilie os consumidores em suas decisões de compra, expondo as características dos produtos com base nas avaliações de usuários reais. O processo de desenvolvimento envolveu o conhecimento e a integração de diversas tecnologias atuais, consolidando o aprendizado e a prática obtidos durante o curso.

O sistema espera contribuir para a experiência do usuário ao fornecer informações transparentes e úteis sobre os produtos em questão, apoiando uma decisão de compra mais informada e confiante.

REFERÊNCIAS

Solomon, M. R. *O Comportamento do Consumidor - 11ed: Comprando, Possuindo e Sendo*, N.p.: Bookman Editora, 2016.

Mitchell, Ryan. *Web Scraping with Python: Collecting Data from the Modern Web*, Estados Unidos: O'Reilly Media, 2015.

EQI - Qualidade do E-commerce Brasil, <<https://dcx.lett.digital/eqi-qualidade-do-ecommerce-brasil/>>, 2023.

Avaliações de Produtos na Rede, <<https://www.capterra.com.br/blog/848/avaliacoes-de-produtos-na-rede>>, 2023.

O que torna as avaliações dos clientes confiáveis para o consumidor, <<https://www.ecommercebrasil.com.br/artigos/o-que-torna-as-avaliacoes-dos-clientes-confiaveis-%E2%80%8B%E2%80%8Bpara-o-consumidor>>, 2023.

Avaliação de produto: por que faz a diferença no e-commerce, <<https://www.ecommercebrasil.com.br/artigos/avaliacao-de-produto-por-que-faz-a-diferenca-no-e-commerce>>, 2023.

Avaliações dos Consumidores, <<https://www.profissionaldeecommerce.com.br/avaliacoes-dos-consumidores/>>, 2023.

Evan You. *Vue.js: The Progressive JavaScript Framework*, Vue.js Developers, 2023.

Hickson, Ian. *Server-Sent Events*, W3C Recommendation, 2011. <<https://www.w3.org/TR/eventsource/>>

Holowaychuk, T.J. et al. *Express.js*, Fast, unopinionated, minimalist web framework for Node.js, 2023. <<https://expressjs.com/>>

Ryan Dahl. *Node.js*, Node.js Foundation, 2023. <<https://nodejs.org/>>

Tondreau, Beth. *Layout Essentials: 100 Design Principles for Using Grids*, Rockport Publishers, 2009.

McFarland, David. *JavaScript & jQuery: The Missing Manual*, O'Reilly Media, 2015.

Kesteren, Anne van. *Cross-Origin Resource Sharing*, W3C Recommendation, 2014. <<https://www.w3.org/TR/cors/>>

Joyent, Inc. *Child Processes in Node.js*, Node.js v14.17.1 Documentation, 2023.
<https://nodejs.org/api/child_process.html>

Mozilla. *EventSource Web API*, EventSource Documentation, 2023. <<https://developer.mozilla.org/en-US/docs/Web/API/EventSource>>

Brooke, John. *SUS - A quick and dirty usability scale*, Redhatch Consulting Ltd, 1996.