

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**Trabalho de Conclusão de Curso**

**APLICAÇÃO WEB PARA DETECÇÃO E NOTIFICAÇÃO DE PROMOÇÕES COM WEB  
SCRAPING E DISPARO DE E-MAIL**

**Autor**

Vítor de Souza Cunha

**Orientador**

Jose Eduardo de Lucca

**Florianópolis**

**2023/1**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
TRABALHO DE CONCLUSÃO DE CURSO**

**APLICAÇÃO WEB PARA DETECÇÃO E NOTIFICAÇÃO DE PROMOÇÕES COM WEB  
SCRAPING E DISPARO DE E-MAIL**

Submetido ao corpo docente do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina como um dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

**Orientador:**

---

Prof. Jose Eduardo de Lucca

## RESUMO

É de conhecimento geral que há um grande crescimento no mercado de compras e vendas pela internet, especialmente em tempos de pandemia, onde o home office se tornou uma realidade para muitos e o investimento para o trabalho feito em casa, aumentou consideravelmente. Porém, apesar do cenário propício para compra, muitas empresas de venda online adotam estratégias de marketing e variações de preço que tendem a iludir o consumidor, que acaba não encontrando um preço justo para os itens que procura. Considerando isso, este trabalho se propõe a desenvolver uma aplicação web que irá fazer uma varredura pelos itens especificados pelo usuário, em dois websites sem fins lucrativos, contendo as melhores promoções, e irá disparar e-mails ao usuário, alertando se um (ou mais) de seus itens de desejo foi encontrado. Os websites em questão são alimentados pela própria comunidade que busca e compartilha promoções. As promoções são postadas por qualquer usuário cadastrado nos mesmos e com a finalidade de ajudar potenciais consumidores, sem qualquer tipo de interesse financeiro ou parcerias. Por fim, a proposta de criação desta aplicação web foi atingida e a varredura dos websites, bem como a notificação de e-mails se mostraram eficazes, cumprindo os objetivos propostos neste trabalho.

**Palavras-chave:** Aplicação Web. Web Scraping. Varredura e promoções. Home Office. Vendas Online.

## ABSTRACT

It is common knowledge that there is great growth in the internet shopping and sales market, especially in times of a pandemic, where the home office has become a reality for many and the investment for work done at home has increased considerably. However, despite the favorable scenario for buying, many online sales companies adopt marketing strategies and price variations that tend to deceive the consumer, who ends up not finding a fair price for the items he is looking for, especially in the hardware market. Considering this, this work proposes to develop a web application that will scan the items specified by the user, in two non-profit websites, containing the best promotions, and will send e-mails to the user, alerting if one (or more) of your wish items has been found. The websites in question are powered by the community that seeks and shares promotions. The promotions are posted by any user registered in them and with the purpose of helping potential consumers, without any type of financial interest or partnerships. Finally, the proposal for creating this web application was achieved and the scanning of websites, as well as the notification of emails, proved to be effective, fulfilling the objectives proposed in this work.

**Keywords:** Web application. Web Scraping. Scanning and promotions. Home Office. E-commerce.

## LISTA DE TABELAS

<b>Tabela 01</b> - Comparativo Web Scraping e Web Crawling.....	26
<b>Tabela 02</b> - Campos do objeto User no banco de dados.....	45
<b>Tabela 03</b> - Campos do objeto Match no banco de dados.....	46

## LISTA DE FIGURAS

<b>Figura 01</b> - Gráfico da previsão de quantidade de pedidos (milhões) no e-commerce para os próximos 5 anos.....	12
<b>Figura 02</b> - Gráfico do perfil das categorias de e-commerce por faturamento.....	13
<b>Figura 03</b> - Gráfico dos fatores que influenciam os consumidores brasileiros a realizar compras online.....	15
<b>Figura 04</b> - Ciclo de vida da prototipação.....	17
<b>Figura 05</b> - Evolução na quantidade de compradores no e-commerce.....	20
<b>Figura 06</b> - Gráfico do histórico de preços de um produto na plataforma Zoom.....	33
<b>Figura 07</b> - Funcionalidades da plataforma Zoom (extensão).....	34
<b>Figura 08</b> - Exemplo de gráficos de descontos reais e falsos JáCotei.....	35
<b>Figura 09</b> - Modelo de negócio JáCotei.....	36
<b>Figura 10</b> - Diagrama de casos de uso.....	40
<b>Figura 11</b> - Modelo MTV.....	41
<b>Figura 12</b> - Comunicação do Modelo MTV entre Cliente e Servidor.....	43
<b>Figura 13</b> - Diagrama entidade relação do banco de dados.....	46
<b>Figura 14</b> - Visão geral do sistema e funcionamento dos Heroku dynos e deploys...	56
<b>Figura 15</b> - Projeto django criado com sucesso e rodando localmente.....	70
<b>Figura 16</b> - Tela de Login/Cadastro.....	71
<b>Figura 17</b> - Formulário de cadastro.....	72
<b>Figura 18</b> - Página inicial.....	73
<b>Figura 19</b> - Página principal Promo Scraper.....	74
<b>Figura 20</b> - Mouse em cima do botão explicativo.....	75
<b>Figura 21</b> - Website Gatry.....	76
<b>Figura 22</b> - Algoritmo de scraping Gatry.....	77
<b>Figura 23</b> - Output da execução algoritmo varredura Gatry.....	77
<b>Figura 24</b> - Função para ajustar preço encontrado Gatry.....	78
<b>Figura 25</b> - Função para lidar com cupom.....	79
<b>Figura 26</b> - Website Boletando.....	80
<b>Figura 27</b> - Algoritmo de scraping Boletando.....	81
<b>Figura 28</b> - Output da execução algoritmo varredura Gatry.....	82

<b>Figura 29</b> - Página inicial mostrando objetos do banco de dados.....	84
<b>Figura 30</b> - Painel django admin listando objetos Match.....	85
<b>Figura 31</b> - Painel do heroku mostrando o histórico de builds.....	87
<b>Figura 32</b> - Promo Scraper rodando em produção e responsivo (iphone).....	88
<b>Figura 33</b> - Mensagem de e-mail com uma promoção encontrada.....	91

## LISTA DE ABREVIACOES

UFSC - Universidade Federal de Santa Catarina

API - Application Programming Interface

MTV - Model Template View

DLT - Django Template Language

HTTP - Hypertext Transfer Protocol

HTTPS - Hypertext Transfer Protocol Secure

MVC - Model View Controller

HTML – HyperText Markup Language

XML - Extensible Markup Language

CSS – Cascading Style Sheets

JS - JavaScript

IOS - Iphone Operating System

ACID - Atomicity, Consistency, Isolation, Durability

LGPD - Lei Geral de Proteo dos Dados

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>12</b>
1.1 JUSTIFICATIVA.....	14
1.2 METODOLOGIA.....	16
1.3 OBJETIVOS.....	18
1.3.1 OBJETIVO PRINCIPAL.....	18
1.3.2 OBJETIVOS ESPECÍFICOS.....	18
<b>2 FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICA.....</b>	<b>19</b>
2.1 E-COMMERCE E SUA EVOLUÇÃO NO CENÁRIO DA PANDEMIA COVID-19.....	19
2.2 APLICAÇÃO WEB.....	21
2.3 WEB SCRAPING.....	22
2.3.1 WEB SCRAPING E WEB CRAWLING.....	24
2.4 LEGALIDADE E ÉTICA AO REALIZAR WEB SCRAPING.....	25
2.4.1 UM POUCO SOBRE A LEI GERAL DE PROTEÇÃO DE DADOS (LGPD).....	26
2.5 CROWDSOURCING.....	28
<b>3 SOLUÇÕES EXISTENTES.....</b>	<b>29</b>
3.1 COMPARADOR DE PREÇOS ZOOM.....	30
3.2 COMPARADOR DE PREÇOS JÁCOTEI.....	31
3.3 ANÁLISE DAS SOLUÇÕES EXISTENTES.....	32
3.3.1 VANTAGENS.....	32
3.3.2 DESVANTAGENS.....	35
<b>4 SOLUÇÃO PROPOSTA.....</b>	<b>37</b>
4.1 REQUISITOS FUNCIONAIS.....	37
4.2 REQUISITOS NÃO FUNCIONAIS.....	38
4.3 DIAGRAMA DE CASOS DE USO.....	39
4.4. ANÁLISE.....	40
4.5 ARQUITETURA DO SISTEMA.....	41
4.5.1 ENTIDADES DO BANCO DE DADOS.....	43

4.6 TECNOLOGIAS UTILIZADAS.....	46
4.6.1 BANCO DE DADOS.....	46
4.6.1.1 BANCO DE DADOS LOCAL.....	46
4.6.1.2 BANCO DE DADOS EM PRODUÇÃO.....	47
4.6.2 FRAMEWORK DJANGO.....	47
4.6.3 BOOTSTRAP.....	49
4.6.4 BIBLIOTECA BEAUTIFULSOUP.....	49
4.6.5 EZGMAIL.....	51
4.6.6 HEROKU.....	51
4.6.6.1 HEROKU SCHEDULER.....	52
4.6.6.2 DEPLOY AUTOMÁTICO HEROKU E GITHUB.....	53
<b>5 DESENVOLVIMENTO.....</b>	<b>54</b>
5.1 IMPLEMENTAÇÃO.....	54
5.1.1 DESCRIÇÃO GERAL DO SISTEMA.....	54
5.1.2 SPRINTS.....	56
5.1.2.1 SPRINT 1 - ESTRUTURAÇÃO DO PROJETO.....	56
5.1.2.2 SPRINT 2 - USUÁRIOS E AUTENTICAÇÃO.....	56
5.1.2.3 SPRINT 3 - FRONT END PÁGINA INICIAL.....	57
5.1.2.4 SPRINT 4 - ALGORITMO DE VARREDURA GATRY.....	57
5.1.2.5 SPRINT 5 - ALGORITMO DE VARREDURA BOLETANDO.....	57
5.1.2.6 SPRINT 6 - POPULANDO O BANCO E MOSTRANDO NO FRONT END.....	57
5.1.2.7 SPRINT 7 - PROJETO EM PRODUÇÃO (LIVE) HEROKU.....	57
5.1.2.8 SPRINT 8 - SERVIÇO SCHEDULER.....	57
<b>6 RESULTADOS OBTIDOS.....</b>	<b>58</b>
<b>7 CONCLUSÃO.....</b>	<b>60</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>62</b>
<b>APÊNDICE A - SPRINTS.....</b>	<b>69</b>

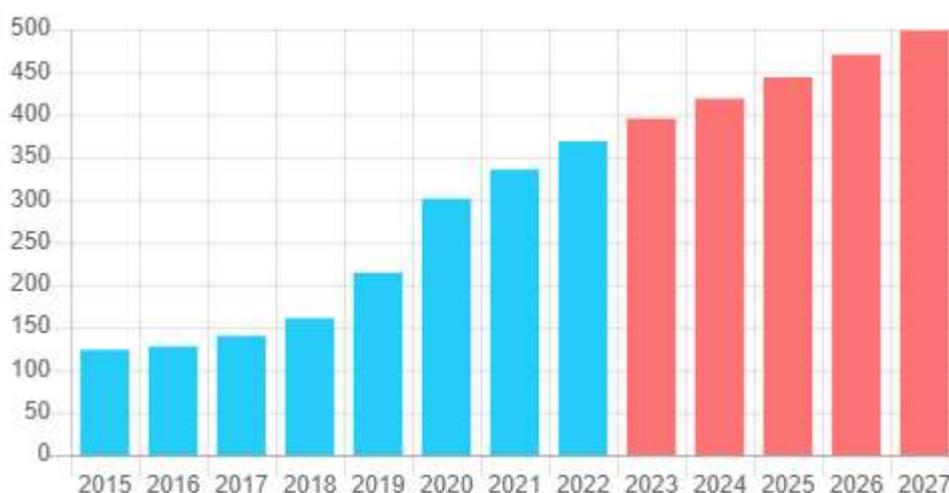
<b>APÊNDICE B - ARTIGO DO TRABALHO.....</b>	<b>92</b>
<b>APÊNDICE C - CÓDIGO DA APLICAÇÃO.....</b>	<b>101</b>

## 1 INTRODUÇÃO

No momento em que o estudo foi iniciado, o Brasil e o mundo todo enfrentavam um período extremamente difícil por conta da pandemia causada pelo vírus Covid-19. Por conta disso, muitas empresas, especialmente na área de tecnologia, adotaram o *home office* como modelo de trabalho durante o isolamento.

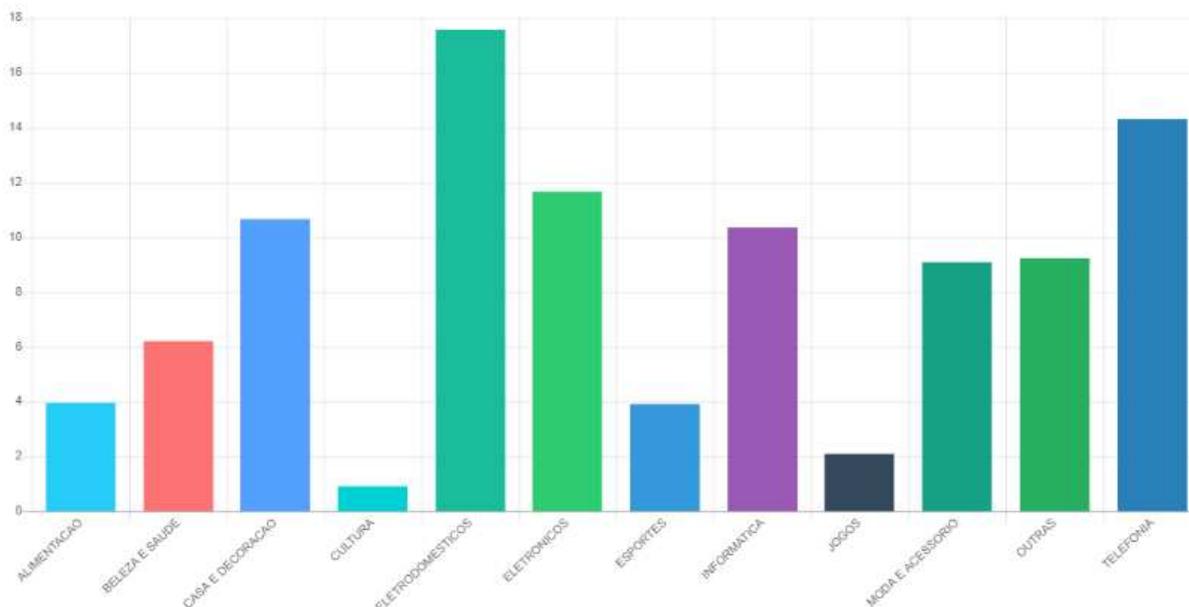
Consequentemente, o investimento em materiais de trabalho aumentaram, e continuam aumentando, drasticamente, e um desses setores é o de eletrônicos, que apresenta um grande crescimento, como mostram os dados da Associação Brasileira de Comércio Eletrônico (ABComm).

**Figura 01 - Gráfico da previsão de quantidade de pedidos (milhões) no e-commerce para os próximos 5 anos:**



Fonte: dados ABComm, 22 fev. 2023.

**Figura 02 - Gráfico do perfil das categorias de e-commerce por faturamento:**



Fonte: dados ABComm, 22 fev. 2023.

De acordo com os gráficos apresentados na figura 2, pode-se perceber o aumento nas vendas por meio eletrônico e as áreas que mais impactaram-se com este aumento. A possibilidade de comprar sem se expor ao risco de contágio, e o acesso a diversos produtos de modo rápido e instantâneo inserido num contexto pandêmico são de grande importância para explicar este movimento.

Nesse cenário, o consumidor costuma procurar as melhores ofertas em toda gama de *e-commerces* que lhe interessam. Existem algumas alternativas, procurar manualmente em cada loja de seu interesse e confiança, ou recorrer a uma plataforma que compara automaticamente o preço entre todas as lojas.

Algumas empresas fazem este tipo de serviço, como: Buscapé, Zoom, Bondfaro, entre outras. Embora atualmente, todas façam parte da Zoom, após sua aquisição em 2019:

Nesta segunda-feira (13), a plataforma de comércio eletrônico Zoom fechou o negócio com a Naspers para a aquisição do seu concorrente Buscapé e de outras companhias do grupo. O acordo inclui as marcas Bondfaro, QueBarato e Modait. Os valores e termos da compra não foram divulgados. (TUDOCELULAR.COM, 2019, texto online)

Não obstante, a solução proposta por este trabalho tem uma abordagem diferente. O usuário primeiro irá inserir palavras-chave num formulário, que serão os itens a serem procurados, para ser alertado apenas quando o algoritmo encontrar um ou mais desses itens em alguma publicação nos sites em que está sendo feita a varredura, sendo então, uma busca passiva.

A vantagem deste modelo é que o usuário terá acesso a promoções que, muitas vezes, estão disponíveis apenas por tempo limitado e tendem a encerrar rapidamente. Como trata-se de uma varredura feita em sites geridos por usuários comuns e que também são consumidores, há a garantia de que não haverá interesse por parte destes.

É uma forma de somar esforços de toda uma comunidade que pode estar à procura dos mesmos itens de interesse do usuário dessa aplicação em questão, e reportar com prontidão da devida oferta.

## 1.1 JUSTIFICATIVA

Desde quando a internet se popularizou, as aplicações web e mobile tornaram-se parte da rotina das pessoas, pois são capazes de auxiliar nas resoluções de problemas do dia-a-dia. Segundo Holding (2016) quando se trata de procurar informações, acessar arquivos ou se comunicar, 64% dos brasileiros usam aplicativos móveis.

Além disso, surgem as plataformas que permitem a divulgação de promoções e ofertas para compras online, garantindo, além do conforto e comodidade na realização da compra, a aquisição com o melhor preço. Porém, há dificuldade em encontrar uma plataforma que ofereça a divulgação de promoções e descontos de forma que a confiabilidade, transparência e personalização na procura sejam garantidas.

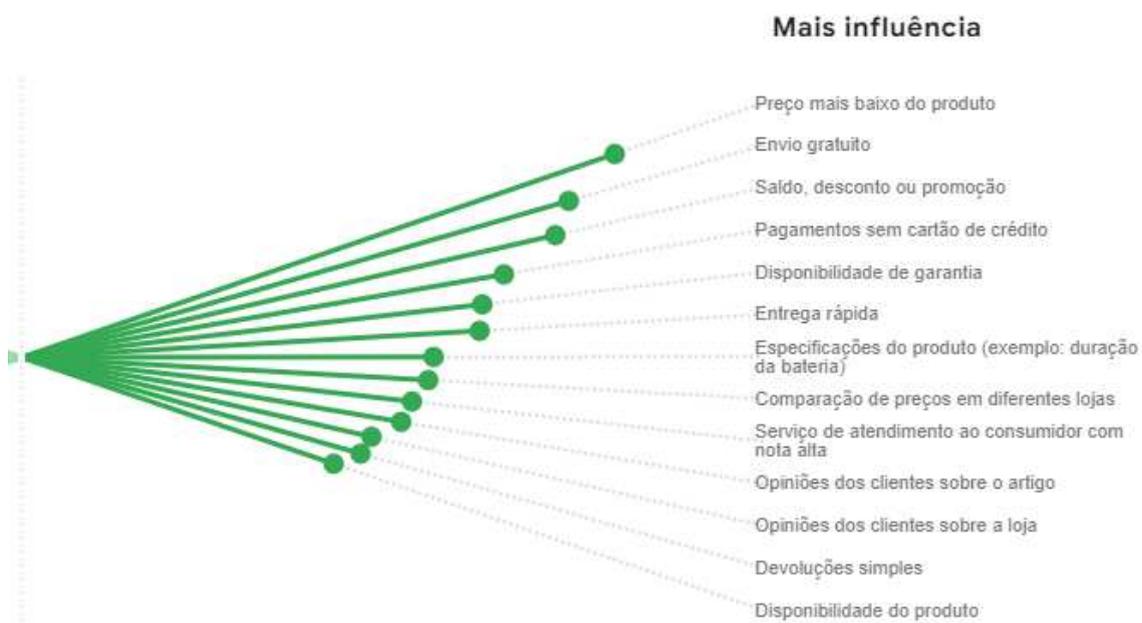
Ademais, pode-se notar um grande crescimento no setor do *e-commerce*. Um estudo conduzido pela ABCOMM em 2022, mostra que o segmento de e-commerce faturou 83.8 milhões em 2021, um crescimento de aproximadamente 5% em comparação com o ano anterior.

Um levantamento encomendado pela Google e conduzido pela Offerwise em 2023, mostra que nove em cada dez brasileiros, após a pandemia, passaram a pesquisar antes de adquirir um produto. Além disso, a pesquisa mostra que são também

os consumidores que mais gastam tempo pesquisando online antes de efetuar uma compra.

É por isso que, um dos fatores que mais influenciam os consumidores brasileiros a realizar compras online é o preço mais baixo do produto, assim como saldos e promoções (THINK WITH GOOGLE, 2022):

**Figura 03 - Gráfico dos fatores que influenciam os consumidores brasileiros a realizar compras online**



Fonte: Think with Google, 2022.

Todas as informações nos mostram, que os consumidores estão mais exigentes quando se trata de suas compras online, e que o mercado de e-commerce vem acompanhando o crescimento do setor de TI e similares.

A justificativa para o presente trabalho encontra-se na necessidade de usar técnicas de web scraping para extrair automaticamente as informações e, posteriormente, reutilizá-las para alertar os usuários destas possíveis promoções, que como citado na figura 03, tem um grande impacto na decisão do consumidor de comprar

um ou mais produtos. A varredura e disparo de e-mail, caso um ou mais itens sejam encontrados, darão ao consumidor a oportunidade de acessar uma promoção momentos após seu lançamento, o que aumenta significativamente as chances de que o item desejado ainda esteja disponível em estoque.

Tal tecnologia mostra-se importante uma vez que as promoções são postadas por qualquer usuário cadastrado em um site e com a finalidade de ajudar potenciais consumidores, sem qualquer tipo de interesse financeiro ou parcerias.

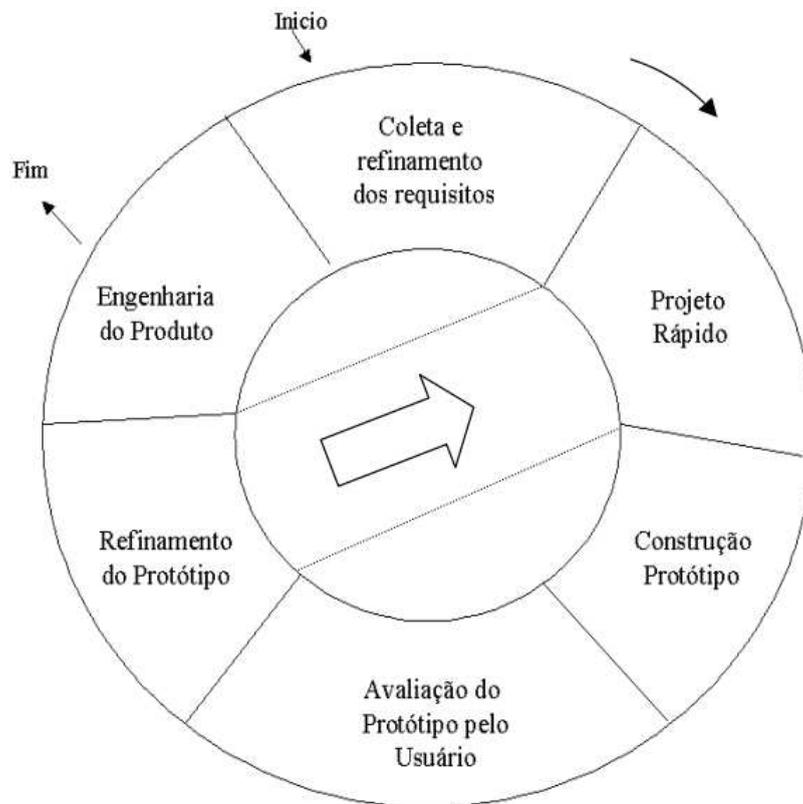
## 1.2 METODOLOGIA

O método de pesquisa utilizado neste estudo é um método de pesquisa de desenvolvimento com um modelo de Prototipação.

No ciclo de vida da prototipação, a definição do sistema ocorre através da descoberta gradual e evolutiva deste por parte do usuário e do desenvolvedor. Assim, obtém-se um conjunto inicial de necessidades e as implementam rapidamente com a intenção de refiná-las de acordo com o aumento do conhecimento do sistema por parte do desenvolvedor e do usuário (YOURDON, 1990).

Os estágios do modelo utilizado são coleta e refinamento de requisitos, projeto rápido, construção do protótipo, avaliação do protótipo, refinamento do protótipo, e por fim, engenharia do produto, como visto na Figura 4.

**Figura 04 - Ciclo de vida da prototipação**



Fonte: ALVES, 2011.

Com base nos problemas que foram estudados e descritos em segundo plano e para o cumprimento dos objetivos, o desenvolvimento do sistema seguirá as seguintes etapas de metodologia:

- Realizar uma análise do mercado e do cenário atual de e-commerce;
- Realizar uma avaliação das plataformas similares a fim de identificar oportunidades de melhoria;
- Criar um sistema web que tenha como principal objetivo ajudar a comunidade de consumidores na compra de produtos (com foco em tecnologia);
- Análise, modelagem e construção do sistema de forma incremental, utilizando uma abordagem ágil adaptada com base no SCRUM;

- Publicação do sistema e funcionamento da ferramenta disponível para o público gratuitamente;

## 1.3 OBJETIVOS

### 1.3.1 Objetivo Principal

Desenvolver uma aplicação web para ajudar os consumidores a encontrarem as melhores promoções de forma passiva (sem a necessidade de estar ativamente procurando), utilizando *web scraping* em sites de promoções *crowdsourcing* e notificando via e-mail quando uma ou mais promoções forem encontradas.

### 1.3.2 Objetivos Específicos

- Realizar uma análise do comportamento e tendências de pesquisas e compras online;
- Estudar e analisar ferramentas de comparativos de preços;
- Estudar e analisar ferramentas de notificação de produtos e preços;
- Analisar, modelar e construir um sistema web de forma incremental, utilizando uma abordagem ágil adaptada com base no SCRUM;
- Desenvolver a solução modelada, com uma interface web para visualização de promoções encontradas e notificar promoções encontradas via e-mail, salvando-as no banco de dados

## 2 FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICA

Para tornar o conjunto de tecnologias e conceitos abordados no desenvolvimento deste trabalho mais compreensível, serão apresentados neste capítulo os conceitos fundamentais para o melhor entendimento do desenvolvimento do trabalho em questão.

### 2.1 E-COMMERCE E SUA EVOLUÇÃO NO CENÁRIO DA PANDEMIA COVID-19

Atualmente, a internet é uma ferramenta fundamental no cotidiano das pessoas, realizando diversas atividades como o uso do comércio de bens e serviços, a realização de transações financeiras, a interação com o poder público e a busca de informações sobre saúde, educação, entre outras.

Um dos negócios em crescimento na era da internet é o e-commerce, definido como o processo de comércio e troca de bens, serviços e informações (JORNAL DO COMÉRCIO, 2020). É um negócio moderno que atende às necessidades das organizações, mercadorias e consumidores para reduzir custos e melhorar a qualidade dos bens e serviços e aumentar a velocidade do serviço de entrega.

Esse tipo de comércio trouxe certos benefícios para os consumidores como: acesso a produtos globalmente, possibilidade de comparação de produtos e preços e redução de tempo. De acordo com informações do IBGE (2022), a Internet chega a 90,0% dos domicílios do país em 2021, com alta de 6 pontos percentuais (p.p.) frente a 2019, quando 84,0% dos domicílios tinham acesso à grande rede.

A mesma pesquisa revelou, ainda, que o celular é o dispositivo mais utilizado para acessar a Internet em casa, representando 99,5%. Já a televisão foi o segundo equipamento mais utilizado para esse fim (44,4%), ultrapassando, pela primeira vez, o computador (42,2%).

O celular é um dispositivo multifuncional que integra chamadas, mensagens, conteúdo multimídia e pode ser usado como reprodutor de música, câmera e computador, além disso, de acordo com o estudo IDC Brazil Mobile Phone Tracker Annual (2022), da IDC Brasil, o número de celulares vendidos no Brasil em 2022 foi de 42,6 milhões de aparelhos no total, e a receita total gerada ficou em R\$ 77,09 bilhões.

É fundamental olhar para o cenário e o impacto que a pandemia COVID-19 trouxe no comportamento do consumidor, bem como dados que comprovam um aumento considerável no número de compras em *e-commerces*.

O relatório Webshoppers 47ª edição, realizado e divulgado recentemente pela Ebit/Nielsen identifica mudanças no comportamento do consumidor.

Na interação de compradores e vendedores online, as mensagens de e-mail e SMS se destacam como os principais canais de comunicação. A época do ano em que os consumidores mais se voltam para o comércio on-line é o Natal (NIQ EBIT, 2023). Entre os problemas enfrentados pelo comércio on-line, segundo a mesma pesquisa, está a privacidade dos dados pessoais, onde os consumidores apontam a desconfiança da segurança do processo de compra como roubo de identidade ou dados de cartão.

A pandemia de Covid-19 causou um aumento nas compras e vendas on-line, aumentando 81% em 2020 em relação a 2019. Da mesma forma, a Covid-19 mudou os hábitos dos consumidores diante das restrições de saúde implementadas pelo governo federal, já que as pessoas optaram por fazer suas compras online, devido à indisponibilidade de produtos nas lojas físicas; a possibilidade de receber os seus produtos em casa; economizar tempo e evitar aglomerações em lojas físicas (BANHARA, 2021).

**Figura 05 – Evolução na quantidade de compradores no e-commerce**



A pandemia de Covid-19 favoreceu o crescimento do comércio eletrônico no Brasil e no mundo como resultado das medidas de prevenção do vírus e do isolamento social da população (EBIT, 2021). Este aumento de demanda evidencia a mudança nos hábitos dos consumidores.

No entanto, grandes segmentos da população não têm acesso ou utilizam o comércio online devido a vários fatores relacionados com o fosso digital. Alguns desses fatores são: rendimentos insuficientes, baixos níveis de educação, origem étnica, sexo, local de residência rural ou urbano, falta de competências digitais, entre outros (JORNAL DO COMÉRCIO, 2020).

Além do fosso digital, outros fatores têm grande interferência na expansão do comércio digital, como o alto nível de empregos informais (EBIT, 2021). Já os principais obstáculos enfrentados pelo comércio online está a desconfiança dos usuários em aspectos como segurança (fraude cibernética e roubo de dados pessoais), formas de pagamento e falta de informação.

Portanto, é necessário estabelecer leis e regulamentos para regular o comércio on-line e proteger os consumidores. Vale ressaltar que esse tema tem sido pouco estudado, devido à falta de dados microeconômicos que permitam uma análise detalhada desse fenômeno multidimensional (BANHARA, 2021).

## 2.2 APLICAÇÃO WEB

Para a solução do problema deste trabalho, foi escolhida a construção de uma aplicação web, mas para compreender essa decisão, se faz necessário entender o conceito dela.

Uma aplicação Web é uma aplicação chamada com um navegador da web pela Internet. Desde 1994, quando a internet tornou-se disponível ao público, e especialmente em 1995, quando a *world wide web* colocou uma face utilizável na internet, tornou-se uma plataforma de escolha para um grande número de sites cada vez mais sofisticados e formulários inovadores (JAZAYERI, 2007).

Uma aplicação da web é um software executado em seu navegador da web. As empresas precisam trocar informações e fornecer serviços remotamente. Eles usam aplicativos da web para se conectar com os clientes de forma conveniente e segura. Os recursos mais comuns do site, como carrinhos de compras, pesquisa e filtragem de produtos, mensagens instantâneas e feeds de notícias de mídia social, são aplicativos da Web em seu design. Eles permitem que você acesse funcionalidades complexas sem instalar ou configurar software (AMAZON AWS, 2023).

Existem diversos benefícios de utilizar uma aplicação web, no próprio site oficial da Amazon AWS, tais como:

**Acessibilidade:** Os aplicativos da Web podem ser acessados de todos os navegadores da Web e em vários dispositivos pessoais e empresariais. Equipes em locais diferentes podem acessar documentos compartilhados, sistemas de gerenciamento de conteúdo e outros serviços de negócios por meio de aplicativos da Web baseados em assinatura.

**Desenvolvimento eficiente:** Conforme detalhado, o processo de desenvolvimento de aplicativos da Web é relativamente simples e econômico para as empresas. Equipes pequenas podem atingir ciclos de desenvolvimento curtos, tornando os aplicativos da Web um método eficiente e acessível de criar programas de computador. Além disso, como a mesma versão funciona em todos os navegadores e dispositivos modernos, você não precisará criar várias iterações diferentes para várias plataformas.

**Simplicidade para o usuário:** Os aplicativos da Web não exigem que os usuários os baixem, facilitando o acesso e eliminando a necessidade de manutenção do usuário final e capacidade do disco rígido. Os aplicativos da Web recebem automaticamente atualizações de software e segurança, o que significa que estão sempre atualizados e correm menos riscos de violações de segurança.

**Escalabilidade:** As empresas que usam aplicativos da web podem adicionar usuários conforme necessário, sem infraestrutura adicional ou hardware caro. Além disso, a grande maioria dos dados de aplicativos da web é armazenada na nuvem, o que significa que sua empresa não precisará investir em capacidade de armazenamento adicional para executar aplicativos da web.

## 2.3 WEB SCRAPING

Dado o volume, variedade, velocidade e veracidade do grande volume de dados disponíveis na Web, a coleta e a organização desses dados dificilmente pode ser feita manualmente por pesquisadores individuais ou mesmo por grandes times. Por conta disto, os pesquisadores frequentemente recorrem a diversas tecnologias e ferramentas para automatizar alguns aspectos da coleta e organização de dados. Esta prática emergente de usar a tecnologia de coleta de dados da Web costuma ser chamada de Web Scraping (KROTOV, 2018).

*Web scraping, web harvesting* ou *web data extraction* é a coleta de dados usada para extrair dados de sites. O software de web scraping pode acessar diretamente a World Wide Web usando o protocolo Hypertext Transfer Protocol (HTTP) ou um navegador da web. (BARREIRA, 2014).

O Web Scraping consiste nas seguintes fases principais e interligadas: análise do site, rastreamento do site e organização de dados. Cada uma das fases do *Web Scraping* requerem compreensão de várias tecnologias da Web e pelo menos uma linguagem de programação popular usada em Data Science, como R ou Python. O problema é que, muitas vezes, essas três fases do Web Scraping não podem ser totalmente automatizadas e geralmente requerem pelo menos algum grau de envolvimento humano (KROTOV, 2018).

A análise do site requer o exame da estrutura subjacente de um site (ou sites) ou de um repositório (por exemplo, um banco de dados online) com a finalidade de entender como os dados necessários são armazenados. Isso requer uma compreensão básica da arquitetura da World Wide Web; linguagens de marcação (por exemplo, HTML, CSS, XML, XBRL, etc); e vários bancos de dados da Web (por exemplo, MySQL). O rastreamento de sites envolve o desenvolvimento e a execução de um script que navega automaticamente no site e recupera os dados necessários. Esses aplicativos de rastreamento (ou scripts) geralmente são desenvolvidos usando tais linguagens de programação como R e Python. Isso tem a ver com a popularidade geral dessas linguagens em Data Science e disponibilidade de bibliotecas (por exemplo, pacote “rvest” em R ou biblioteca Beautiful Soup em Python) para rastreamento automático e análise de dados da Web (KROTOV, 2018).

No início dos anos 2000 viu a introdução de um dos primeiros bots de raspagem da web de comércio eletrônico potencialmente maliciosos, chamado Bidder's Edge, que

agregava preços de competir entre sites de leilões. As empresas on-line devem implementar uma proteção técnica eficaz contra bots e medidas de detecção de scraper contra a raspagem da web (BARREIRA, 2014).

### 2.3.1 WEB SCRAPING E WEB CRAWLING

A coleta de dados não envolve necessariamente a web. Ferramentas de *Data scraping* que ajudam na busca de dados podem referir-se à extração de informações de uma máquina local, um banco de dados. Mesmo que seja da internet, um mero link “Salvar como” na página também é um subconjunto do universo de coleta de dados. (GRACIANO, HELTON. 2022)

*Data crawling*, por outro lado, difere em escala e em alcance. Em primeiro lugar, *crawling* se refere a *web crawling*, o que significa que na web, podemos apenas “rastrear” dados (GRACIANO, HELTON. 2022).

*Web crawling* pode ser visto como uma extensão de *web scraping* que não apenas extrai os dados de uma página da Web, mas também segue os links na página da Web e salta para várias páginas realizando extração de dados da web simultaneamente. *Web scraping* limita-se a raspar uma determinada página da Web, enquanto o *web crawling* extrai os dados de todas as páginas da web que podem ser acessadas de uma determinada página, portanto, grandes quantidades de dados podem ser extraídas usando um *crawler* em um único processo automatizado.

Um exemplo de *web crawler* que executa a extração de dados em grande escala pode ser “um mecanismo de pesquisa do Google”. Um mecanismo de pesquisa, também chamado de aranhas do mecanismo de pesquisa (*search engine spiders*) são programas de software escritos por humanos que podem automaticamente e constantemente visitar milhões de sites todos os dias e incluem o que encontram, em bancos de dados de mecanismos de pesquisa. Esse processo é chamado de *crawling* ou *spidering* (NIGAM, HARSHIT. 2020).

**Tabela 01 - Comparativo Web Scraping e Web Crawling**

<b>WEB SCRAPING</b>	<b>WEB CRAWLING</b>
<b>Envolve a extração de dados de várias fontes da web.</b>	<b>Refere-se a fazer o download de páginas da web.</b>
<b>Pode ser feito em qualquer escala.</b>	<b>Geralmente feito em larga escala.</b>
<b>Não envolve necessariamente deduplicação.</b>	<b>Desduplicação é uma parte essencial</b>
<b>Precisa de um agente "crawler" e um agente "parser".</b>	<b>Precisa apenas de um agente "crawler".</b>

Fonte: Próprio autor, adaptado de (promptcloud.com, 2023).

## 2.4 LEGALIDADE E ÉTICA AO REALIZAR WEB SCRAPING

Infelizmente, a legalidade e a ética do uso dessas ferramentas de Web Scraping são frequentemente esquecidas. No passado, os cientistas sociais lutavam para encontrar dados para suas pesquisas (MUNZERT et al, 2015).

Hoje, o aumento da digitalização e virtualização de processos sociais resultou em zettabytes (bilhões de gigabytes) de dados disponíveis na World Wide Web. Esses dados fornecem uma representação granular e em tempo real de vários processos, relacionamentos e interações no contexto sociomaterial (KROTOV; TENNYSON 2018).

Assim, esses vastos volumes de dados da Web apresentam a pesquisadores acadêmicos, oportunidades de coleta de dados com o objetivo de responder a novas e velhas questões de pesquisa com rigor, precisão e pontualidade e melhoria do desempenho organizacional (CONSTANTIOU; KALLINIKOS, 2015).

Infelizmente, o aproveitamento desses vastos volumes de dados da Web apresenta sérios aspectos técnicos, jurídicos e éticos. Embora tenha ocorrido uma proliferação de ferramentas e tecnologias que podem ser usadas para Web Scraping (MUNZERT et al. 2015), a legalidade e a ética da coleta de dados da Web ainda são uma “área cinzenta” (SNELL; MENALDO 2016) e embora as estruturas jurídicas existentes possam ser aplicadas, até certo ponto, a prática emergente de Web Scraping, as questões éticas que envolvem o Web scraping foram amplamente ignoradas.

Uma conclusão que emerge dessa literatura é que há um paradoxo inerente envolvendo os dados da Web, o que torna sua recuperação e análise complexas do ponto de vista legal e ético. Por um lado, a Web deveria ser aberta e facilmente acessível ao público. O mesmo princípio de abertura impulsiona muitos modelos de negócios online: os proprietários de sites se beneficiam de uma base de usuários mais ampla acessando os dados disponíveis em seus sites. Por outro lado, esses dados da Web são um ativo crítico aos proprietários de sites e, portanto, precisam ser protegidos. Idealmente, muitos proprietários de sites gostariam que esses dados fossem vistos como propriedade, ou seja, propriedade do indivíduo ou entidade por trás do site que contém estes dados. Essa propriedade é difícil de solidificar, pois o proprietário do site não é necessariamente o proprietário dos dados gerados pelos usuários do site do ponto de vista legal (Dryer e Stockton, 2013).

#### 2.4.1 UM POUCO SOBRE A LEI GERAL DE PROTEÇÃO DE DADOS (LGPD)

O data scraping tem incontáveis aplicações e benefícios e pode ser utilizado para ajudar a identificar e extrair dados para análise, agregar informações de diferentes fontes, encontrar ofertas e descontos em serviços online, mapear redes inexploradas de servidores e sites, entre outros (SELLARS, 2018, p. 373-374). Dada a ampla esfera de aplicabilidade, a técnica foi adotada largamente (SELLARS, 2018, p. 373-374), tendo como objeto, inclusive, o processamento de dados pessoais, o que inclui aqueles disponíveis publicamente, em especial.

Contudo, quando o objeto do processamento tem como foco dados pessoais, surgem importantes questionamentos e preocupações quanto à esfera da privacidade dos indivíduos e proteção desses dados.

Segundo Paula (2020) a lei nº 13.709 foi criada como meio de regulamentar o tratamento de dados pessoais, possui o fundamento de proteger a privacidade, liberdade de expressão, opinião, informação, honra e imagem, inviolabilidade da intimidade, exercício da cidadania pelas pessoas, livre desenvolvimento da personalidade, desenvolvimento econômico e tecnológico e a inovação, livre concorrência, a livre iniciativa, os direitos humanos e a defesa do consumidor.

A lei nº 13.709 foi aprovada em 14 de agosto de 2018, no entanto, entrou em vigor somente em 18 de setembro de 2020, sendo fortemente baseada na General Data Protection Regulation (GDPR), uma regulamentação europeia para os mesmos fins (PEREIRA DA ROCHA, 2019). Deverá ser seguida em qualquer situação em que os dados pessoais sejam tratados e/ou coletados dentro do território nacional.

Art. 1º Esta Lei dispõe sobre o tratamento de dados pessoais, inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado, com o objetivo de proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natural (LGPD, 2020, texto online).

A Lei geral de proteção de dados (LGPD) é uma lei muito abrangente e complementa o Marco Civil da Internet, no entanto alguns aspectos dela estão mais conectados ao universo da Internet das coisas. O titular ao qual as informações se referem deve possuir total controle sobre a coleta, uso e armazenamento de seus dados pessoais (PAULA, 2020).

Assim que concluída a relação entre as partes, todos os dados deverão ser excluídos caso solicitado pelo titular, com exceção em casos previsto por lei algum tipo de guarda obrigatória desses dados. No que se refere a proteção de dados, as partes envolvidas, seja direta ou indiretamente, deverão ter suas intimidades, vida privada e sua honra e imagem preservadas (PAULA, 2020).

Para aqueles que não cumprirem a lei, poderão ser alvo de algumas penalidades sancionadas pela LGPD: advertência contendo um prazo em que os agentes de tratamento de dados deverão aplicar as correções necessárias para adequar-se à lei. A partir disso existem as multas:

“...multa de até 2% (dois por cento) do faturamento da empresa no último ano, limitada, no total, a R\$ 50.000.000,00 (cinquenta milhões de reais) por infração; multa diária até que o problema seja resolvido, publicização da infração após apuração e confirmação de sua ocorrência, bloqueio dos dados pessoais a que se refere a infração até que esta seja regularizada, eliminação dos dados pessoais a que se refere a infração, suspensão parcial do banco de dados a que se refere a infração, suspensão das atividades de tratamento de dados a que se refere a infração por até 6 (seis) meses, prorrogável por igual período e proibição parcial ou total das atividades de tratamento de dados” (PAULA, 2020).

Em caso de algum incidente relacionado a Segurança dos dados, tanto a autoridade nacional, como os titulares deverão ser comunicados imediatamente sobre o acontecido. Deverá ser informado os dados pessoais afetados, quais foram os usuários afetados, quais os riscos que esse o incidente provocou, quais são as medidas de segurança utilizadas para a proteção desses dados e quais serão as medidas que serão adotadas para a mitigação ou, no melhor dos casos, a resolução do incidente (PECK, 2016).

Um regulamento que afeta todas as empresas que processam dados de cidadãos brasileiros, mesmo que sejam dos Estados Unidos, como Google ou Facebook. As multas enfrentadas por quem não a cumpre são um dos pontos mais polêmicos e midiáticos. Este é o primeiro regulamento nesta matéria que afeta todo o país e, portanto, unifica direitos e obrigações). De fato, durante anos foi uma demanda de muitas empresas e setores, como o de tecnologia, que tiveram que lidar com legislações diferentes sobre o uso e processamento de dados pessoais para poder oferecer seus serviços em território nacional (PEREIRA DA ROCHA, 2019).

Este novo regulamento determina que todas as empresas, independentemente do seu país de origem ou atividade, devem cumpri-lo caso recolham, armazenem, processem, utilizem ou gerem qualquer tipo de dados sobre cidadãos brasileiros. Essa lei veio para dar mais direitos e mecanismos aos usuários em relação aos seus dados (PAULA, 2020).

O direito à portabilidade é permitido se os seus dados forem tratados de forma automatizada para sem recuperados num formato e os transferir para outro responsável. Esses dados devem estar em um formato estruturado, comumente usados e legível mecanicamente para que possa ser facilmente transmitido para outro responsável e, assim, facilitar uma mudança de provedor, por exemplo (PAULA, 2020).

Devido ao exposto, deve-se tomar muito cuidado ao executar tais estratégias de web scraping, especialmente envolvendo possíveis dados sensíveis de usuários, que estão protegidos pela LGPD.

## 2.5 CROWDSOURCING

Crowdsourcing (em português, contribuição colaborativa ou colaboração coletiva) é uma aglutinação da língua inglesa. É composta de crowd (multidão) e outsourcing (terceirização). O termo foi descrito pela primeira vez em 2006 por Jeff Howe na revista Wired (HOWE, WIRED MAGAZINE 2006) e é definido pelo Dicionário Merriam-Webster (1996) como o processo de obtenção de serviços, ideias ou conteúdo mediante a solicitação de contribuições de um grande grupo de pessoas e, especialmente, de uma comunidade online, em vez de usar fornecedores tradicionais ou uma equipe de empregados.

Abordando o fenômeno *crowdsourcing* de um ponto de vista prático, com suas diversas formas de realização, bem como de um ponto de vista teórico com diferentes pontos de vista de definição podemos deduzir a seguinte definição geral de crowdsourcing: *Crowdsourcing* é definido como o ato de tarefas de terceirização originalmente executadas dentro de uma organização, ou atribuídas externamente na forma de um relacionamento comercial, para uma massa indefinidamente grande e heterogênea de atores potenciais. Isso acontece por meio de uma chamada aberta via Internet para fins gratuitos, e de valor ou uso criativo. O incentivo para participar pode ser monetário e/ou não monetário na natureza. (BONABEAU 2009; BRABHAM 2009)

Este trabalho utilizará dois sites de *crowdsourcing* de ofertas para fazer a varredura dos mesmos e tentar encontrar os resultados de acordo com a busca do usuário e as palavras-chave cadastradas na plataforma *Promo Scraper*, sendo eles: <https://gatry.com> e <https://boletando.com/>.

O website Gantry é uma plataforma de *social e-commerce* para compartilhamento de promoções e interação entre os usuários. (GATRY, 2023)

A plataforma Boletando é também um website para encontrar descontos, ofertas e cupons, alimentado pela sua própria comunidade, porém, com um foco em tecnologia. (BOLETANDO, 2023)

### **3 SOLUÇÕES EXISTENTES**

Este capítulo tem como objetivo identificar e analisar ferramentas com soluções relacionadas, porém não idênticas à deste trabalho. Além disso, avaliar e comparar os

lados positivos e negativos destas ferramentas similares, e por quais motivos ainda se mostra cabível a solução proposta por este trabalho.

### 3.1 COMPARADOR DE PREÇOS ZOOM

A primeira a ser analisada será a plataforma Zoom (pode ser acessada em <http://zoom.com.br>) que segundo seu site oficial, se define como:

Hoje, mais do que um comparador de preços, somos um serviço completo de apoio à compra. Anunciamos ofertas de centenas de lojas parceiras em nosso site e app para auxiliar na busca pelo melhor preço. Assim, oferecemos conforto e segurança para você encontrar o produto ideal pelo melhor preço sem precisar ir de site em site. (ZOOM, 2022)

A plataforma se encontra em diversas plataformas, tais como: o próprio *browser*, extensão para *browser*, aplicativo para android e aplicativo para IOS. As características e funcionalidades da plataformas são:

- Comparação de preços;
- Busca de promoções;
- Buscas e promoções de sites parceiros;
- Divulgação de produtos de sites parceiros;
- Filtragem de produtos;
- Avaliações dos produtos;
- Histórico de preços;
- Alerta de preço;
- Cashback;
- Venda de produtos dentro da plataforma.

Apenas observando e utilizando o sistema, não foi possível ter evidências concretas sobre o método utilizado pela plataforma para reunir os dados dos produtos, bem como seu histórico de preços e suas avaliações. Por este motivo, não cabe uma comparação com o presente trabalho, que por sua vez, deixa explícito que a metodologia utilizada para obtenção dos dados se dá via *Web Scraping*.

### 3.2 COMPARADOR DE PREÇOS JÁCOTEI

Outra opção amplamente conhecida é o comparador de preços Já Cotei. Este também está presente em todas as plataformas como seu concorrente, Zoom. No website oficial do JáCotei é explicado com detalhes em suas condições de uso como ocorre o funcionamento da ferramenta:

O JáCotei fornece através do site informações coletadas nas lojas virtuais da Internet, capturadas através de softwares de desenvolvimento próprio (robôs), atendendo aos requisitos legais para coleta de preços. O trabalho é realizado unicamente pelo JáCotei sem que haja interferência ou ajuda de quaisquer empresas coletadas (lojas). (JÁCOTEI, 2023)

O serviço tem diversas categorias e subcategorias para organizar o catálogo de ofertas, além de trazer um programa de milhagem e parceria com as plataformas Smiles e Multiplus (TECHTUDO, 2023).

Esta plataforma, bem como sua concorrente, também pode ser utilizada via browser, extensão para browser, aplicativo para android e aplicativo para IOS. As características e funcionalidades da plataforma JáCotei são:

- Comparação de preços;
- Busca de promoções;
- Buscas e promoções de sites parceiros;
- Filtragem de produtos;
- Verificação de fraudes Black Friday;
- Guia de produtos e comparativo no blog;
- Divulgação de produtos de sites parceiros;
- Histórico de preços;
- Alerta de preço.

Assim como na avaliação da plataforma evidenciada na seção 3.1, não se obteve provas substanciais acerca do procedimento empregado pela plataforma para coletar os dados dos produtos. Por conseguinte, não se torna pertinente realizar uma comparação com o presente estudo. A atual pesquisa se utilizará apenas das duas plataformas citadas na seção 3 deste trabalho.

### 3.3 ANÁLISE DAS SOLUÇÕES EXISTENTES

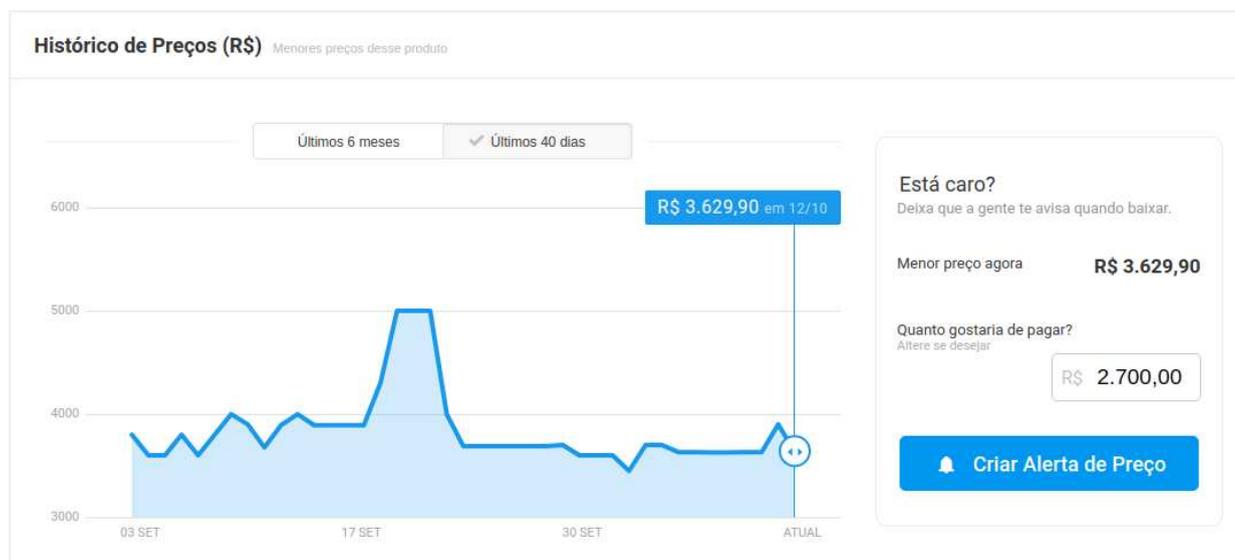
A análise foi realizada através da utilização de cada uma das plataformas, realizando ações de cadastro, utilização de cupons, compras teste com cashback, navegação no site de modo geral, uso dos aplicativos e uso das extensões. O uso dos mesmos foi sempre em sua fonte oficial (seja esta o próprio site, extensão, ou aplicativo oficial). Após a análise, originaram-se os tópicos das principais vantagens e desvantagens dessas plataformas, as comparando com a solução proposta por este trabalho.

#### 3.3.1 VANTAGENS

A principal vantagem destes modelos é a facilidade e robustez da solução. O usuário que acessa a plataforma, tem a possibilidade de pesquisar qualquer produto e receber em poucos segundos um comparativo completo dos principais e mais confiáveis e-commerces que estão vendendo este item, e comparar o preço que cada uma dessas lojas está vendendo.

Há também a opção de ver o histórico de preços desse determinado produto, que é mostrado em forma de gráfico. Para usuários cadastrados, há também a opção “Criar Alerta” que permite que o usuário seja notificado quando um item de seu interesse alcance um preço que o próprio usuário estipulou.

**Figura 06 - Gráfico do histórico de preços de um produto na plataforma Zoom**



Fonte: Site oficial Zoom, 2022

Esta opção é a que mais se assemelha a solução proposta por este trabalho, onde o usuário não tem a intenção da compra imediata, e estaria disposto a esperar até que um produto de seu interesse chegue a um valor mais acessível. Neste caso, o valor que o próprio usuário teria estipulado.

A plataforma Zoom também conta agora com a opção Cashback Zoom. Alguns produtos de lojas parceiras agora contam com o cashback, isto é, receber em dinheiro uma porcentagem do valor da compra e o resgate pode ser feito pelo aplicativo Zoom para celular (ZOOM, 2022). A plataforma Zoom também contempla parcerias com alguns e-commerces e disponibiliza cupons para as compras provenientes do Zoom para o site parceiro.

Figura 07 - Funcionalidades da plataforma Zoom (extensão)



Fonte: Site oficial Zoom, 2022.

Segundo o livro "*Holiday Shopping Strategy Guide for Black Friday*" do jornal Tampa Bay Times, *Black Friday* é um termo coloquial para a sexta-feira após o Dia de Ação de Graças nos Estados Unidos. Tradicionalmente marca o início da temporada de compras de Natal nos Estados Unidos. Muitas lojas oferecem promoções altamente divulgadas com desconto e geralmente abrem cedo, às vezes até meia-noite ou até mesmo no dia de ação de graças.

A plataforma JáCotei possui também uma ferramenta que mostra o verdadeiro desconto de produtos na *black friday*. Segundo o site oficial da JáCotei, além de buscar pelos melhores preços, um gráfico é feito comparando os verdadeiros e falsos descontos no período de *black friday*.

**Figura 08: Exemplo de gráficos de descontos reais e falsos JáCotei**



Fonte: Site oficial JáCotei, 2022

### 3.3.2 DESVANTAGENS

Uma das grandes desvantagens deste modelo, é a presença de parcerias entre diversas lojas com a plataforma Zoom, como mencionado no seu site oficial “Anunciamos ofertas de centenas de lojas parceiras em nosso site e app para auxiliar na busca pelo melhor preço.” (ZOOM, 2022). Atualmente o site da Zoom possui mais de 2,5 milhões de produtos monitorados e 300 lojas parceiras (ZOOM, 2022).

Desta maneira, não se tem a garantia de que todas as lojas, sejam elas parceiras ou não parceiras da Zoom, serão levadas em consideração na hora de procurar e comparar os melhores preços. Outro aspecto importante de se destacar, é o fato de se tratar de uma organização privada com objetivos financeiros bem definidos.

Já no modelo de negócios da empresa JáCotei, segundo seu website oficial, está explícito os planos que as empresas podem contratar a fim de ter a visibilidade que desejam para determinados produtos (vide figura 09).

### Figura 09: Modelo de negócio JáCotei

## ESCOLHA A MELHOR FORMA DE INVESTIR

Investimento inicial é o menor do mercado. Oferecemos o melhor pacote de investimento adequado a sua loja. Pague antes ou depois.

#### Modelo de negócio - Pós-pago

Modelo de investimento em que o lojista disponibiliza sem limites de verba seus produtos no site, e no final do mês realiza o pagamento de acordo com a movimentação dos cliques gerados através do JáCotei.

- Valores dos CPCs ( custo por clique) são de acordo com a tabela vigente com a categoria do produto (disponível para consulta em nossa Central do Anunciante).
- Investimento mensal mínimo de R\$ 5.000,00.
- Pagamento faturado 15 DFQ.

#### Modelo de negócio - Pré-pago

Modelo de investimento em que o lojista determina a verba a ser utilizada de acordo com as opções de crédito (R\$) do JáCotei, podendo realizar suas recargas para que sua campanha não fique sem saldo e fora do ar.

- Investimento Mínimo de R\$ 300,00
- Investimento de R\$ 500,00 com bonificação de R\$ 100,00
- Investimentos acima de R\$ 1.000,00 com bonificação de R\$ 250,00

Fonte: Site oficial Jacotei, 2022

Isso implica em um objetivo intrínseco e comum a todas as empresas, que é a geração de lucro. Diferente de projetos sem fins lucrativos e feitos pela comunidade de maneira colaborativa, não há garantia alguma de que a plataforma em questão tenha o foco apenas no benefício do usuário final. Desta forma, há uma grande fatia de público que considera e confia mais em um modelo não ligado a grandes corporações e que seja feito de e para a comunidade. E é visando essa fatia demográfica, que o trabalho em questão se torna útil.

## 4 SOLUÇÃO PROPOSTA

Após realizar a análise das ferramentas anteriores, foi concluído que há um nicho de consumidores que se encaixam na proposta deste trabalho, principalmente pelo fato de optarem por uma alternativa sem fins lucrativos e que visa apenas a ajuda entre os membros da comunidade.

### 4.1 REQUISITOS FUNCIONAIS

- **RF01 - Cadastro:** O sistema deve permitir o cadastro de usuário. Na tela de cadastro será necessário inserir as seguintes informações: Nome de usuário, primeiro nome, sobrenome, e-mail, senha e confirmação de senha (vide figura 18)
- **RF02 - Login:** O sistema deve permitir que o usuário faça login utilizando o mesmo e-mail e senha previamente cadastrados.
- **RF03 - Inserção de palavras-chave:** O sistema deve permitir que o usuário insira até 8 (oito) palavras-chave, representando os produtos que são do seu interesse.
- **RF04 - Intervalo de tempo da varredura:** O sistema deve permitir que o usuário escolha um intervalo de tempo para que o algoritmo faça a varredura, buscando por combinações nas palavras-chave. Opções disponíveis: 6(seis), 12(doze) ou 24(vinte e quatro) horas.
- **RF05 - Receber e-mail:** O sistema deve permitir que o usuário receba e-mails quando uma oferta for encontrada, ou (caso tenha marcado esta opção na página inicial) de também receber um e-mail quando uma oferta não for encontrada.
- **RF06 - Ver informações relevantes na página inicial:** O sistema deve permitir que o usuário veja na tela inicial informações como: Nome, sobrenome, e-mail

cadastrado, data e título do último e-mail enviado, *checkbox* com opção para receber e-mails de oferta não encontrada.

- **RF07 - Botão explicativo:** O sistema deverá mostrar ao usuário, um botão com uma breve explicação sobre o funcionamento da ferramenta.
- **RF08 - Logout:** O sistema deve permitir que o usuário realize o *logout* a qualquer momento, clicando no botão de *logout*.
- **RF09 - Histórico:** O sistema deverá mostrar um histórico de todas as promoções já encontradas para o usuário.
- **RF10 - Remover promoções:** O sistema deverá permitir que promoções sejam removidas manualmente (apenas administrador).
- **RF11 - Adicionar promoções:** O sistema deverá permitir que promoções sejam adicionadas manualmente (apenas administrador).
- **RF12 - Adicionar usuário:** O sistema deverá permitir que um usuário seja adicionado (apenas administrador).
- **RF13 - Remover usuário:** O sistema deverá permitir que um usuário seja removido (apenas administrador).

#### 4.2 REQUISITOS NÃO FUNCIONAIS

- **RNF01 - Carregamento da página:** O tempo de carregamento da página deve ser de no máximo 5 segundos.
- **RNF02 - Segurança de senha:** A senha dos usuários deve ser armazenada criptografada.

- **RNF03 - Implementação MVT:** O sistema deve ser desenvolvido com arquitetura em camadas, utilizando o padrão MVT (Model-View-Template).
- **RNF04 - Compatibilidade entre navegadores:** O sistema deve ser compatível com os navegadores Edge 80.0 ou superior, Safari 5.0 ou superior e Chrome 20.0 ou superior.
- **RNF05 - Responsividade:** O sistema deve ter responsividade para desktop, tablets e celulares.
- **RNF06 - Framework web:** O sistema deverá usar o *web framework* Django versão 3.
- **RNF07 - Banco de dados:** O sistema deverá persistir as informações das entidades usuário e promoção em um banco de dados relacional SQLite localmente e PostgreSQL em produção.
- **RNF08 - Hospedagem:** O sistema deverá usar Heroku para hospedagem e deploy da aplicação.
- **RNF09 - Certificado SSL e domínio:** O sistema deverá utilizar Heroku para escolha de domínio seguro com certificado SSL.
- **RNF10 - Websites a serem varridos:** O sistema deverá fazer a varredura dos websites "gatry.com" e "boletando.com".

#### 4.3 DIAGRAMA DE CASOS DE USO

Na Figura 10, podemos visualizar o diagrama de casos de uso para os atores identificados.

Figura 10 - Diagrama de casos de uso



Fonte: Produção do próprio autor (2023)

#### 4.4 ANÁLISE

Por tratar-se de um projeto que tem futuro *open-source*, foi decidido que ele terá o código escrito na língua inglesa. Visto que está sendo utilizado um framework *full stack* para web, muitas das classes, objetos e métodos já são provenientes do próprio framework, porém existem entidades relevantes ao sistema em questão, que são *User* e *Promotion*.

No preenchimento do formulário deverão ser colocadas as palavras-chave (*keywords*), que representam os interesses do usuário para encontrar promoções, sejam

estes os nomes de marcas ou de produtos e o intervalo no qual o algoritmo irá fazer a varredura novamente, podendo ser seis, doze ou 24 horas.

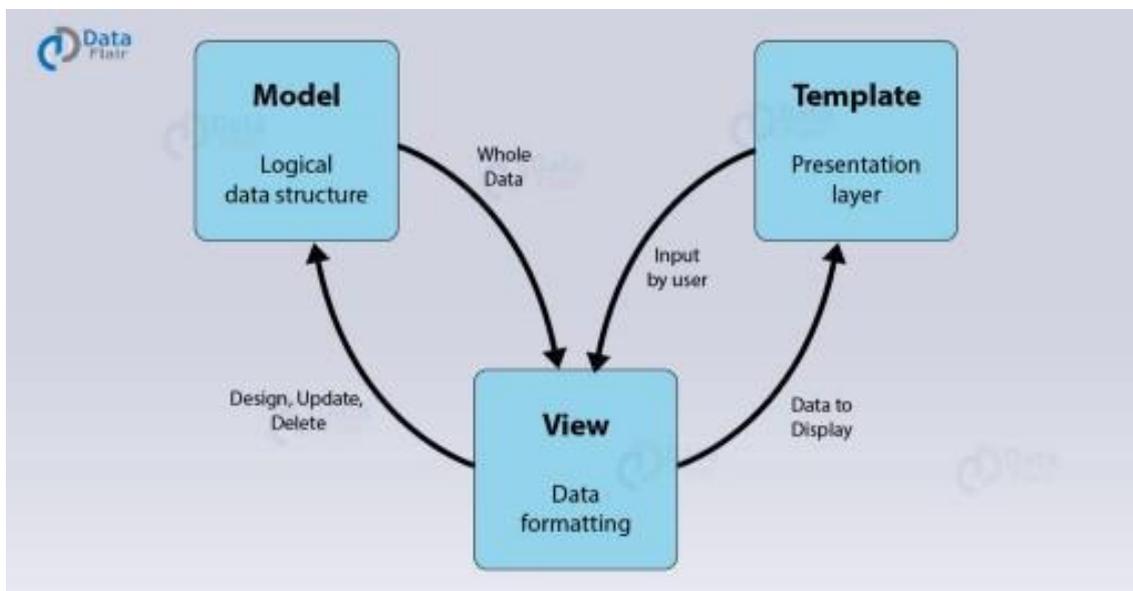
Para encontrar as devidas promoções, e baseada nas palavras-chave informadas pelo usuário, será utilizada a técnica de *Web Scraping* para fazer a varredura de dois websites onde a própria comunidade alimenta com promoções e disponibiliza o link.

Para cada website a ser realizado a varredura, uma implementação diferente deverá ser feita devido a diferença na árvore HTML deles. Para o disparo de e-mails, será utilizada a tecnologia de microsserviços, disparados por uma plataforma em nuvem, e o intervalo será determinado pelo usuário na tela inicial.

#### 4.5 ARQUITETURA DO SISTEMA

A aplicação em questão, proveniente do framework Django, utiliza a arquitetura *model-template-views* (MTV).

**Figura 11 - Modelo MTV (Data Flair, 2022)**



Fonte: Data flair, 2022

Esta arquitetura é uma variação do padrão MVC (*Model, View e Controller*), pode ser observado no próprio acrônimo, a palavra-chave *Template* substitui o *Controller*.

Porém, o *Template* não funciona exatamente como o controlador e tem algumas propriedades diferentes do *Controller* (DATA FLAIR, 2022).

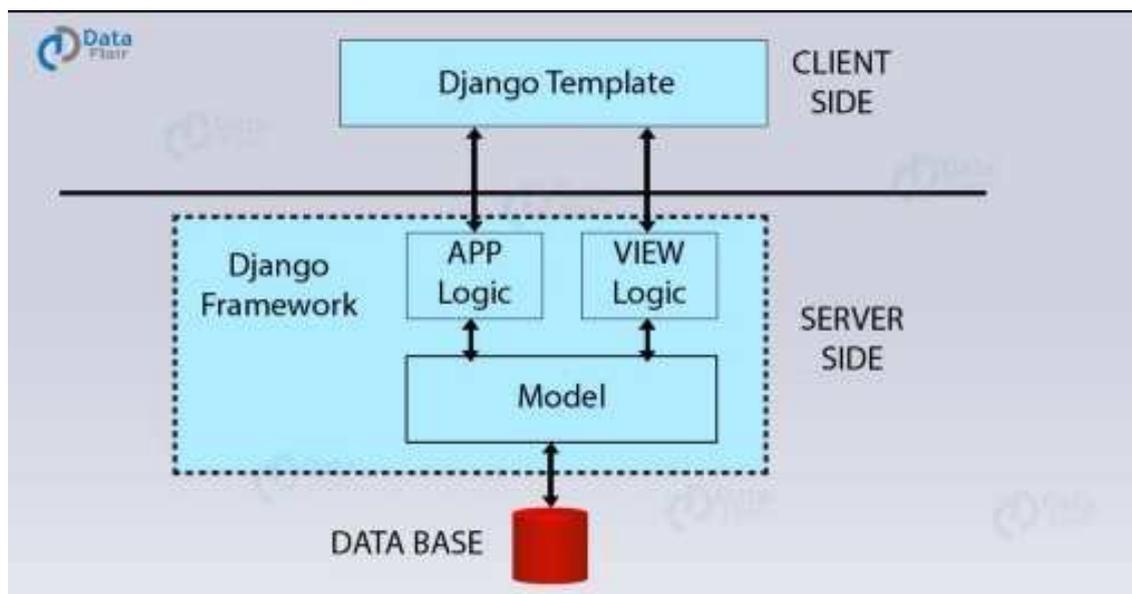
As definições de *Model* ainda permanecem as mesmas, isto é, o *Model* contém a estrutura lógica do arquivo do projeto e é o middleware e o manipulador de dados entre o banco de dados e a visualização. O modelo fornece uma definição de como os dados são formatados como provenientes da visualização, de modo que ele armazena no banco de dados e vice-versa, ou seja, a informação de recuperação do banco de dados é transferida para a visualização no formato exibível (DATA FLAIR, 2022).

De acordo com Data Flair (2022), "A arquitetura *View* em MTV pode parecer com o controlador, mas não é. O *View* nesta arquitetura MTV formata os dados por meio do modelo. Por sua vez, ele se comunica com o banco de dados e os dados que são transferidos para o modelo para visualização."

O principal objetivo do *Template* é manter tudo o que o navegador renderiza. Os dados do modelo que vêm do servidor em diferentes partes ao mesmo tempo que se integram quando o usuário interage com o site. Esta camada de modelo no Django é mais semelhante à camada de *Views* no padrão MVC e fornece muito mais extensibilidade para os desenvolvedores de *front-end* do que a arquitetura MVC estava dando (DATA FLAIR, 2022).

No *front-end*, será utilizada a própria ferramenta do django para templates, chamada Django Template Language. Django Template Language ou DTL é uma linguagem de modelo baseada em texto que fornece uma ponte entre scripts como HTML, CSS, JS, etc. e linguagens de programação como Python. O DTL foi criado especificamente para desenvolvedores incorporarem códigos lógicos Django em arquivos de modelo HTML. DTL também tem uma grande vantagem sobre outras linguagens de modelo baseadas em texto por causa de sua simplicidade, facilidade de aprender a sintaxe e extensibilidade (ASK PYTHON, 2023).

Figura 12 - Comunicação do Modelo MTV entre Cliente e Servidor (Data Flair, 2022)



Fonte: Data flair (2022)

#### 4.5.1 ENTIDADES DO BANCO DE DADOS

São abordados nesta seção, as principais entidades do banco de dados, seus devidos campos e tipos, e as relações entre as tabelas (vide Figura 13). Para facilitar o entendimento, serão traduzidos os campos das entidades para língua portuguesa, porém no código e no banco de dados, estes campos estão escritos na língua inglesa.

- **Usuário (User):** Representa o usuário da aplicação. Pode ou não ser administrador, e tem os campos preenchidos após fazer o cadastro na aplicação, ou se for criado por um usuário administrador manualmente. Esta entidade realiza a maioria das ações descritas no diagrama de casos de uso. É referenciado com chave estrangeira pelo modelo Combinação (Match). Cada objeto usuário poderá ser referenciado por um ou mais objetos Match. Veja a tabela 02 a seguir com os devidos campos, seus tipos e tamanhos:

**Tabela 02 - Campos do objeto User no banco de dados**

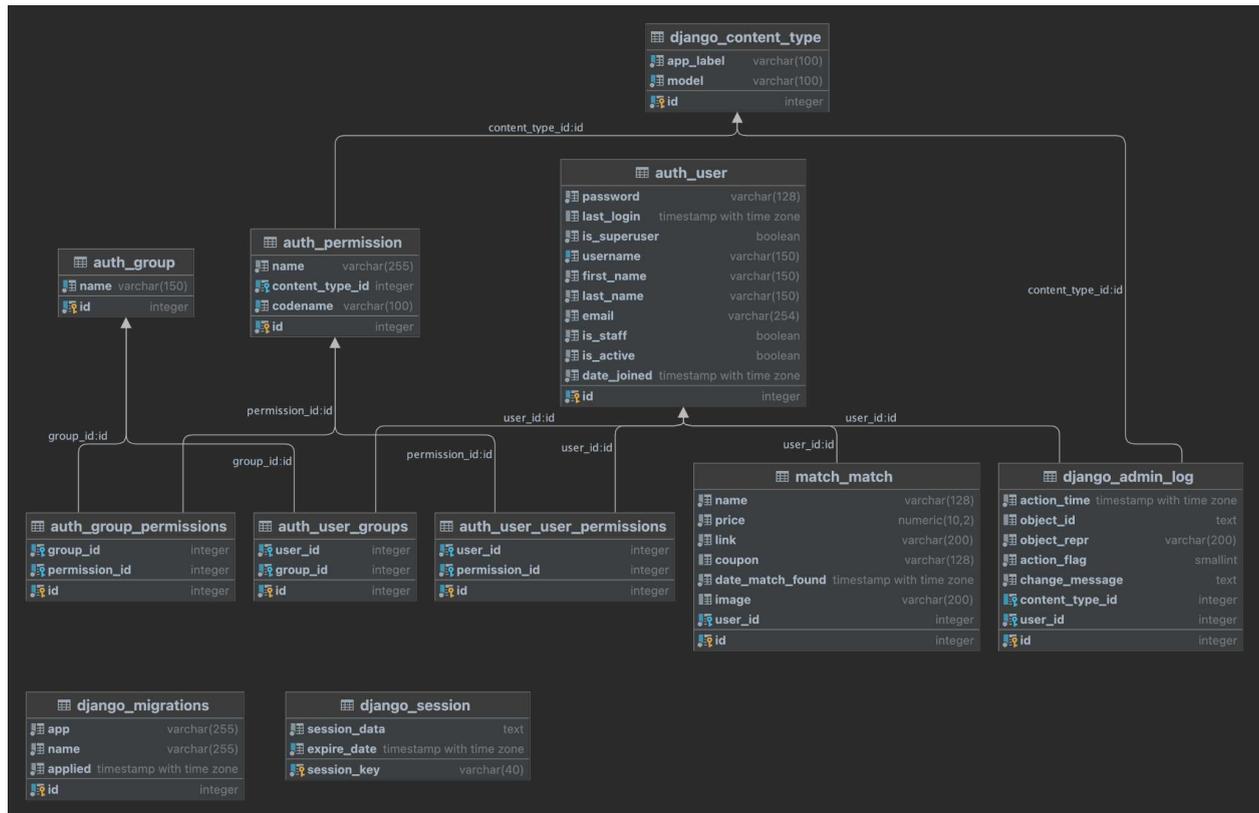
Nome do campo (nome no código)	Tipo do campo
identificador (id)	primary key integer
senha (password)	varchar (128)
último login (last_login)	timestamp com timezone
data de criação (date_joined)	timestamp com timezone
é super usuário (is_superuser)	boolean
nome de usuário (username)	varchar (150)
primeiro nome (first name)	varchar (150)
último nome (last name)	varchar (150)
palavras chave (keywords)	varchar (254)
email	varchar (254)
é staff (is_staff)	boolean
é ativo (is_active)	boolean

- **Combinação (Match):** Representação do objeto que é criado quando o algoritmo de varredura encontra uma (ou mais) promoção baseado nas palavras-chave fornecidas pelo usuário. Utiliza o campo id do usuário (user\_id) como chave estrangeira para relacionamento com o usuário. Cada objeto Match pertencerá a um usuário. Veja a tabela 03 a seguir com os devidos campos, seus tipos e tamanhos:

**Tabela 03 - Campos do objeto Match no banco de dados**

<b>Nome do campo (nome no código)</b>	<b>Tipo do campo</b>
identificador (id)	primary key integer
id do usuário (user_id)	foreign key integer
nome (name)	varchar (128)
preço (price)	numérico (10,2)
link	varchar (200)
cupom (coupon)	varchar (128)
data combinação encontrada (date_match_found)	timestamp com timezone

Figura 13: Diagrama entidade relação do banco de dados



Fonte: Produção do próprio autor

## 4.6 TECNOLOGIAS UTILIZADAS

### 4.6.1 BANCO DE DADOS

#### 4.6.1.1 BANCO DE DADOS LOCAL

Para uso local, o banco de dados escolhido para esta aplicação foi o banco relacional SQLite, devido a ser uma alternativa leve, de código aberto, *serverless* e facilita o trabalho em vários bancos de dados na mesma sessão simultaneamente, tornando-o flexível. (SIMPLELEARN, 2023).

Este banco de dados é gerado localmente após executar o comando para aplicar as migrações. Um arquivo com extensão *.sqlite* é criado, e este, representa o banco de dados local.

SQLite é uma biblioteca de linguagem C que implementa um mecanismo de banco de dados SQL pequeno, rápido, independente, de alta confiabilidade e completo.

O SQLite é o mecanismo de banco de dados mais usado no mundo. O SQLite está embutido em todos os telefones celulares e na maioria dos computadores e vem embutido em inúmeros outros aplicativos que as pessoas usam todos os dias. O formato de arquivo SQLite é estável, multiplataforma e compatível com versões anteriores. (SQLite, 2023).

#### 4.6.1.2 BANCO DE DADOS EM PRODUÇÃO

Já na versão em produção, que está disponível em "<http://promo-scrapers.herokuapp.com>" a própria plataforma se encarrega de ler as informações nas configurações do projeto, no arquivo *settings.py* e criar uma instância de um banco de dados PostgreSQL, dando acesso total ao administrador do projeto.

O PostgreSQL é um poderoso sistema de banco de dados objeto-relacional de código aberto que usa e estende a linguagem SQL combinada com muitos recursos que armazenam e dimensionam com segurança as cargas de trabalho de dados mais complicadas. As origens do PostgreSQL remontam a 1986 como parte do projeto POSTGRES da Universidade da Califórnia em Berkeley e tem mais de 35 anos de desenvolvimento ativo na plataforma principal.

O PostgreSQL conquistou uma forte reputação por sua arquitetura comprovada, confiabilidade, integridade de dados, conjunto robusto de recursos, extensibilidade e dedicação da comunidade de código aberto por trás do software para oferecer consistentemente soluções inovadoras e de alto desempenho. O PostgreSQL é executado em todos os principais sistemas operacionais, é compatível com ACID desde 2001 e possui complementos poderosos, como o popular extensor de banco de dados geoespacial PostGIS. Hoje, PostgreSQL tem se tornado o banco de dados relacional de software livre preferido por muitas pessoas e organizações. (POSTGRES, 2023).

#### 4.6.2 FRAMEWORK DJANGO

Django é uma ferramenta que pode ser usada para o desenvolvimento full-stack de aplicações e páginas web, bem como para o desenvolvimento de servidores. É considerado o melhor framework para o desenvolvimento de aplicações web com Python

e é um dos frameworks de desenvolvimento mais requisitados pelos programadores que trabalham com essa linguagem no desenvolvimento web (DJANGO DOCS, 2020).

Criado por Adrian Holovaty e Simon Willison enquanto eles estavam trabalhando em PHP e precisavam de algo que facilitasse a atualização de um site para que uma página cumprisse os prazos. Naquela época, eles mudaram para Python e, em 2005, lançaram o Django para tornar a programação ainda mais fácil (GOUR, 2019).

O Django fornece quase tudo o que os programadores precisam e podem querer usar. É uma ferramenta que segue princípios de design consistentes e possui uma boa base de documentação para facilitar o trabalho dos desenvolvedores. Também funciona com componentes substituíveis e intercambiáveis. Isso significa que ele pode ser dimensionado com bastante facilidade. Nesse sentido, um exemplo de uso do Django temos no Instagram ou Disqus, plataformas que o utilizaram para melhorar seus servidores graças à escalabilidade do ambiente de desenvolvimento (GOUR, 2019).

Devido a sua versatilidade este ambiente de desenvolvimento tem sido usado para a criação de todos os tipos de páginas web. De sistemas que são puramente para gerenciamento de conteúdo, como, por exemplo, um wiki; para redes sociais ou sites de notícias (DJANGO DOCS, 2020).

O Django é compatível com os frameworks mais populares usados no lado do cliente, para que possa enviar conteúdo em qualquer formato, facilita a detecção e a correção de possíveis falhas de segurança em sites do lado do servidor. Ele fornece gerenciamento seguro de usuários e senhas e evita erros comuns no design e desenvolvimento de back-end (GOUR, 2019).

Outra característica é que ele é portátil, já pode ser usado em qualquer sistema e plataforma e é apoiado por muitos provedores de hospedagem que também costumam fornecer a documentação necessária para implementá-lo. Django facilita a manutenção e rapidez no desenvolvimento web (GOUR, 2019).

Ele incentiva a criação de código reutilizável, o que simplifica muito o trabalho dos desenvolvedores web. Segundo o site oficial do framework em questão, Django é um *framework web* de alto nível escrito em Python que incentiva o desenvolvimento rápido e um design limpo e pragmático.

Construído por desenvolvedores experientes, ele cuida de grande parte do trabalho de desenvolvimento da Web, para que você possa se concentrar em escrever

seu aplicativo sem precisar reinventar a roda (GOUR, 2019). É gratuito e de código aberto. O autor do trabalho em questão, possui sólida experiência profissional com este framework, o que tornou a experiência de desenvolvimento mais ágil e assertiva.

#### 4.6.3 BOOTSTRAP

Para auxiliar no desenvolvimento de um site responsivo, e com estilos CSS já definidos e padronizados, foi utilizada a tecnologia Bootstrap.

Bootstrap é um biblioteca (também chamado de kit de ferramentas de front-end) com código-fonte aberto para desenvolvimento de componentes de interface e front-end para sites e aplicações web, usando HTML, CSS e JavaScript, baseado em modelos de design para a tipografia, melhorando a experiência do usuário em um site amigável e responsivo (BOOTSTRAP, 2023).

#### 4.6.4 BIBLIOTECA BEAUTIFULSOUP

Os usuários da Internet, têm acesso a muitas informações sobre clientes, ofertas, cotações de ações, fenômenos físicos. Os usuários podem ler dados, mas também poder explorá-los, transformando-os em um formato operacional para finalmente analisá-los e aproveitá-los (BEAUTIFUL SOUP, 2020)

Web scraping é a técnica que permite que essas informações sejam extraídas em um formato que pode ser usado por programas de computador. A biblioteca utilizada para fazer a varredura dos sites desejados será BeautifulSoup, que de acordo com sua própria documentação oficial, é uma biblioteca Python para extrair dados de arquivos HTML e XML.

Ele funciona com seu analisador favorito para fornecer maneiras idiomáticas de navegar, pesquisar e modificar a árvore de análise. Geralmente, isso economiza horas ou dias de trabalho dos programadores. A biblioteca BeautifulSoup permite extrair conteúdo e transformá-lo em uma lista, matriz ou dicionário Python (BEAUTIFUL SOUP, 2020)

Esta biblioteca é muito popular porque possui documentação completa e suas funcionalidades são bem estruturadas. Além disso, há uma grande comunidade que

oferece soluções variadas para usar esta biblioteca. Os sites são escritos em linguagens de computador HTML e CSS, que são usadas para projetar páginas da web.

HTML é usado para marcar e organizar o conteúdo. A estilização e a aparência da página da web (cores, tamanho do texto) é feito usando a linguagem CSS. No campo do desenvolvimento web, "tag soup" é um termo depreciativo para a escrita sintática ou estruturalmente incorreta do HTML de uma página da web (BEAUTIFUL SOUP, 2020)

#### 4.6.5 EZGMAIL

Para fazer o envio de e-mails quando uma ou mais promoções forem encontradas, a biblioteca (na linguagem python) de envio de e-mails utilizada foi a ezgmail.

Segundo a documentação oficial do ezgmail, a biblioteca serve como uma interface "pythonica" para a API do Gmail que simplesmente funciona. O nome, em inglês, tem a sonoridade de "easy gmail" que pode ser traduzido como "gmail fácil".

Eles afirmam que a biblioteca do próprio gmail é mais robusta, porém requer muito mais ajustes e configurações para que a integração seja feita (PYPI EZGMAIL, 2023).

Estes fatores de facilidade e simplicidade para instalar e utilizar as funcionalidades de envio de e-mails foram decisivos para que o trabalho em questão utilizasse essa biblioteca.

Para que isto fosse possível, foi necessário criar um projeto no "Google Cloud project" e habilitar o serviço do Google API, para que assim, aceite a biblioteca ezgmail, como sugere a documentação do ezgmail e também do próprio Google.

#### 4.6.6 HEROKU

Segundo o site oficial da plataforma, Heroku é uma plataforma em nuvem que permite às empresas criar, entregar, monitorar e dimensionar aplicativos - "Somos a maneira mais rápida de ir da ideia à URL, evitando todas as dores de cabeça da infraestrutura" (HEROKU, 2022).

A Heroku é uma oferta de plataforma como serviço amplamente confiável que permite que os desenvolvedores implementem, dimensionem e gerenciem aplicativos sem problemas. Esta plataforma oferece suporte para uma ampla gama de linguagens de programação como Java, Ruby, PHP, Node.js, Python, Scala e Clojure.

A Heroku executa aplicativos por meio de contêineres virtuais conhecidos como Dynos. A Heroku cobra de seus usuários com base em recursos de computação, como RAM e máquinas virtuais alocadas. A plataforma Heroku e os aplicativos criados pelo

usuário usam o Amazon Web Services como a infraestrutura subjacente (HEROKU, 2022).

Os desenvolvedores podem alcançar o desenvolvimento rápido de aplicativos usando-o, pois é bastante conveniente. Um serviço de PaaS como o Heroku fornece um serviço totalmente gerenciado, que cuida da infraestrutura e pode escalar automaticamente os servidores para cima ou para baixo.

Em comparação com um provedor IaaS puro, o usuário acrescenta camadas adicionais de valor sobre os recursos da máquina virtual. Para conseguir um domínio e hospedagem segura, bem como um processo inteligente e rápido de *deploy*, foi escolhida a plataforma Heroku (HEROKU, 2022).

#### 4.6.6.1 HEROKU SCHEDULER

Para que o algoritmo de varredura fosse executado recorrentemente (a cada 6, 12 ou 24 horas) um serviço de "*scheduler*" (agendador) se faz necessário.

Para entender o funcionamento do *Scheduler*, é necessário primeiro entender o que são *dynos* dentro da plataforma Heroku:

"A plataforma Heroku usa o modelo de contêiner para executar e dimensionar todos os aplicativos Heroku. Os contêineres usados no Heroku são chamados de "dynos". *Dynos* são contêineres Linux virtualizados e isolados, projetados para executar código com base em um comando especificado pelo usuário. Seu aplicativo pode ser dimensionado para qualquer número especificado de dinâmômetros com base em suas demandas de recursos. Os recursos de gerenciamento de contêiner do Heroku fornecem uma maneira fácil de dimensionar e gerenciar o número, tamanho e tipo de dinâmômetros que seu aplicativo pode precisar a qualquer momento." (HEROKU, 2023).

O *Scheduler* é um complemento gratuito para executar trabalhos em seu aplicativo em intervalos de tempo agendados, muito parecido com o *cron* em um ambiente de servidor tradicional. Embora o *Scheduler* seja um complemento gratuito, ele executa trabalhos agendados por meio de dinâmômetros únicos que contam para o seu uso mensal. A execução do trabalho do agendador é esperada, mas não garantida. Depois de acessar sua interface, um painel permitirá que você configure tarefas para serem executadas a cada 10 minutos, a cada hora ou a cada dia, em um horário

especificado. Quando invocados, esses trabalhos serão executados como *dynos* únicos e aparecerão em seus logs como um *dyno* denominado "scheduler.X" (HEROKU, 2023).

Por conta do projeto deste trabalho já estar hospedado na plataforma Heroku, ficou ainda mais prático utilizar a sua própria ferramenta de scheduler.

#### 4.6.6.2 DEPLOY AUTOMÁTICO HEROKU E GITHUB

O projeto em questão utiliza a praticidade e agilidade dos deploys automáticos no heroku integrados ao github, e assim deixa a aplicação disponível para todo o público, o tempo inteiro, e podendo ter melhorias e correções assim que se empurra um novo *commit* para *branch* master.

Quando *deploys* automáticos são ativados para uma ramificação do GitHub, o Heroku cria e implanta todos os *pushes* nessa ramificação. Se, por exemplo, existir um aplicativo de desenvolvimento no Heroku, este poderá configurar pushes para sua ramificação de desenvolvimento do GitHub para serem criados e implantados automaticamente nesse aplicativo (HEROKU, 2022).

## 5 DESENVOLVIMENTO

### 5.1 IMPLEMENTAÇÃO

Após a fase inicial de planejamento do sistema, foi dada a implementação do sistema de fato, focando primeiramente em gerar e codificar uma estrutura que comportasse a arquitetura prevista. Em seguida, foi iniciado o desenvolvimento das funcionalidades previstas nos requisitos funcionais desta aplicação.

Seguindo os conceitos do Scrum e os adaptando para equipes de um só membro, este capítulo mostra todo o desenvolvimento do projeto seguindo sprints. Cada uma das próximas seções mostra o que foi feito para chegar no resultado esperado pela Sprint. Cada Sprint planejada pelo autor teve duração de aproximadamente 25 horas, sendo este tempo utilizado entre pesquisas em tutoriais, adequação de layout e codificação. O tempo total de horas estimadas para as Sprints foi de 225 horas, tendo uma soma ao decorrer do desenvolvimento de aproximadamente 9 semanas, com trabalhos de 25 horas semanais.

#### 5.1.1 DESCRIÇÃO GERAL DO SISTEMA

O aplicativo web foi desenvolvido usando o framework Django, que segue o padrão MVT (Model View Template), como já explicado na seção 4.5. Sendo assim, na parte "*Model*" estão as classes que são a abstração dos objetos no banco de dados, os principais *models* desta aplicação são User e Match, que estão detalhados na seção 4.5.1. Para o *front end*, ainda fazendo parte do framework django, foi utilizado o DLT (*Django Language Template*) que integra facilmente com o backend (visto que é uma ferramenta do próprio framework), renderizando as views e podendo mostrar facilmente informações dos objetos do banco de dados na tela, que são trazidos para o DLT como contexto (*context*). Ainda no front end, Bootstrap é importado e utilizado para estilizar os elementos HTML e a própria página como um todo (vide seção 4.6.3).

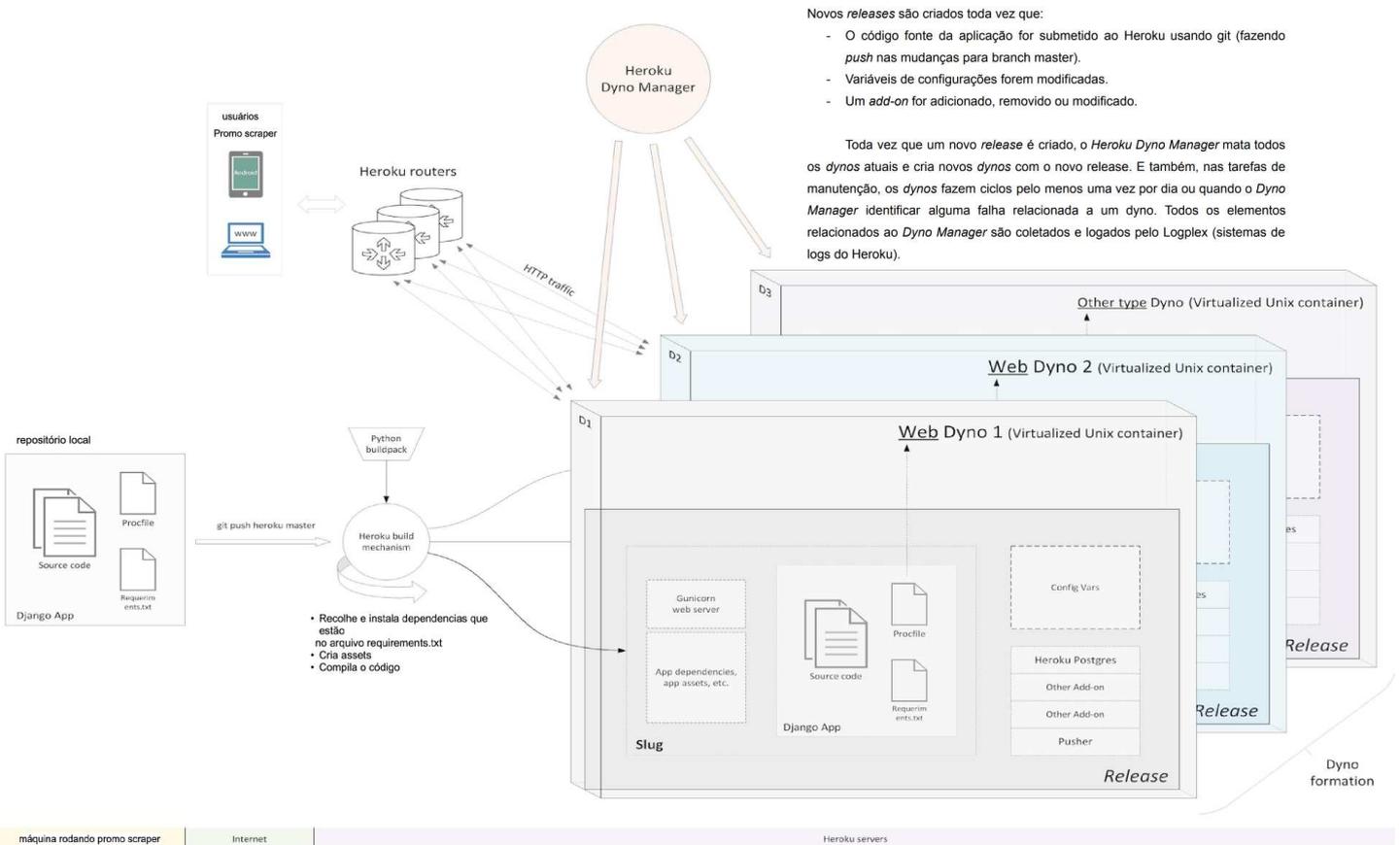
O trabalho em questão, está no ar (*live*) e hospedado na plataforma Heroku, e para que isto fosse possível, alguns ajustes tiveram que ser feitos no projeto, como criar alguns arquivos de configuração, adicionar add-ons (sendo o banco de dados

PostgreSQL e heroku scheduler, exemplos), especificar a buildpack de python e integrar o deploy automático com *GitHub*. Estes passos estão devidamente detalhados no apêndice A, nas sprints 1 e 7.

O serviço de web scraping e de disparo de e-mail é um tipo de serviço recorrente e utiliza-se do heroku scheduler para fazer a varredura dos websites de promoção em um certo período, e caso uma promoção tenha sido encontrada, também disparar o e-mail ao usuário. A implementação dos algoritmos de web scraping podem ser encontradas nas sprints 4 e 5.

Para facilitar o entendimento da comunicação entre web app django e Heroku, o comportamento da plataforma Heroku, seus respectivos *dynos*, a parte de requisições HTTP, cliente servidor, e também de novas releases, a figura 16 abaixo demonstra essas comunicações e interações.

**Figura 14: Visão geral do sistema e funcionamento dos *Heroku dynos* e deploys**



Fonte: Próprio autor

## 5.1.2 SPRINTS

Esta seção faz uma breve descrição de cada sprint, onde cada uma será devidamente detalhada no apêndice A.

### 5.1.2.1 SPRINT 1 - ESTRUTURAÇÃO DO PROJETO

Etapa inicial de desenvolvimento e estruturação do projeto, onde foi criado o projeto django, com seus devidos diretórios e arquivos, bem como rodar o servidor local pela primeira vez.

### 5.1.2.2 SPRINT 2 - USUÁRIOS E AUTENTICAÇÃO

Foram feitas as funcionalidades de registro e autenticação de usuários na plataforma.

#### 5.1.2.3 SPRINT 3 - FRONT END PÁGINA INICIAL

Foi desenvolvido o *front end* da página principal da aplicação.

#### 5.1.2.4 SPRINT 4 - ALGORITMO DE VARREDURA GATRY

Nesta etapa, foram criados a lógica e o algoritmo de varredura do website <https://gatry.com>.

#### 5.1.2.5 SPRINT 5 - ALGORITMO DE VARREDURA BOLETANDO

Foram desenvolvidos nesta sprint, a lógica e o algoritmo de varredura do website <https://boletando.com/>.

#### 5.1.2.6 SPRINT 6 - POPULANDO O BANCO E MOSTRANDO NO FRONT END

Esta sprint foi designada para alinhar as funcionalidades de varredura com o banco de dados, populando o banco com objetos encontrados após a varredura dos websites e mostrando estas informações no front end.

#### 5.1.2.8 SPRINT 8 - SERVIÇO SCHEDULER

Desenvolvimento dos scripts para serviços (*jobs*) recorrentes, para o presente trabalho em questão, o serviço de varredura e envio de e-mails.

## 6 RESULTADOS OBTIDOS

A execução deste projeto permitiu documentar o todo processo da criação de uma aplicação web, destacando a relevância das fases preliminares à implementação do software, tais como: investigação de mercado e contexto histórico, avaliação de soluções já existentes, elaboração do modelo de negócio, levantamento de requisitos, adoção de uma abordagem ágil adaptada, estruturação da arquitetura do sistema e publicação do sistema em produção para o público em geral.

Após o desenvolvimento deste trabalho, foi realizada uma análise do cumprimento dos requisitos, e os seguintes requisitos funcionais foram alcançados:

- RF01 - Realização de cadastro
- RF02 - Realização de Login
- RF03 - Inserção de palavras-chave
- RF04 - Inserção do intervalo de tempo da varredura
- RF05 - Recebimento de e-mail
- RF06 - Visão de informações relevantes na página inicial (parcial)
- RF07 - Botão explicativo
- RF08 - Realização de Logout
- RF10 - Remoção de promoções (administrador)
- RF11 - Adição de promoções (administrador)
- RF12 - Adição de usuários (administrador)
- RF13 - Remoção de usuários (administrador)

Os seguintes requisitos funcionais não foram cumpridos:

- RF06 - Visão de informações relevantes na página inicial (parcial): Este requisito está parcialmente incompleto, pois apesar de mostrar nome, sobrenome e e-mail, ainda não mostra o título e data dos últimos emails enviados.
- RF09 - Histórico: A página de histórico para ter organizadamente as últimas promoções encontradas, ordenadas por data, ainda não foi desenvolvida.

Todos os requisitos não funcionais foram alcançados, exceto:

- RNF09 - Certificado SSL e domínio: Este requisito não foi alcançado, devido a limitações no plano atual da ferramenta Heroku, para adicionar a certificação SSL e assim, obter o selo de site seguro, seria necessário comprar um plano mais robusto, o que se tornou financeiramente inviável para este trabalho.

## 7 CONCLUSÃO

Devido ao exposto, observa-se que o projeto em questão se mostra uma alternativa viável para os consumidores no âmbito da internet. Ainda que não possa substituir outras plataformas similares mais robustas com soluções sofisticadas, no aspecto de alerta sobre possíveis promoções, olhando apenas websites alimentados pela comunidade, é uma alternativa viável.

Embora nem todos os requisitos funcionais tenham sido completamente entregues (vide capítulo 6), os objetivos deste projeto, que foram citados previamente na seção 1.3, foram considerados um sucesso.

A análise do comportamento e tendências de pesquisas e compras online não teve um teor desafiador, não obstante, foi de grande importância para que o embasamento sobre o projeto tivesse uma boa fundamentação e para entender como o mercado e-commerce vem se comportando.

O estudo e análise das ferramentas comparativas de preços já existentes também foram importantes para a confecção deste projeto, visto que, de uma perspectiva de complexidade técnica, vasta variedade de funcionalidades, e um modelo de negócios sólido, foi necessário o uso e entendimento destas ferramentas, bem como compará-las com o que este projeto se dispôs a desenvolver.

Sem dúvidas a parte mais desafiadora foi o desenvolvimento da solução do aplicativo web. O estudo e leitura de documentações, todo o processo de desenvolvimento do software como um todo, da modelagem até o *deploy*, foi bastante desafiador e teve uma exigência técnica grande, para que todos os elementos de software funcionassem corretamente. O aprendizado sobre as tecnologias de *web scraping*, o próprio framework *django*, o design do *front end*, a implementação de notificações recorrentes, e além de tudo, colocar o projeto no ar, fazendo o *deploy* no Heroku, foi enorme, porém extremamente gratificante ao mesmo tempo.

O projeto continua em fase de implementação, porém já cumpre grande parte dos requisitos funcionais e não funcionais. Além disso, o projeto tem pretensão de permanecer *open-source* e está disposto a receber novas modificações e sugestões da comunidade.

Para trabalhos futuros, foram identificadas algumas funcionalidades que tornariam o sistema mais completo, e com mais funcionalidades para o usuário, sendo estas:

- Adição de mais websites a serem varridos: Com mais websites a serem varridos, maior será a possibilidade de encontrar uma promoção que combine com as palavras chave cadastradas pelo usuário.
- Refatorar o sistema de procura de palavras compostas, com expressões regulares: Existem melhorias a serem feitas para encontrar palavras chave compostas ou produtos de duas ou mais palavras cadastradas pelo usuário, por exemplo: Placa de vídeo, guarda roupa, micro-ondas.
- Implementação da página de histórico com gráficos: A implementação da página de histórico de promoções encontradas, juntamente com gráficos para ajudar na visualização do preço ao longo do tempo.
- Implementação da funcionalidade de pesquisa instantânea: A implementação de uma busca ativa, apesar de não ser o foco deste trabalho, também seria de grande ajuda para o usuário, uma vez que ele poderia querer fazer uma pesquisa ativa, ainda que tivesse um limite diário de pesquisas ativas (para não sobrecarregar o sistema).

## REFERÊNCIAS BIBLIOGRÁFICAS

ABCOMM. **E-commerce de produtos durante a pandemia da COVID-19**. Disponível em: <<https://abcomm.org/Pesquisas/ecommerce-no-covid-konduto-abcomm.pdf>>. Acesso em: setembro 2020.

ALVES, Rafael Ferreira; VANALLE, Rosângela Maria. **Ciclo de Vida de Desenvolvimento de Sistemas-visão conceitual dos modelos clássico, espiral e prototipação**. Associação Brasileira de Engenharia de Produção, 2001.

BARBOSA, F. **Zoom compra Buscapé e mais três marcas da Naspers**. Disponível em: <<https://www.tudocelular.com/mercado/noticias/n141574/zoom-compra-buscapede-mais-tres-marcas-naspers.html>>. Acesso em: setembro 2020.

BARREIRA, Elisa da Conceição Marques. **População e Enriquecimento de Ontologias através de Web Scraping**. Dissertação (Mestrado em Engenharia Informática) – Instituto Superior de Engenharia do Porto, Porto. 2014.

BANHARA, Renato Francisco. **Impacto e mudanças do varejo brasileiro com a COVID-19: a transformação do e-commerce**. 2021. Tese de Doutorado.

BEAUTIFUL SOUP. **Beautiful Soup Documentation**. Disponível em: <<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>>. Acesso em: setembro 2020.

BIGDATACORP. **Estudo perfil do E-commerce brasileiro**, 7ª ed. Disponível em: <https://bigdatacorp.com.br/estudo-perfil-do-e-commerce-brasileiro-7a-edicao/>. Acesso em: fevereiro 2023.

DATA FLAIR. **Understanding Components & Working**. Disponível em: <<https://data-flair.training/blogs/django-mtv-architecture/>>. Acesso em: setembro 2022.

DJANGO DOCS. **Django FAQ: General**. Disponível em: <<https://docs.djangoproject.com/en/3.0/faq/general/>>. Acesso em: setembro 2020.

DJANGO. **Django website**. Disponível em: <<https://www.djangoproject.com/>>. Acesso em: setembro 2020.

DOS SANTOS GRACIANO, HELTON LUIZ. **ScraperCI: um protótipo de Web scraper para coleta de dados**. 2022. 79 p. Dissertação de mestrado — Universidade Federal de São Carlos, São Carlos, 2022.

E-COMMERCE BRASIL. **Com pandemia, e-commerce cresce 81% em abril e fatura R\$ 9,4 bilhões**. 2021. Disponível em: <[https://www.ecommercebrasil.com.br/noticias/e-commerce-cresce-abril-fatura-compreco-nfie-coronavirus/?utm\\_source=artigo&utm\\_medium=ecbr&utm\\_campaign=crescimentoe-commercecorona](https://www.ecommercebrasil.com.br/noticias/e-commerce-cresce-abril-fatura-compreco-nfie-coronavirus/?utm_source=artigo&utm_medium=ecbr&utm_campaign=crescimentoe-commercecorona)>. Acesso em: setembro 2022

GITHUB.COM **GitHub: What is GitHub?** Disponível em: <<https://guides.github.com/activities/hello-world/>>. Acesso em: setembro 2022.

E-COMMERCE BRASIL. **Mais usuários de internet no Brasil:** mais oportunidades para o e-commerce. Disponível em: <<https://www.ecommercebrasil.com.br/artigos/mais-usuarios-internet-mais-oportunidades/#:~:text=Segundo%20os%20dados%20levantados%20na,usu%C3%A1rios%20de%20internet%20no%20Brasil>>. Acesso em: setembro 2020.

E-COMMERCE BRASIL. **Com pandemia, e-commerce cresce 81% em abril e fatura R\$ 9,4 bilhões.** 2020. Disponível em: <https://www.ecommercebrasil.com.br/noticias/e-commerce-cresce-abril-fatura-compreconfie-coronavirus>. Acesso em: junho 2023.

EXPECTATIVA de vendas no e-commerce. Disponível em: <https://dados.abcomm.org/previsao-de-vendas-online>. Acesso em: fevereiro 2023.

GOUR, R. **Working Structure of Django MTV Architecture**, 2019. Disponível em: <<https://towardsdatascience.com/working-structure-of-django-mtv-architecture-a741c8c64082>>. Acesso em: setembro 2020.

G1. **Zoom compra concorrente Buscapé.** Disponível em: <<https://g1.globo.com/economia/noticia/2019/05/14/zoom-compra-concorrente-buscapede.html>>. Acesso em: setembro 2020.

HEROKU. **What is Heroku?** Disponível em: <<https://www.heroku.com/what>>. Acesso em: setembro 2022.

HEROKU. **About Heroku.** Disponível em: <<https://www.heroku.com/about>>. Acesso em: setembro 2022.

HEROKU. **GitHub Integration (Heroku GitHub Deploys).** Disponível em: <<https://devcenter.heroku.com/articles/github-integration>>. Acesso em: janeiro 2023

HEROKU. **Heroku Scheduler.** Disponível em: <<https://devcenter.heroku.com/articles/scheduler>>. Acesso em: janeiro 2023

HEROKU. **Heroku Dynos.** Disponível em: <<https://www.heroku.com/dynos>>. Acesso em: janeiro 2023

HOLDING, A. **Global Internet Industry Research.** 2016. Avazu Holding. Acesso em: fevereiro 2023.

IBGE. **Internet já é acessível em 90,0% dos domicílios do país em 2021 | Agência de Notícias.** 2022. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/>

34954-internet-ja-e-acessivel-em-90-0-dos-domicilios-do-pais-em-2021. Acesso em: fevereiro 2023.

JORNAL DO COMÉRCIO. **Quarentena aumenta vendas por e-commerce**. Disponível em:

<[https://www.jornaldocomercio.com/\\_conteudo/especiais/coronavirus/2020/03/731946-quarentena-aumenta-vendas-por-e-commerce.html](https://www.jornaldocomercio.com/_conteudo/especiais/coronavirus/2020/03/731946-quarentena-aumenta-vendas-por-e-commerce.html)>. Acesso em: setembro 2020.

JUS. **Web sscrapping na lei geral de proteção de dados pessoais**. Disponível em: <https://jus.com.br/artigos/87950/web-sscrapping-na-lei-geral-protacao-de-dados-pessoais>. Acesso em: fevereiro 2023.

CONSTANTIOU, I. D.; KALLINIKOS, J. **New Games, New Rules: Big Data and the Changing Context of Strategy**. Journal of Information Technology, v. 30, n. 1, p. 44–57, 2015

KROTOV, Vlad; SILVA, Leiser. **Legality and ethics of web scraping**. 2018.

MERRIAM-WEBSTER; WEBSTER. **Merriam Webster's Vocabulary Builder**. [S. l.]: I B S Books Stocked, 1996. ISBN 9780877794585.

MILAGRE, José. **Raspagem Web ficam ilegais com a GDPR**. Disponível em: <https://josemilagre.jusbrasil.com.br/artigos/769118869/os-web-crawlers-scrapping-e-servi-cos-de-raspagem-web-ficam-ilegais-com-a-gdpr-e-lei-geral-de-protacao-de-dados>. Acesso em: fevereiro 2023.

MUNZERT, Simon et al. **Automated data collection with R: A practical guide to web scraping and text mining**. John Wiley & Sons, 2014.

Landers, R. N., Brusso, R. C., Cavanaugh, K. J., and Collmus, A. B. 2016. “**A Primer on Theory-Driven Web Scraping: Automatic Extraction of Big Data from the Internet for use in Psychological Research**,” Psychological Methods (21:4), pp. 475-492.

NIQ EBIT. **Webshoppers**. 2023. Disponível em: <https://nielseniq.com/global/pt/landing-page/ebit/nielseniq-ebit-brasil/webshoppers/>. Acesso em: abril 2023.

RELATÓRIO WEBSHOPPERS 2022. **Webshoppers 2023 47ª Edição**. E-bit, 2023. Disponível em: <<http://www.ebit.com.br/webshoppers>>. Acesso em: fevereiro de 2023

PAULA, Marcus Vinícius Cândido de. **Segurança da informação e a internet das coisas**. 2020. Monografia (Graduação em Engenharia da Computação) - Faculdade de Tecnologia e Ciências Sociais Aplicadas, Centro Universitário de Brasília, Brasília, 2020.

PARSEHUB | **Free web scraping - The most powerful web scraper**. 2023. Disponível em: <https://parsehub.com/>. Acesso em: junho 2023.

PECK, Patrícia. **Direito Digital**. 6. ed. rev., atual. e ampl. – São Paulo: Saraiva, 2016.

PEREIRA DA ROCHA, Camila et al. **Segurança da Informação: A ISO 27.001 como Ferramenta de Controle para LGPD**. Revista de Tecnologia da Informação e Comunicação da Faculdade Estácio do Pará, [S.l.], v. 2, n. 3, p. 78-97, ago. 2019. ISSN 2595-8798. Disponível em: <<http://www.revistasfap.com/ojs3/index.php/tic/article/view/285>>. Acesso em: fevereiro 2023.

PINHEIRO, Guilherme. **Venda de smartphones deve ultrapassar 6,5 bilhões de unidades em 2022**. 17 abr. 2022. Disponível em: <https://mundoconectado.com.br/noticias/v/24698/venda-de-smartphones-deve-ultrapassar-65-bilhoes-de-unidades-em-2022#:~:text=O%20interesse%20das%20pessoas%20por,celulares%20em%20todo%20o%20mundo>. Acesso em: março 2023.

ROCHA, FELIPE. **Modelos de Desenvolvimento de Software: resumo**. 2012. Disponível em: <https://felipelirarocho.wordpress.com/2012/04/15/diversos-modelos-de-desenvolvimento-de-software-resumo/>. Acesso em: março 2023.

SÃO PAULO. **Decreto nº 65.347, de 9 de dezembro de 2020. Decreto nº 65.347, de 09/12/2020**. Disponível em: <https://www.lexml.gov.br/urn/urn:lex:br;sao.paulo:estadual:decreto:2020-12-09;65347>. Acesso em: junho 2023.

SIQUEIRA, Diego; ANDRADE LIMA, Assis; PAULO, João; MORENO, David. **Segurança em Redes de Sensores sem Fio: Desafios, Tendências e Orientações**. v.13, ed. 1, p 402-411, 2015 [S.l.]: Revista Gestão. Org.

SCHWABER, K.; SUTHERLAND, J. **The scrum guide**. 2017. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>>. Acesso em: setembro 2022.

SCRUM ALLIANCE. **Overview: What is Scrum?** Disponível em: <<https://www.scrumalliance.org/about-scrum/overview>>. Acesso em: setembro 2020.

**SCRUM ORG**. Disponível em: <[scrum.org](http://scrum.org)>. Acesso em: fevereiro 2022.

TAKASHI, V., MORAES, A. D. **Segurança Em IoT: Entendendo os riscos e ameaças em IoT**. Brasil: Alta Books. LGPD. Camptecnica, 2021. Disponível em: <[https://www.camptecnica.com.br/lgpd/?gclid=Cj0KCQjw8eOLBhC1ARIsAOzx5cFN4qp19AjYF5WveBlewtAN2Q\\_nOifGJHwpkDneZ1a9UxWSIgmIcaAhOzEALw\\_wcB](https://www.camptecnica.com.br/lgpd/?gclid=Cj0KCQjw8eOLBhC1ARIsAOzx5cFN4qp19AjYF5WveBlewtAN2Q_nOifGJHwpkDneZ1a9UxWSIgmIcaAhOzEALw_wcB)>. Acesso em: fevereiro 2023.

TECHTERMS. **Techterms: Web Application**. Disponível em: <[https://techterms.com/definition/web\\_application](https://techterms.com/definition/web_application)>. Acesso em: setembro 2020.

TECHTUDO. **Lista tem dez sites para comparar e encontrar os menores preços online.** Disponível em: <<https://www.techtudo.com.br/noticias/2014/07/lista-tem-dez-sites-para-comparar-e-encontrar-os-menores-precos-online.ghtml>>. Acesso em: maio 2023.

Think with Google. **TENDÊNCIAS globais de e-commerce e compras online.** Disponível em: [https://www.thinkwithgoogle.com/feature/online-shopping-trends/the-findings?lang=pt\\_BR](https://www.thinkwithgoogle.com/feature/online-shopping-trends/the-findings?lang=pt_BR). Acesso em: fevereiro 2023.

Offerwise. **Líder em Obtenção de Insights na América Latina.** Disponível em: <<https://www.offerwise.com/>>. Acesso em: fevereiro 2023.

TUDOCELULAR.COM. **Fim da concorrência? Zoom compra Buscapé e mais três marcas da Naspers.** 14 maio 2019. Disponível em: <https://www.tudocelular.com/mercado/noticias/n141574/zoom-compra-buscapede-mais-tres-marcas-naspers.html>. Acesso em: fevereiro 2023.

TRELLO.COM. **What is Trello?** Disponível em: <<https://help.trello.com/article/708-what-is-trello>>. Acesso em: setembro 2022.

VERHEYEN, G. Scrum: **Framework, not methodology.** Disponível em: <<https://guntherverheyen.com/2013/03/21/scrum-framework-not-methodology/>>. Acesso em: setembro 2020.

ZOOM. **O que é o Zoom?** Disponível em: <<https://tiresuaduvida.zoom.com.br/hc/pt-br/articles/360000942023-O-que-%C3%A9-o-Zoom->>. Acesso em: setembro 2022.

ZOOM. **Como usar o Cashback Zoom?** Disponível em: <<https://www.zoom.com.br/landing-page/tire-suas-duvidas-sobre-o-cashback-zoom>>. Acesso em: setembro 2022.

ZOOM. **Cupons de desconto, ofertas e cashback** Disponível em: <<https://www.zoom.com.br/cupom-de-desconto>>. Acesso em: setembro 2022.

JACOTEI. **Inclua sua loja no JáCotei** Disponível em: <<https://www.jacotei.com.br/sualoja/>>. Acesso em: setembro 2022.

JACOTEI. **Compare preços e ainda ganhe prêmios em milhas.** Disponível em: <<https://www.techtudo.com.br/tudo-sobre/jacotei-app/>>. Acesso em: fevereiro 2023.

JACOTEI. **Jacotei Light friday.** Disponível em: <<https://www.jacotei.com.br/lightfriday/>>. Acesso em: fevereiro 2023.

IDC. **IDC: Mercado de celular fecha segundo trimestre de 2022 com alta de 3,1%, de acordo com estudo da IDC Brasil.** Disponível em: <<https://www.idc.com/getdoc.jsp?containerId=prLA49834722>>. Acesso em: fevereiro de 2023.

NIGAM, Harshit; BISWAS, Prantik. **From web scraping to web crawling.** In: **Applications of Artificial Intelligence and Machine Learning: Select Proceedings of ICAAAIML 2020.** Springer Singapore, 2021. p. 97-112.

SELLARS, Andrew. **Twenty years of web scraping and the computer fraud and abuse act.** Journal of Science & Technology Law (JoSTL). Boston University, v. 24, p. 372-415, 2018. Disponível em: [https://scholarship.law.bu.edu/faculty\\_scholarship/465](https://scholarship.law.bu.edu/faculty_scholarship/465). Acesso em: janeiro 2023.

M. Jazayeri, **"Some Trends in Web Application Development,"** in **Future of Software Engineering**, 2007. FOSE '07, 2007, pp. 199-213.

AMAZON AWS. **Amazon: What Is A Web Application?** Disponível em: <<https://aws.amazon.com/what-is/web-application/>>. Acesso em: fevereiro 2023.

BONABEAU, Eric. **Decisions 2.0: The power of collective intelligence.** MIT Sloan management review, v. 50, n. 2, p. 45, 2009.

BRABHAM, Daren C. **Crowdsourcing the public participation process for planning projects.** Planning Theory, v. 8, n. 3, p. 242-262, 2009.

HOWE, Jeff. **Crowdsourcing: A definition.** Crowdsourcing: Tracking the rise of the amateur. Accessed March, v. 2, p. 2016, 2006.

HOWE, Jeff et al. **The rise of crowdsourcing.** Wired magazine, v. 14, n. 6, p. 1-4, 2006.

Albright, Mark. **Holiday Shopping Strategy Guide for Black Friday.** Tampa Bay Times. 2012.

SQLITE. **What Is SQLite?** Disponível em: <<https://www.sqlite.org/index.html/>>. Acesso em: fevereiro 2023.

POSTGRESQL. **What Is PostgreSQL?** Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: fevereiro 2023.

ASK PYTHON. **What is Django Template Language?** Disponível em: <<https://www.askpython.com/django/django-template-language>>. Acesso em: fevereiro 2023.

SIMPLELEARN. **What is SQLite? Everything You Need to Know** Disponível em: <<https://www.simplilearn.com/tutorials/sql-tutorial/what-is-sqlite>>. Acesso em: fevereiro 2023.

PYPI EZGMAIL. **EZGmail** Disponível em: <<https://pypi.org/project/EZGmail/>>. Acesso em: fevereiro 2023.

BOOTSTRAP. **BOOTSTRAP**. Disponível em: <<https://getbootstrap.com.br/>> Acesso em: fevereiro 2023.

GATRY. **GATRY**. Disponível em: <<https://gatry.com/>> Acesso em: fevereiro 2023.

BOLETANDO. **BOLETANDO**. Disponível em: <<https://www.boletando.com/>> Acesso em: fevereiro 2023.

PROMPTCLOUD. **Data Scraping vs Data Crawling**. Disponível em: <<https://www.promptcloud.com/blog/2012/05/data-scraping-vs-data-crawling.html>> Acesso em: fevereiro 2023.

YOURDON, Edward. **Análise Estruturada Moderna**. 3ª Ed. Trad. Dalton C. de Alencar. Rio de Janeiro: Campus, 1990.

DREYER, A.; STOCKTON, Jamie. **Internet “data scraping”: A primer for counseling clients**. New York Law Journal, v. 7, p. 1-3, 2013.

## APÊNDICE A - SPRINTS

### SPRINT 1 - ESTRUTURAÇÃO DO PROJETO

Para a estruturação do projeto, foi levado em conta as pesquisas, pensamento crítico para abstrair as regras de negócio para tecnologias e frameworks tangíveis, e estudo das documentações das tecnologias a serem utilizadas.

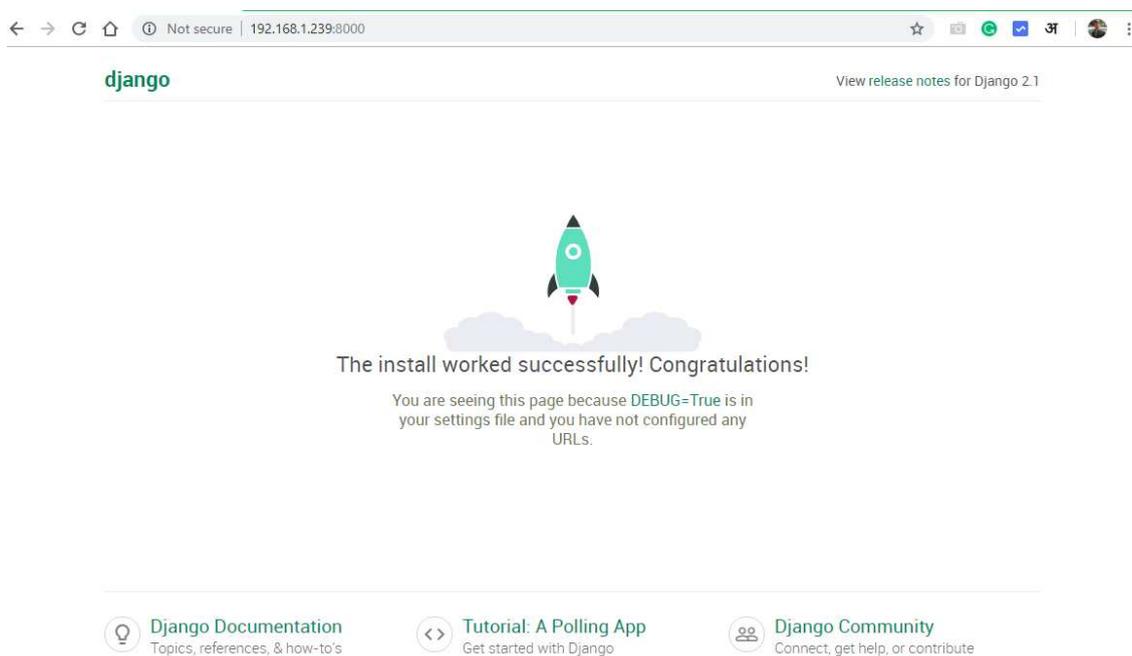
A tecnologia principal neste aspecto, foi o framework Django, que, como citado na seção 4.6.2, é um framework para aplicações web muito robusto, repleto de funcionalidades, e full stack, isto é, com um só framework é possível rodar uma aplicação com back end, e front end. Alguns frameworks para front end (React, VueJs) poderiam ter sido utilizados em conjunto com django, mas o próprio framework Django possui a DLT (Django Language Template) que em conjunto com o Bootstrap para estilizar o HTML, cumpriu perfeitamente o papel para lidar com o front end. Além disso, a DLT se comunica com objetos renderizados no back end com muito mais facilidade, visto que é a maneira padrão do Django de lidar com o front end.

Para criar a estrutura inicial do projeto, eis o que foi feito:

- **Planejamento e Configuração do Ambiente:** Antes de iniciar o desenvolvimento, foi feito um planejamento cuidadoso para definir os objetivos da aplicação, identificar os requisitos e estabelecer uma estratégia de implementação. Em seguida, o ambiente de desenvolvimento foi configurado com a instalação do Python e a criação de um ambiente virtual isolado.
- **Criação do Projeto:** Com o ambiente configurado, o projeto Django foi criado usando o comando apropriado. Isso gerou a estrutura inicial do projeto, incluindo o arquivo de configuração principal e diretórios padrão.
- **Definição dos Modelos:** Os modelos, que representam as entidades e a estrutura de dados da aplicação, foram definidos usando classes Python. Cada modelo corresponde a uma tabela no banco de dados, e os campos e relacionamentos foram especificados de acordo com as necessidades da aplicação (vide figura 13).

- Criação do "app": Um aplicativo Django é uma parte modular do projeto que possui funcionalidades específicas. Para criar um novo aplicativo, estando do o diretório raiz do projeto, foi executado o comando para criação do "app".
- Gerando e aplicando migrações: Após isso, foi utilizado o comando para gerar migrações, criando arquivos de migração com base nas alterações nos modelos e em seguida foi executado o comando para aplicá-las, para de fato criar as tabelas no banco de dados.
- Criando super usuário: Também foi criado um super usuário, utilizando o devido comando, para que fosse possível acessar o painel de administrador do Django, chamado de Django Admin. Este usuário terá todos os acessos e privilégios, inclusive de editar e remover qualquer objeto do banco de dados, inclusive outros usuários.

**Figura 15: Projeto django criado com sucesso e rodando localmente na porta 8000**



A primeira Sprint foi concluída com sucesso, e analisando a figura 15, podemos verificar que além da estrutura do projeto estar pronta, já é possível rodar um servidor local na porta 8000, onde o próprio framework mostra uma página de instalação de sucesso.

## SPRINT 2 - USUÁRIOS E AUTENTICAÇÃO

Para esta Sprint, o objetivo foi fazer um template com os formulários para o login e cadastro de usuário, e redirecioná-los a uma página inicial, mesmo que ainda em branco e não estando populada.

**Figura 16: Tela de Login/Cadastro**

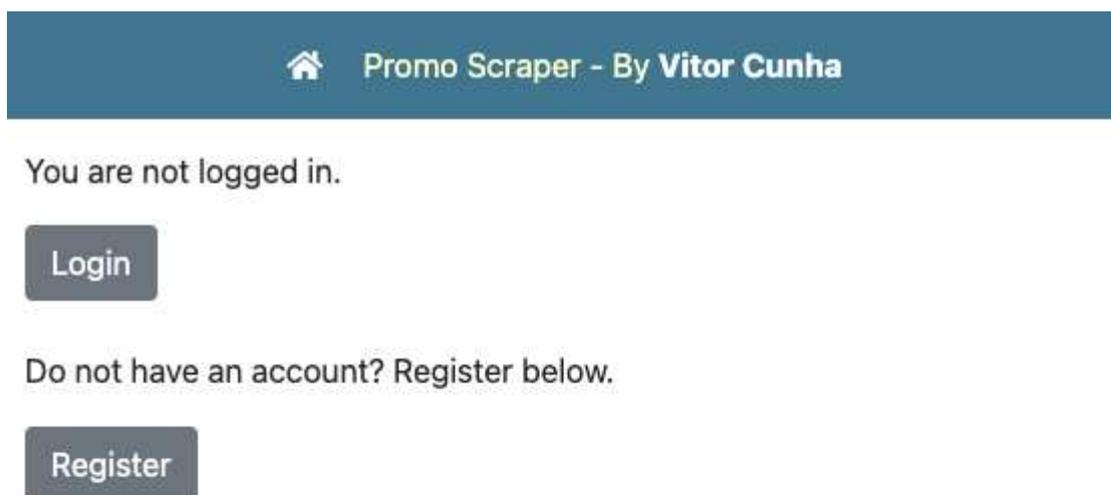


Figura 17: Formulário de cadastro

 [Promo Scraper - By Vitor Cunha](#)

## Registration form

Username\*

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

First name\*

Last name\*

Email\*

Required. Inform a valid email address.

Password\*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation\*

Enter the same password as before, for verification.

**Figura 18: Página inicial**



Por fim, a segunda iteração de Sprint foi considerada um êxito e atendeu aos seus critérios, que eram possibilitar o registro de usuários, bem como realizar a verificação com suas credenciais de acesso. Além disso, as páginas acessíveis somente a usuários autenticados foram configuradas, impedindo que usuários não autenticados realizassem ações restritas aos usuários autenticados.

### SPRINT 3 - FRONT END PÁGINA INICIAL

Nesta esta sprint, foi designado o esforço para desenvolver o *front end* da página principal e torná-la responsiva. Foram adicionados botões Home e Logout na parte superior esquerda e direita, respectivamente, vide Figura 19. Além disso, o layout também foi finalizado, possibilitando a inserção de palavras-chave e selecionar o intervalo de varredura (o intervalo para o algoritmo utilizando *scraping* procurar por *matches*).

Figura 19: Página principal Promo Scraper



# Promo Scraper

By [Vitor Cunha](#)

How does it work?

Add words for products and brands of your interest (maximum 8).

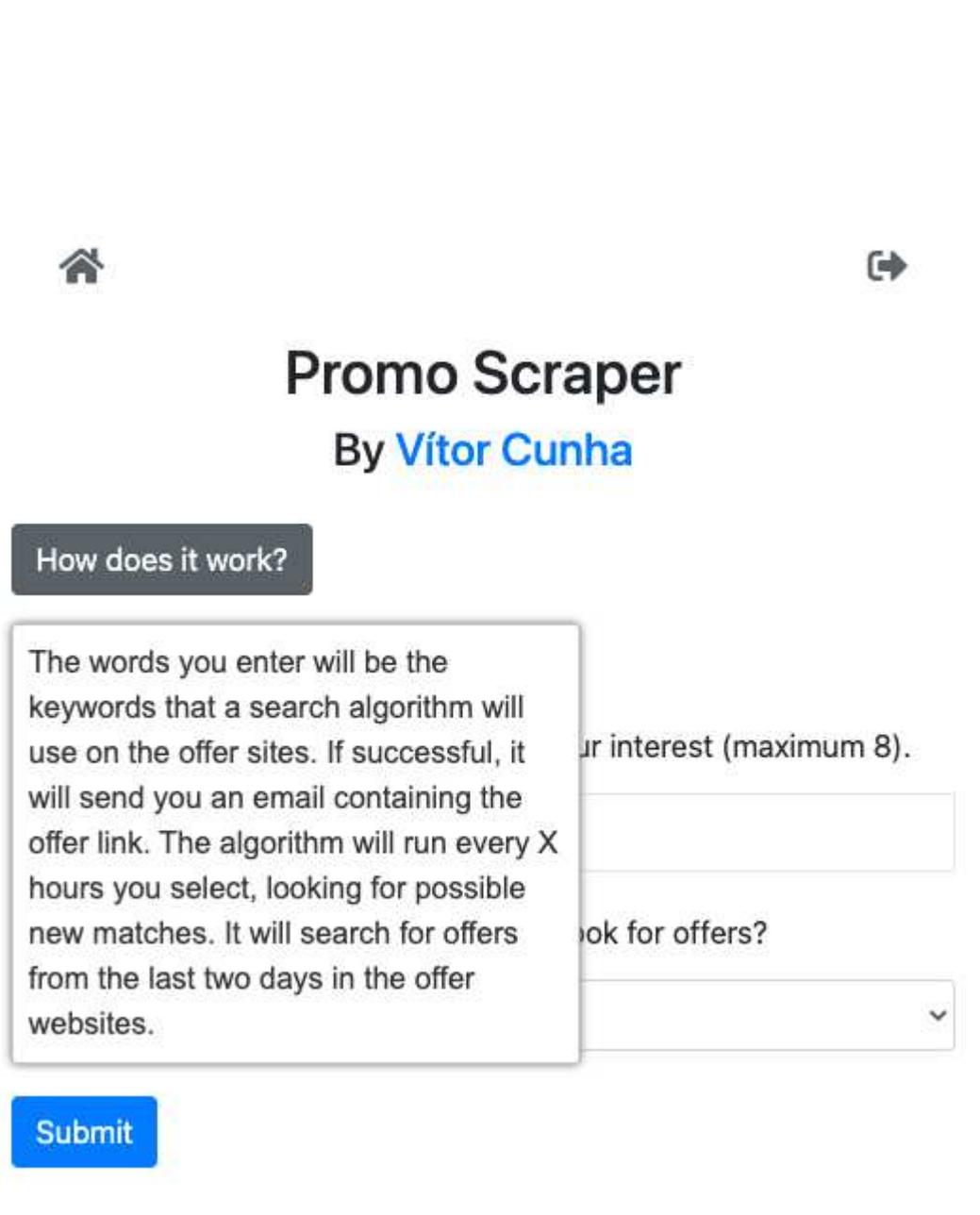
How often do you want the algorithm to look for offers?

[Submit](#)

## Check out matches we already found for you!

Matches found: 3

Figura 20: Mouse em cima do botão explicativo



Como pode ser observado nas figuras 19 e 20, o front end da página inicial foi finalizado, bem como a responsividade da página, podendo ser renderizada em celulares, tablets ou monitores de qualquer resolução. Sendo assim, esta foi considerada uma Sprint com êxito.

SPRINT 4 - ALGORITMO DE VARREDURA GATRY

Nesta sprint, foi designada a tarefa de escrever o algoritmo de *scraping* para realizar a varredura do website "gatry.com" (veja figura 21).

**Figura 21: Website Gatry**

The screenshot displays the Gatry website interface. At the top, there is a navigation bar with the 'Gatry' logo, a plus sign, and a user profile icon. Below this, there are links for 'Avaliações', 'Promoções' (with a red notification badge '2'), 'Cupons', and 'Livre' (with a red notification badge '25'). A search bar is located below the navigation, with the text 'Pesquisar em Promoções' and a magnifying glass icon. To the right of the search bar is a green button labeled 'Criar Alerta'. The main content area features four promotional items, each with an image, title, price, and user comment:

- Item 1:** Vonder Chave De Impacto Bateria Intercambiável De 18 V Encaixe De 1/2' Sem Bateria E Sem Carregador Iciv 1807. Price: R\$ 392,20. Comment: 'Achei caro'. 6 Comentários. Link: 'Ir para Amazon'. User: 3 minutos atrás.
- Item 2:** MLB.TV - Temporada Completa. Price: R\$ 291,00. Comment: 'Curling hahaha'. 19 Comentários. Link: 'Ir para MLB'. User: 5 minutos atrás.
- Item 3:** Equabio Calcio MDK 60 Cápsulas. Price: R\$ 1,99. Comment: 'Vencimento próximo'. 1 Comentário. Link: 'Ir para Farmadelivery'. User: 8 minutos atrás.
- Item 4:** Kit Xícaras 80ml Com Pires Rosa Transparente 6 Unidades Crippa 403025-010. Price: R\$ 49,30. Comment: (None). Link: 'Ir para Amazon'. User: 21 minutos atrás.

At the bottom of the promotional list, there are two sorting options: 'Últimas Promoções' (with a downward arrow) and 'Comentários' (with an upward arrow).

Fonte: gatry.com, 2023.

Como já mencionado na seção 5.6.2, a biblioteca utilizada para realizar esta varredura, é a biblioteca em Python BeautifulSoup. Conforme a figura 22, pode-se observar como o algoritmo utilizado para fazer a varredura da árvore HTML da página "gatry.com" foi feito.

**Figura 22: Algoritmo de scraping Gatry**

```
def get_match_from_gatry_page(request, url, headers, interests_list):
    """Tries to find a match on gatry's page and populate Match object with the data found."""
    page = requests.get(url, headers=headers)
    soup_object = BeautifulSoup(page.content, 'html.parser')
    h3s = soup_object.findAll('h3')
    allow = string.ascii_letters + string.digits
    match_array = []
    for h3 in h3s:
        for word in h3.text.split():
            word = re.sub('[^%s]' % allow, '', word).lower()
            if word in interests_list:
                """This is basically where the match happens. If there's a match, we'll
                try to get the name, link, coupon, price and image of the product."""
                name = h3.text
                link = h3.contents[0].attrs['href']
                coupon = handle_gatry_coupon(h3.parent)
                price = handle_gatry_price(h3.parent)
                image = h3.parent.parent.find('div', {'class': 'image'}).find('a').find('img').attrs['src']
                match_array = [name, price, link, coupon, image]

    print("Name: "+match_array[0]+"
    Price: "+str(match_array[1])+"
    Link: "
    +match_array[2]+"
    Coupon: "+match_array[3]+"
    Image: "+match_array[4])
```

A saída para a execução deste algoritmo, usando a palavra-chave "smart" como teste, foi a seguinte:

**Figura 23: Output da execução algoritmo varredura Gatry**

```
[14/Jun/2023 14:33:43] "GET /core/ HTTP/1.1" 200 10047
Name: Smart TV Samsung 55" QLED 4K Modo Game Som Em Movimento Tela Sem Limites 55Q60C 2023
Price: 2819.06
Link: https://www.girafa.com.br/p/smart-tv-samsung-55-qled-4k-modo-game-som-em-movimento-tela-sem-limites-q60c-2023.htm?utm\_medium=afiliados&utm\_campaign=33936110&utm\_source=lomadee&lmsid=NTcyNywzNTk2MTY4NywxNjg2NzQ5ODM3NzI2LG51bGwsNzg5NywxNjU2MGI5YzZhNCwlmhdjacfrlaalrfweal
Coupon:
Cupom: QUER06
Image: https://cdn.gatry.com/gatry-static/promocao/imagem/377f8c335a925ed2f145a6fb3132d1aa.jpg
[14/Jun/2023 14:34:02] "POST /core/ HTTP/1.1" 200 10047
```

Como pode ser observado na figura 23, o *scraper* conseguiu fazer a varredura do website e também juntar, formatar e mostrar todas as informações relevantes sobre o produto. Mas, para que isso fosse possível, alguns tratamentos tiveram que ser feitos.

Alguns desafios foram encontrados para conseguir chegar neste resultado, primeiramente, através da análise do HTML da página em questão, foi feita uma estratégia para buscar os elementos h3, e iterar sobre eles num laço de repetição, e para verificar se foi uma combinação de fato (Match) foram utilizadas expressões regulares. Ou seja, se dentro dessas tags h3 fosse encontrado uma palavra igual a palavra-chave proposta pelo usuário (neste caso "smart") o algoritmo entraria nessa tag h3 e começaria a buscar por todas as informações relevantes para o produto, como: nome, preço, link para loja do produto, cupom e imagem.

Outro desafio encontrado, foi a formatação desses dados, para que pudessem ser utilizados e vistos de uma forma mais amigável. O preço, por exemplo, teve que ser tratado com a função "handle\_gatry\_price()", individualmente, para que fosse formatado e mostrado corretamente (vide figura 24).

**Figura 24: Função para ajustar preço encontrado Gatry**

```
1 usage  👤 Vitor Cunha *
def handle_gatry_price(parent):
    price = parent.find('p', {'class': 'price'}).text
    match = re.search('[\d\s.]*\d', price)
    if match:
        price = match.group(0)
        # For some reason, Gatry's price is in the format 1.234,56.
        # So we need to remove the dot.
        price = price.replace('.', '')
        price = Decimal(price.replace(',', '.'))
        return price
    else:
        return None
```

Além disso, outro campo que se mostrou um desafio para encontrar e formatar, foi o campo "coupon". Como este campo pode, ou não, existir, uma função específica foi criada para fazer o tratamento desse dado, bem como popular o array, caso fosse encontrado (vide figura 25).

**Figura 25: Função para lidar com cupom**

```
1 usage  👤 Vitor Cunha
def handle_gatry_coupon(parent):
    coupon = parent.find('p', {'class': 'comment text-break'})
    if coupon:
        allow = string.ascii_letters + string.digits
        for word in coupon.text.split():
            word = re.sub('[^%s]' % allow, '', word).lower()
            if word == 'cupom':
                return coupon.text.replace(' ', '')
    else:
        return None
```

Por fim, apesar de vários desafios terem sido encontrados nesta sprint, o resultado final foi satisfatório, e a funcionalidade principal do sistema, que é a de varrer os dados dos websites, foi concluída com sucesso.

## SPRINT 5 - ALGORITMO DE VARREDURA BOLETANDO

Nesta sprint, foi designada a tarefa de escrever o algoritmo de *scraping* para realizar a varredura do website "boletando.com", conforme figura 26.

Figura 26: Website Boletando

The screenshot displays the Boletando website interface. At the top, there is a navigation bar with the Boletando logo, social media icons, a search bar labeled 'Pesquisar Ofertas', and buttons for 'Enviar Oferta' and 'Entrar'. Below the navigation bar, a horizontal menu lists various product categories: Recentes, Populares, Lançamentos, Notebooks, Smartphones, Tvs, Periféricos, Consoles, Placas de Vídeo, Cupons, and Mais Comentadas. The main content area is a grid of six product listings. Each listing includes a product image, price, title, description, promotional text, a 'IR PARA LOJA' button, a coupon code (AMOR25), the source (e.g., AliExpress, Amazon, Carrefour), and the time since the offer was posted. The products shown are: 1. AMD Ryzen 5 5500 processor (R\$ 388,85, AliExpress, 3 minutos atrás). 2. Logitech G435 headset (R\$ 235,54, AliExpress, 30 minutos atrás). 3. Samsung Neo QLED 55" TV (R\$ 3.990,00, Carrefour, 13 horas atrás). 4. Dell P2722H monitor (R\$ 1.599,00, Amazon, 14 horas atrás). 5. Samsung Galaxy S21 FE 5G smartphone (R\$ 1.899,00, Ponto, 15 horas atrás). 6. TGT Altay T3H 23.8" gaming monitor (R\$ 739,90, Amazon, 15 horas atrás).

Conforme a figura 27, pode-se observar como o algoritmo utilizado para fazer a varredura da árvore HTML da página "boletando.com" foi feito.

Figura 27: Algoritmo de scraping Boletando

```
def get_match_from_boletando_page(request, url, headers, interests_list):
    """Tries to find a match on gatry's page and populate Match object with the data found."""
    page = requests.get(url, headers=headers)
    soup_object = BeautifulSoup(page.content, 'html.parser')
    articles = soup_object.findAll('article')
    allow = string.ascii_letters + string.digits
    match_array = []
    for article in articles:
        for word in article.find('h3').find('a').text.split():
            word = re.sub('[^%s]' % allow, '', word).lower()
            if word in interests_list:
                """This is basically where the match happens. If there's a match, we'll
                try to get the name, link, coupon, price and image of the product."""
                name = article.find('h3').find('a').text
                coupon = ''
                if article.find('div', {'class': 'rehub_offer_coupon'}):
                    coupon = article.find('div', {'class': 'rehub_offer_coupon'}).find('span').text
                link = article.find('h3').find('a').attrs['href']
                price = handle_boletando_price(article.find('span', {'class': 'rh_regular_price'}).text)
                image = article.find('figure').find('img').attrs['src']
                match_array = [name, price, link, coupon, image]

    print("Name: " + match_array[0] + "\nPrice: " + str(match_array[1]) + "\nLink: "
          + match_array[2] + "\nCoupon: " + match_array[3] + "\nImage: " + match_array[4])
```

A saída para a execução deste algoritmo, usando a palavra-chave "processador" como teste, foi a seguinte:

**Figura 28: Output da execução algoritmo varredura Gatry**

```
Django version 3.1, using settings 'promo_scraper.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
Name: Processador AMD Ryzen 5 5500, Cache 19MB, 3.6GHz (4.2GHz Max
Turbo), AM4
Price: 388.85
Link: https://boletando.com/processador-amd-ryzen-5-5500-cache-19mb
-3-6ghz-4-2ghz-max-turbo-am4/
Coupon: AM0R25
Image: https://boletando.com/wp-content/uploads/2023/06/r5-5500
-150x150.jpg
Name: Smartphone Samsung Galaxy S21 FE 5G Preto 128GB, 6GB RAM, Tela
Infinita 6.4", Câmera Traseira Tripla, Android 11 e Processador
Octa-Core + Carregador
Price: 1899.00
Link: https://boletando.com/smartphone-samsung-galaxy-s21-fe-5g
-preto-128gb-6gb-ram-tela-infinita-6-4-camera-traseira-tripla
-android-11-e-processador-octa-core-carregador/
Coupon:
Image: https://boletando
.com/wp-content/uploads/2022/01/e61df17d4e04c633f36b090c51002c08
-200x150.jpg
[14/Jun/2023 16:48:57] "POST /core/ HTTP/1.1" 200 12814
```

Como pode ser observado na figura 28, a saída do programa de varredura do site Boletando foi concluída com êxito. Podemos observar, que inclusive, para a palavra-chave "processador" foram obtidas duas combinações (Matches).

O nível de dificuldade para realizar esta varredura, comparada com o algoritmo de varredura do website "gatry.com" foi menor, visto que boa parte da lógica e estrutura para realizar a varredura da árvore HTML foi similar ao do algoritmo anterior.

## SPRINT 6 - POPULANDO O BANCO E MOSTRANDO NO FRONT END

Como já mencionado na seção 4.5, os modelos (*models*), em django, são a representação do banco de dados, contendo seus respectivos campos e

relacionamentos entre si. Nesta sprint, o objetivo foi de adequar os algoritmos de varredura e as informações encontradas, com os modelos do banco de dados. Isto é, após uma ou mais promoções serem encontradas, um objeto Match deveria ser criado e armazenado no banco de dados, lembrando que, cada objeto User poderá ter vários objetos Match, e um objeto Match terá apenas um objeto User, que será vinculado pela chave estrangeira "user\_id", conforme mostrado anteriormente na figura 13.

Agora, quando o usuário clica no botão "submit", isto será uma *request* POST para o servidor, e isso iniciará o algoritmo de varredura imediatamente. Além disso, após a varredura, se uma ou mais "matches" forem encontradas, o objeto Match será populado no banco de dados, e o template mostrará as informações dos Matches encontrados no front end, como pode ser observado na figura 29.

Figura 29: Página inicial mostrando objetos do banco de dados

🏠 ↶

## Promo Scraper

By [Vitor Cunha](#)

[How does it work?](#)

Add words for products and brands of your interest (maximum 8).

monitor ✕ perfume ✕ Press "enter" or "tab" to confirm

How often do you want the algorithm to look for offers?

Every 6 hours▼

[Submit](#)

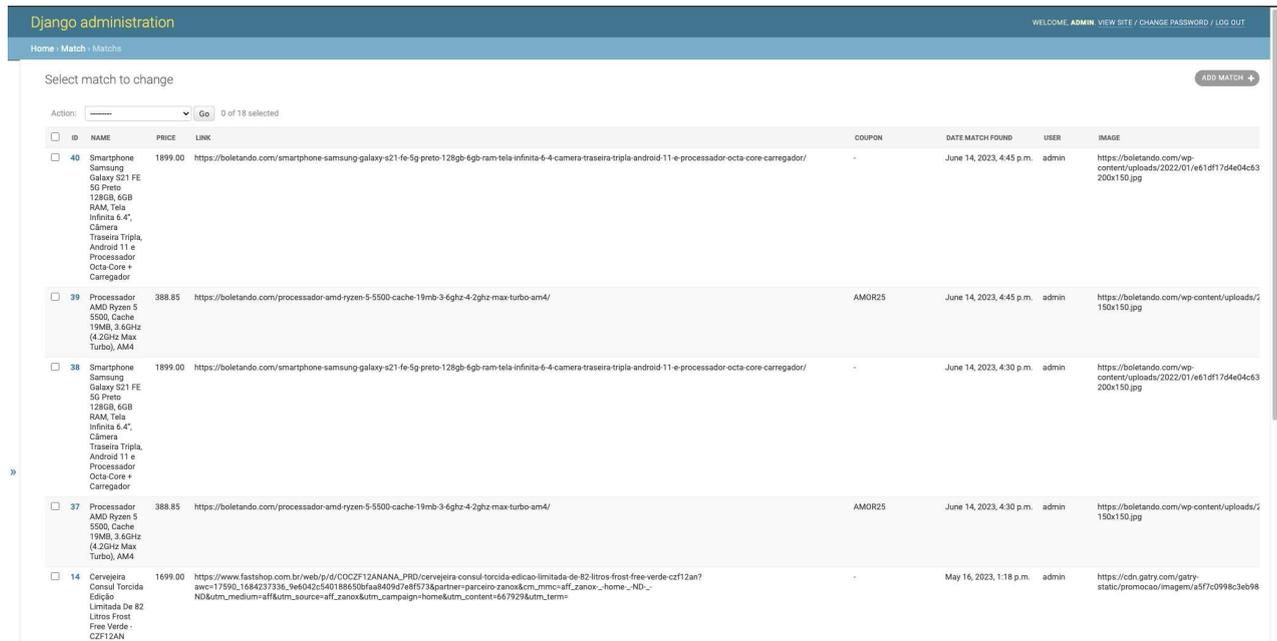
### Check out matches we already found for you!

Matches found: 2

<p><b>Monitor Gamer LG 26"</b> IPS Ultra Wide 75Hz Full HD 1ms FreeSync Premium HDR10 - 26WQ500</p> <p>Price: R\$ 759.99</p> <p>Cupom:UP5MONITOR</p> <p><a href="#">Product link</a></p> 	<p><b>Acqua For Men Orgânica</b> Eau De Toilette Perfume Masculino 50 MI, Organica</p> <p>Price: R\$ 65.68</p> <p>No Coupon found.</p> <p><a href="#">Product link</a></p> 
--	---

Também é possível observar os objetos do *back end* pelo painel de administrador do django, bem como editar, remover ou adicionar outros objetos, conforme mostra a figura 30.

**Figura 30: Painel django admin listando objetos Match**



## SPRINT 7 - PROJETO EM PRODUÇÃO (LIVE) HEROKU

Com o intuito de poder compartilhar com qualquer pessoa, e também com o pensamento de abrir este projeto a público e torná-lo *open-source*, o objetivo desta sprint será adequar as configurações do projeto, para que ele seja colocado em produção. Para tal, como já explicado na seção 4.6.5, a plataforma utilizada para isto será o Heroku. A própria documentação do Heroku é robusta e mostra tutoriais de como colocar uma aplicação de django em produção.

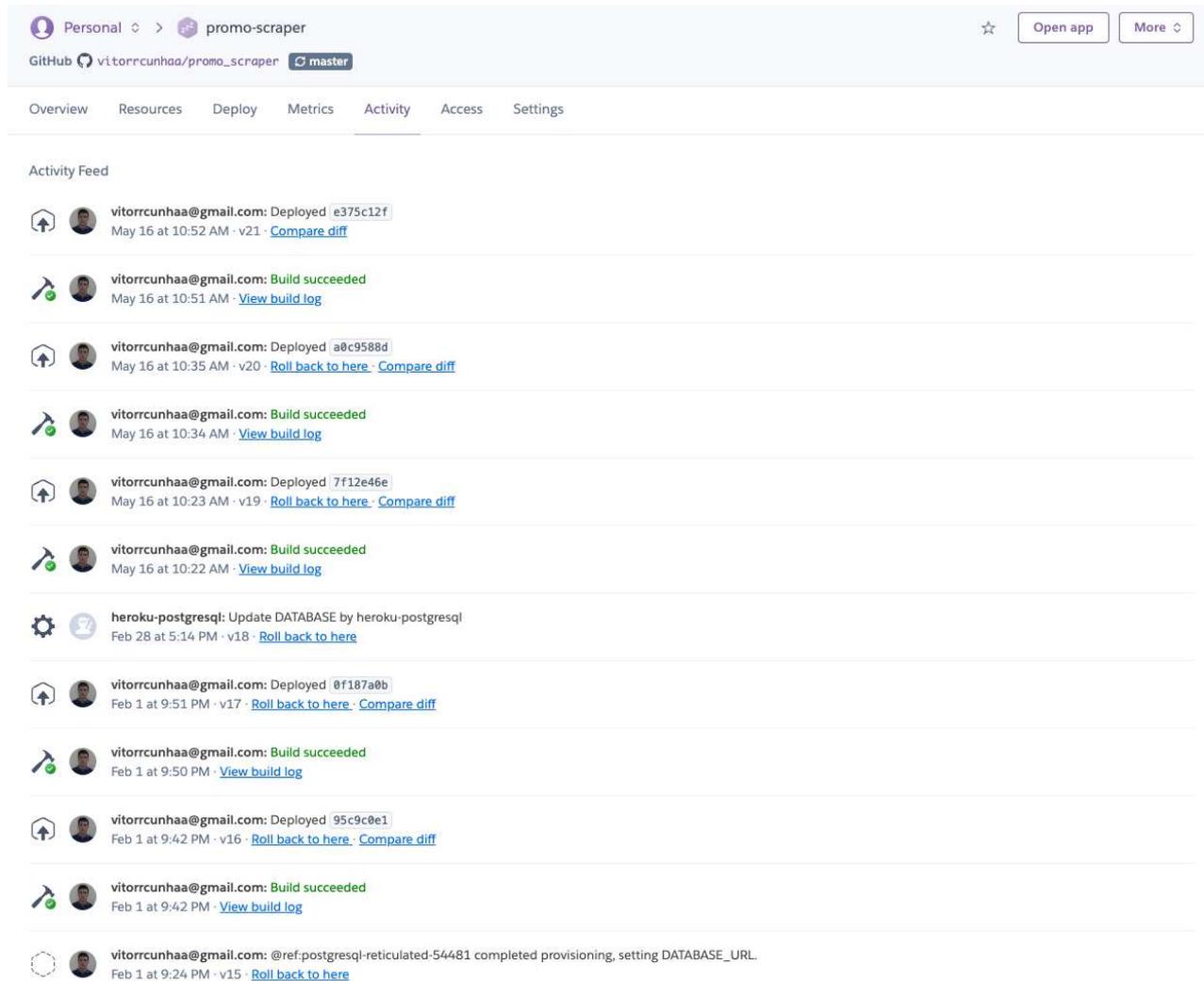
Para atingir este objetivo, eis o que foi feito :

- Configuração do Ambiente Local: Verificou-se se a aplicação Django estava funcionando corretamente localmente. Todas as funcionalidades foram testadas e não foram identificados erros. Verificou-se também se o Git estava instalado no sistema para gerenciar o controle de versão do código-fonte.

- O Arquivo de Requisitos foi configurado: Foi criado um arquivo requirements.txt na raiz do projeto Django. Esse arquivo lista todas as dependências necessárias para a aplicação, incluindo o Django e outras bibliotecas utilizadas. O arquivo foi gerado automaticamente executando o comando "pip freeze > requirements.txt".
- Configuração do arquivo Procfile: Foi criado um arquivo chamado Procfile na raiz do projeto. Esse arquivo especificou os comandos que seriam executados pelo Heroku para iniciar a aplicação.
- Configuração do Banco de Dados: As credenciais de acesso ao banco foram configuradas na seção correspondente do arquivo "settings.py". É importante ressaltar que, no Heroku, o banco de dados foi gerenciado por um serviço adicional, chamado Heroku Postgres. Então localmente banco de dados é apenas um arquivo .sqlite, enquanto em produção, há um banco de dados real PostgreSQL.
- Criação do projeto no Heroku: Foi criada uma conta no Heroku e fez-se o login no painel de controle. Um novo aplicativo foi criado no, fornecendo um nome exclusivo para a aplicação. Na guia "Deploy", a opção de integração com o Git foi selecionada e o repositório Git que continha o projeto Django foi vinculado.
- Configuração das variáveis de ambiente: Na guia "Settings" do painel de controle do Heroku, foi acessada a seção "Config Vars" e foram configuradas as variáveis de ambiente necessárias para a aplicação. Isso pode incluir chaves de API, configurações específicas do ambiente, entre outros.
- Deploy da aplicação: Utilizou-se o Git para fazer o push do código-fonte para o repositório remoto do Heroku. O seguinte comando foi executado no terminal: "git push heroku main".
- Migrações do banco de dados: Após o deploy, as migrações do banco de dados foram executadas. No terminal, o seguinte comando foi executado: "heroku run python manage.py migrate".
- Escalonamento da aplicação: No painel de controle do Heroku, acessou-se a guia "Resources" e adicionou-se o serviço de banco de dados (como o Heroku Postgres) caso ainda não tivesse sido feito. Certificou-se também que a

aplicação estava escalada para pelo menos uma instância para que pudesse ser acessada.

**Figura 31: Painel do heroku mostrando o histórico de *builds***



E assim, o acesso à aplicação no Heroku foi garantido, e a mesma foi disponibilizada online para os usuários na URL <http://promo-scraper.herokuapp.com/> como pode ser visto na figura 31.

Figura 32: Promo Scraper rodando em produção e responsivo (iphone)

19:57 ▲ promo-scraper.herokuapp.com

Home Share

# Promo Scraper

By [Vitor Cunha](#)

How does it work?

Add words for products and brands of your interest (maximum 8).

✕ Press "enter" or "tab" to confirm

How often do you want the algorithm to look for offers?

▾

[Submit](#)

## Check out matches we already found for you!

Matches found: 3

**Paleta De Sombras Mariana Saad By Océane - 12 Shades 21,5g**

Price: R\$ 49.00

No Coupon found.

[Product link](#)

## SPRINT 8 - SERVIÇO SCHEDULER

Para que de fato o sistema se torne útil de forma passiva, isto é, rode o algoritmo recorrente com um *scheduler*, buscando todos os dias por possíveis "*matches*" baseados nas palavras-chave cadastradas pelo usuário, e assim, enviando as "*matches*" por e-mail ao usuário, será necessária a implementação do serviço *scheduler*.

A ferramenta *scheduler* escolhida, conforme já explicado na seção 4.6.5.1, é a própria ferramenta *scheduler* da plataforma Heroku. Para atingir este resultado, os seguintes passos foram feitos:

- Configuração do serviço scheduler: Um serviço scheduler foi configurado na plataforma Heroku para permitir o agendamento de tarefas automatizadas, como o disparo de e-mails. Esse serviço é responsável por executar um script Python periodicamente, de acordo com a programação definida.
- A instalação da biblioteca ezgmail foi instalada: A biblioteca ezgmail, que facilita o envio de e-mails através da API do Gmail, foi instalada no ambiente de desenvolvimento.
- Configuração das credenciais do gmail foi realizada: As credenciais do Gmail foram configuradas para permitir o envio de e-mails autenticados. Isso incluiu a criação de uma conta de serviço no Google Cloud Platform e a geração de um arquivo JSON contendo as informações de autenticação.
- Criação do script de envio de e-mails foi: Um script em python foi criado para enviar os e-mails desejados utilizando a biblioteca ezgmail. Esse script incluiu a importação da biblioteca, a autenticação utilizando as credenciais do Gmail e a definição do conteúdo do e-mail, como destinatário, assunto e corpo da mensagem.
- Configuração do script no serviço scheduler do Heroku: O script de envio de e-mails foi configurado para ser executado periodicamente pelo serviço scheduler do Heroku. As configurações de agendamento foram definidas de acordo com o intervalo escolhido pelo usuário (6, 12 ou 24 horas).
- Implantação da Aplicação no Heroku: A aplicação, juntamente com o script de envio de e-mails e as configurações do serviço scheduler, foi implantada na

plataforma Heroku. Isso inclui o uso do Git para fazer o push do código-fonte para o repositório remoto do Heroku.

- Iniciação do serviço Scheduler: Após a implantação, o serviço scheduler foi iniciado na plataforma Heroku para que os disparos de e-mails automatizados pudessem ocorrer de acordo com a programação definida.
- Envio automático de e-mails: Com o serviço scheduler em execução, o script de envio de e-mails foi executado nos horários agendados, e os e-mails foram enviados automaticamente aos destinatários especificados.
- O Monitoramento e Verificação dos Envios foram realizados: Os envios de e-mails foram monitorados para garantir que estavam ocorrendo conforme o esperado. Foram realizadas verificações regulares para garantir que os e-mails estavam sendo entregues corretamente aos destinatários. Conforme pode ser observado na figura 33.

### Figura 33: Mensagem de e-mail com uma promoção encontrada

Promo scraper has found a promotion for you!  Caixa de entrada x



**Vitor Cunha** <vitorrcunhaa@gmail.com>

para mim ▾

According to your keyword: monitor.

The following promotion was found:



Monitor Gamer LG 26" IPS Ultra Wide 75Hz Full HD 1ms FreeSync Premium HDR10 - 26WQ500

Price: R\$ 759.99

Coupon: UP5MONITOR

Product link: [https://www.kabum.com.br/produto/371253/monitor-gamer-lg-26-ips-ultra-wide-75hz-full-hd-1ms-freesync-p3db0883ae160780354872a5c1ecacb0e&utm\\_source=AWIN&utm\\_medium=AFILIADOS&utm\\_campaign=PS5FIFA\\_out2](https://www.kabum.com.br/produto/371253/monitor-gamer-lg-26-ips-ultra-wide-75hz-full-hd-1ms-freesync-p3db0883ae160780354872a5c1ecacb0e&utm_source=AWIN&utm_medium=AFILIADOS&utm_campaign=PS5FIFA_out2)

Best Regards,  
Promo Scraper Team

↩ Responder

➡ Encaminhar

## APÊNDICE B - ARTIGO DO TRABALHO

# Aplicação Web para detecção e notificação de promoções com Web Scraping e disparo de e-mail

Vítor de Souza Cunha

<sup>1</sup>Departamento de Sistemas de Informação – Universidade Federal de Santa Catarina (UFSC)  
Florianópolis – SC – Brasil  
vitorrcunhaa@gmail.com

**Abstract.** *This work proposes to develop a web application that will scan the items specified by the user, in two non-profit websites, containing the best promotions, and will send e-mails to the user, alerting if one (or more) of your wish items has been found. The websites in question are powered by the community that seeks and shares promotions. The promotions are posted by any user registered in them and with the purpose of helping potential consumers, without any type of financial interest or partnerships.*

**Resumo.** *Este trabalho se propõe a desenvolver uma aplicação web que irá fazer uma varredura pelos itens especificados pelo usuário, em dois websites sem fins lucrativos, contendo as melhores promoções, e irá disparar e-mails ao usuário, alertando se um (ou mais) de seus itens de desejo foi encontrado. Os websites em questão são alimentados pela própria comunidade que busca e compartilha promoções. As promoções são postadas por qualquer usuário cadastrado nos mesmos e com a finalidade de ajudar potenciais consumidores, sem qualquer tipo de interesse financeiro ou parcerias.*

## 1. Introdução

No momento em que o estudo foi iniciado, o Brasil e o mundo todo enfrentavam um período extremamente difícil por conta da pandemia causada pelo vírus Covid-19. Por conta disso, muitas empresas, especialmente na área de tecnologia, adotaram o *home office* como modelo de trabalho durante o isolamento. Consequentemente, o investimento em materiais de trabalho aumentaram, e continuam aumentando drasticamente, e um desses setores é o de eletrônicos, que apresenta um grande crescimento, como mostram os dados da Associação Brasileira de Comércio Eletrônico (ABComm, 2023).

Nesse cenário, o consumidor costuma procurar as melhores ofertas em toda gama de *e-commerces* que lhe interessam. Existem algumas alternativas, procurar manualmente em cada loja de seu interesse e confiança, ou recorrer a uma plataforma que compara automaticamente o preço entre todas as lojas. Algumas empresas fazem este tipo de serviço, como: Zoom e Jácomei.

A solução proposta por este trabalho tem uma abordagem diferente. O usuário primeiro irá inserir palavras-chave num formulário, que serão os itens a serem

procurados, para ser alertado apenas quando o algoritmo encontrar um ou mais desses itens em alguma publicação nos sites em que está sendo feita a varredura, sendo então, uma busca passiva.

A vantagem deste modelo é que o usuário terá acesso a promoções que, muitas vezes, estão disponíveis apenas por tempo limitado e tendem a encerrar rapidamente. Como trata-se de uma varredura feita em sites geridos por usuários comuns e que também são consumidores, há a garantia de que não haverá interesse por parte destes.

É uma forma de somar esforços de toda uma comunidade que pode estar à procura dos mesmos itens de interesse do usuário dessa aplicação em questão, e reportar com prontidão da devida oferta.

## **2. Objetivos**

### **2.1 Objetivos gerais**

Desenvolver uma aplicação web para ajudar os consumidores a encontrarem as melhores promoções de forma passiva (sem a necessidade de estar ativamente procurando), utilizando web scraping em sites de promoções crowdsourcing e notificando via e-mail quando uma ou mais promoções forem encontradas.

### **2.2 Objetivos específicos**

Como objetivos específicos foram definidos: (i) Realizar uma análise do comportamento e tendências de pesquisas e compras online; (ii) Estudar e analisar ferramentas de comparativos de preços; (iii) Estudar e analisar ferramentas de notificação de produtos e preços; (iv) Analisar, modelar e construir um sistema web de forma incremental, utilizando uma abordagem ágil adaptada com base no SCRUM; e (v) desenvolver a solução modelada, com uma interface web para visualização de promoções encontradas e notificar promoções encontradas via e-mail, salvando-as no banco de dados.

## **3. Soluções existentes**

Este capítulo tem como objetivo identificar e analisar ferramentas com soluções relacionadas, porém não idênticas à deste trabalho. Na tabela 1, pode-se observar algumas funcionalidades dos principais concorrentes no segmento de comparação de preços.

Tabela 1. Funcionalidades de soluções existentes similares

<b>ZOOM</b>	<b>JÁCOTEI</b>
Comparação de preços	Comparação de preços

Busca de promoções	Busca de promoções
Buscas e promoções de sites parceiros	Buscas e promoções de sites parceiros
Divulgação de produtos de sites parceiros	Divulgação de produtos de sites parceiros
Filtragem de produtos	Filtragem de produtos
Avaliações dos produtos	Verificação de fraudes Black Friday
Histórico de preços	Histórico de preços
Alerta de preço	Alerta de preço
Cashback	Guia de produtos e comparativo no blog
Venda de produtos dentro da plataforma	

### 3.1 Análise das soluções existentes

A análise foi realizada através da utilização de cada uma das plataformas, realizando ações de cadastro, utilização de cupons, compras teste com cashback, navegação no site de modo geral, uso dos aplicativos e uso das extensões. O uso dos mesmos foi sempre em sua fonte oficial (seja esta o próprio site, extensão, ou aplicativo oficial). Após a análise, é importante comentar as principais vantagens e desvantagens dessas plataformas, as comparando com a solução proposta por este trabalho.

A principal vantagem destes modelos é a facilidade e robustez da solução. O usuário que acessa a plataforma, tem a possibilidade de pesquisar qualquer produto e receber em poucos segundos um comparativo completo dos principais e mais confiáveis e-commerces que estão vendendo este item, e comparar o preço que cada uma dessas lojas está vendendo. Há também a opção de ver o histórico de preços desse determinado produto, que é mostrado em forma de gráfico. Para usuários cadastrados, há também a opção “Criar Alerta” que permite que o usuário seja notificado quando um item de seu interesse alcance um preço que o próprio usuário estipulou. Esta opção é a que mais se assemelha a solução proposta por este trabalho, onde o usuário não tem a intenção da compra imediata, e estaria disposto a esperar até que um produto de seu interesse

chegue a um valor mais acessível. Neste caso, o valor que o próprio usuário teria estipulado.

Uma das grandes desvantagens deste modelo, é a presença de parcerias entre diversas lojas com a plataforma Zoom, como mencionado no seu site oficial “Anunciamos ofertas de centenas de lojas parceiras em nosso site e app para auxiliar na busca pelo melhor preço.” (Zoom, 2022). Atualmente o site da Zoom possui mais de 2,5 milhões de produtos monitorados e 300 lojas parceiras (Zoom, 2022).

Desta maneira, não se tem a garantia de que todas as lojas, sejam elas parceiras ou não parceiras da Zoom, serão levadas em consideração na hora de procurar e comparar os melhores preços. Outro aspecto importante de se destacar, é o fato de se tratar de uma organização privada com objetivos financeiros bem definidos. Já no modelo de negócios da empresa JáCotei, segundo seu website oficial, está explícito os planos que as empresas podem contratar a fim de ter a visibilidade que desejam para determinados produtos (Jácotei, 2023).

#### 4. Levantamento de Requisitos

A seguir os casos de uso desenhados para a aplicação:

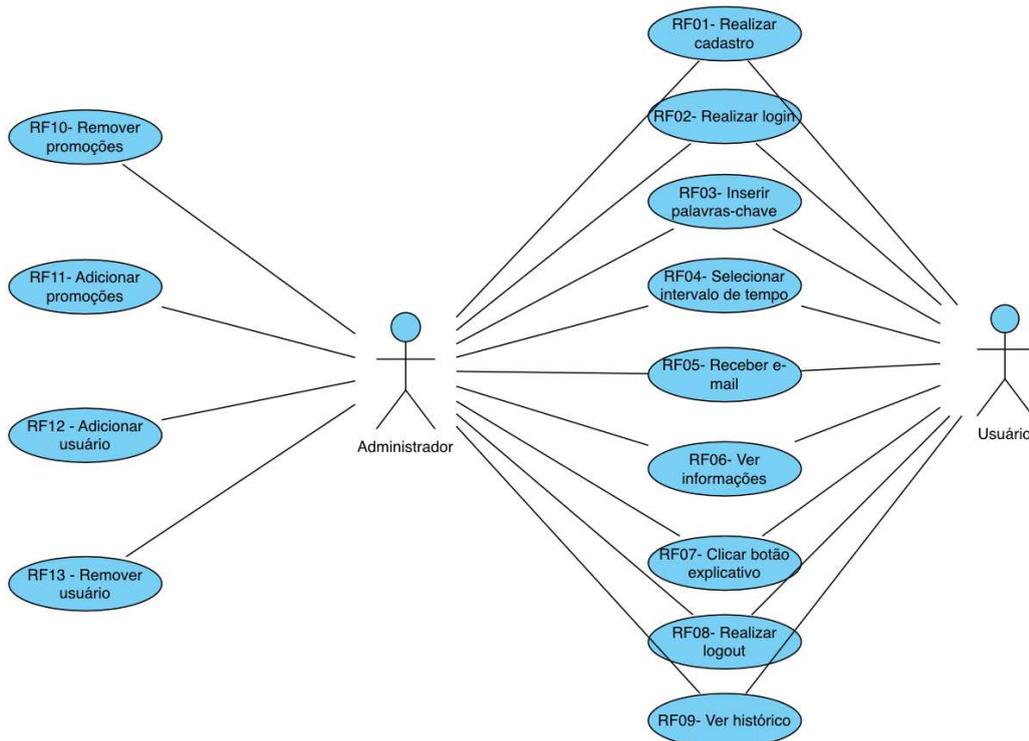


Figura 1. Diagrama de casos de uso

#### 5. Estrutura do sistema

O aplicativo web foi desenvolvido usando o framework Django, que segue o padrão MVT (Model View Template). Sendo assim, na parte "Model" estão as classes que são a abstração dos objetos no banco de dados, os principais models desta aplicação são User e Match. Para o front end, ainda fazendo parte do framework django, foi utilizado o DLT (Django Language Template) que integra facilmente com o backend (visto que é uma ferramenta do próprio framework), renderizando as views e podendo mostrar facilmente informações dos objetos do banco de dados na tela, que são trazidos para o DLT como contexto (context). Ainda no front end, Bootstrap é importado e utilizado para estilizar os elementos HTML e a própria página como um todo. O trabalho em questão, está no ar (*live*) e hospedado na plataforma Heroku, e para que isto fosse possível, alguns ajustes tiveram que ser feitos no projeto, como criar alguns arquivos de configuração, adicionar add-ons (sendo o banco de dados PostgreSQL e heroku scheduler, exemplos), especificar a buildpack de python e integrar o deploy automático com GitHub. O serviço de web scraping e de disparo de e-mail é um tipo de serviço recorrente e utiliza-se do heroku scheduler para fazer a varredura dos websites de promoção em um certo período, e caso uma promoção tenha sido encontrada, também disparar o e-mail ao usuário. Para facilitar o entendimento da arquitetura, bem como da estruturação do projeto como um todo, a comunicação entre os elementos da aplicação está detalhada na figura 2.

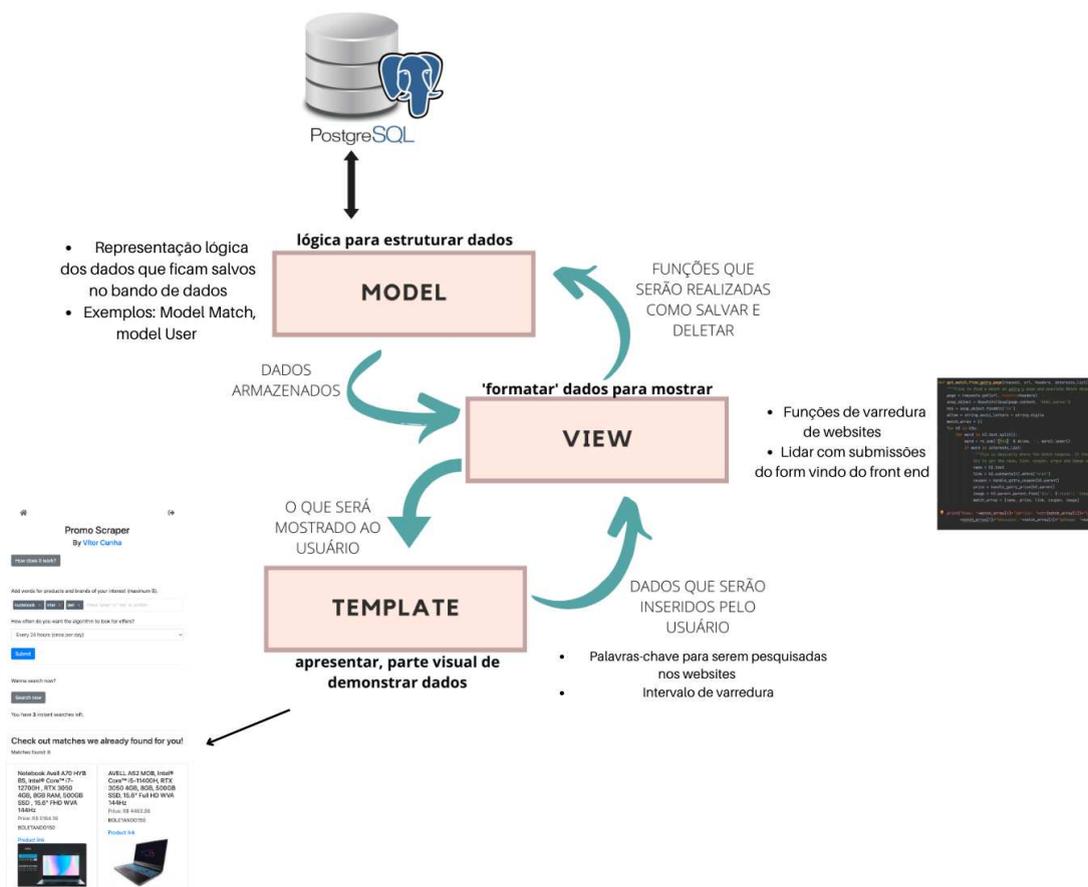


Figura 2. Arquitetura do sistema e comunicação entre MVT

## 6. Utilizando a aplicação

A aplicação foi desenvolvida pensando em qualquer tipo de tela, sendo assim, é totalmente responsiva e pode ser acessada de celulares, tablets, ou monitores de diferentes tamanhos. Para fazer login, basta clicar no botão "Login", e caso o usuário ainda não tenha uma conta, poderá criar uma clicando em "Register" e preenchendo o formulário.

Após ter uma conta criada e fazer o login, o usuário poderá preencher suas palavras-chave, ou seja, os seus itens de interesse que o algoritmo de varredura irá buscar, no campo "Add words for products or brands of your interest". E também, selecionar o intervalo de tempo que o algoritmo de varredura irá buscar pelas promoções no campo "How often do you want the algorithm to look for offers?". Assim que estes campos forem preenchidos, é possível fazer a submissão deste formulário e aguardar para que o algoritmo rode, e possivelmente, encontre alguma promoção baseada nas palavras-chave do usuário, e por fim, notifique por e-mail caso tenha encontrado uma (ou mais) promoções.

Quando promoções são encontradas, estas podem ser visualizadas na página principal, onde em forma de cartões, são mostradas as informações relevantes para cada promoção, como: título, preço, cupom, link para o produto e imagem.

Para usuários administradores, também é possível acessar o painel do administrador, onde ações de inserção, modificação ou remoção de objetos do banco de dados podem ser feitas.

Para realização do Logout, um botão a extrema direita no topo, mostrará um ícone de Logout, que quando clicado, o usuário será deslogado do sistema. Já na extrema esquerda no topo, há um botão em formato de casa, que quando clicado, redireciona o usuário a página Home de boas-vindas.

The screenshot shows the main interface of the 'Promo Scraper' web application. At the top, there are navigation icons for home and search. The title 'Promo Scraper' is centered, followed by 'By Vítor Cunha'. A button labeled 'How does it work?' is visible. Below this, a text input field contains the instruction 'Add words for products and brands of your interest (maximum 8)'. The input field shows two tags: 'monitor' and 'perfume', with a note to 'Press "enter" or "tab" to confirm'. A dropdown menu asks 'How often do you want the algorithm to look for offers?' and is set to 'Every 6 hours'. A blue 'Submit' button is positioned below the dropdown. The main content area is titled 'Check out matches we already found for you!' and shows 'Matches found: 2'. Two product cards are displayed side-by-side. The first card features a 'Monitor Gamer LG 26" IPS Ultra Wide 75Hz Full HD 1ms FreeSync Premium HDR10 - 26WQ500' with a price of R\$ 759.99 and a coupon code 'UP5MONITOR'. The second card features 'Acqua For Men Orgânica Eau De Toilette Perfume Masculino 50 Ml, Organica' with a price of R\$ 65.68 and a note 'No Coupon found.'. Both cards include a 'Product link' and a small image of the product.

Figura 3. Página principal da aplicação web Promo Scraper

## 7. Conclusão

Devido ao exposto, observa-se que o projeto em questão se mostra uma alternativa viável para os consumidores no âmbito da internet. Ainda que não possa substituir outras plataformas similares mais robustas com soluções sofisticadas, no aspecto de alerta sobre possíveis promoções, olhando apenas websites alimentados pela comunidade, é uma alternativa viável.

Embora nem todos os requisitos funcionais tenham sido completamente entregues, os objetivos deste projeto foram considerados um sucesso. A análise do

comportamento e tendências de pesquisas e compras online não teve um teor desafiador, não obstante, foi de grande importância para que o embasamento sobre o projeto tivesse uma boa fundamentação e para entender como o mercado e-commerce vem se comportando.

O estudo e análise das ferramentas comparativas de preços já existentes também foram importantes para a confecção deste projeto, visto que, de uma perspectiva de complexidade técnica, vasta variedade de funcionalidades, e um modelo de negócios sólido, foi necessário o uso e entendimento destas ferramentas, bem como compará-las com o que este projeto se dispôs a desenvolver.

Sem dúvidas a parte mais desafiadora foi o desenvolvimento da solução do aplicativo web. O estudo e leitura de documentações, todo o processo de desenvolvimento do software como um todo, da modelagem até o deploy, foi bastante desafiador e teve uma exigência técnica grande, para que todos os elementos de software funcionassem corretamente. O aprendizado sobre as tecnologias de web scraping, o próprio framework django, o design do front end, a implementação de notificações recorrentes, e além de tudo, colocar o projeto no ar, fazendo o deploy no Heroku, foi enorme, porém extremamente gratificante ao mesmo tempo.

O projeto continua em fase de implementação, porém já cumpre grande parte dos requisitos funcionais e não funcionais. Além disso, o projeto tem pretensão de permanecer open-source e está disposto a receber novas modificações e sugestões da comunidade.

Para trabalhos futuros, foram identificadas algumas funcionalidades que tornariam o sistema mais completo, e com mais funcionalidades para o usuário, sendo estas: (i) Adição de mais websites a serem varridos: Com mais websites a serem varridos, maior será a possibilidade de encontrar uma promoção que combine com as palavras chave cadastradas pelo usuário; (ii) Refatorar o sistema de procura de palavras compostas, com expressões regulares: Existem melhorias a serem feitas para encontrar palavras chave compostas ou produtos de duas ou mais palavras cadastradas pelo usuário, por exemplo: Placa de vídeo, guarda roupa, micro-ondas; (iii) Implementação da página de histórico com gráficos: A implementação da página de histórico de promoções encontradas, juntamente com gráficos para ajudar na visualização do preço ao longo do tempo; (iv) Implementação da funcionalidade de pesquisa instantânea: A implementação de uma busca ativa, apesar de não ser o foco deste trabalho, também seria de grande ajuda para o usuário, uma vez que ele poderia querer fazer uma pesquisa ativa, ainda que tivesse um limite diário de pesquisas ativas (para não sobrecarregar o sistema).

## **8. Referências**

EXPECTATIVA de vendas no ecommerce. Disponível em: <https://dados.abcomm.org/previsao-de-vendas-online>. Acesso em: fevereiro 2023.

ZOOM. O que é o Zoom? Disponível em: <<https://tiresuaduvida.zoom.com.br/hc/pt-br/articles/360000942023-O-que-%C3%A9-o-Zoom->>. Acesso em: setembro 2022.

ZOOM. Como usar o Cashback Zoom? Disponível em: <<https://www.zoom.com.br/landing-page/tire-suas-duvidas-sobre-o-cashback-zoom>>. Acesso em: setembro 2022.

ZOOM. Cupons de desconto, ofertas e cashback Disponível em: <<https://www.zoom.com.br/cupom-de-desconto>>. Acesso em: setembro 2022.

JACOTEI. Inclua sua loja no JáCotei Disponível em: <<https://www.jacotei.com.br/sualoja/>>. Acesso em: setembro 2022.

JACOTEI. Compare preços e ainda ganhe prêmios em milhas. Disponível em: <<https://www.techtodo.com.br/tudo-sobre/jacotei-app/>>. Acesso em: fevereiro 2023.

## APÊNDICE C - CÓDIGO DA APLICAÇÃO

Código da aplicação: [https://github.com/vitorrcunhaa/promo\\_scraper](https://github.com/vitorrcunhaa/promo_scraper)