

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA  
ENGENHARIA DE SOFTWARE

**ELISEU XAVIER BERNARDO SANTOS**  
**PEDRO FELIPE DE MORAES CARRIJO**

COMPARADOR DE PREÇOS DE SKINS DO CSGO UTILIZANDO WEB  
SCRAPING

Anápolis – GO  
Dezembro, 2021

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA  
ENGENHARIA DE SOFTWARE

**ELISEU XAVIER BERNARDO SANTOS**  
**PEDRO FELIPE DE MORAES CARRIJO**

COMPARADOR DE PREÇOS DE SKINS DO CS GO UTILIZANDO WEB SCRAPING

Trabalho apresentado ao Curso de Engenharia de Software da Universidade Evangélica de Goiás – UniEVANGÉLICA, da cidade de Anápolis-GO como requisito parcial para obtenção do Grau de Bacharel em Engenharia de Software.

Orientador (a): Prof. Alexandre Moraes Tannus

Anápolis -GO  
Dezembro, 2021

## RESUMO

As ferramentas de comparação de preços ajudam os utilizadores de internet a identificar o preço de produtos variados de diferentes lojas e-commerce. Com a presença dessas aplicações o mercado tende a padronizar os preços e aumentar a concorrência, pois eles ajudam os usuários a realizar suas compras, localizando as melhores oportunidades de ofertas. Desta forma, pretende-se implementar um comparador de preços no mercado de *skins* do jogo counter-strike: global offensive (cs:go), que vem crescendo e consequentemente abrindo um espaço maior para novas lojas e-commerce. O projeto pretende colocar em prática a potencialidade tecnológica existente, utilizando a extração de informação da internet, por meio do web scraping que proporcionarão a indexação de informações dos produtos, sendo essas informações exibidas em um web site de comparação de preços. O parecer final da aplicação será realçar a diferença de preços e possíveis promoções.

**Palavras-chave:** Web Crawler. Web Scraping. Comparador de preços. Serviços de pesquisa. Counter Strike Global Offensive.

## **ABSTRACT**

Price comparison tools help internet users to identify the price of different products from different e-commerce stores. with the presence of these applications, the market tends to standardize prices and increase competition, as they help users to make their purchases, locating the best offer opportunities. in this way, it is intended to implement a price comparator in the counter-strike game skins market: global offensive (cs:go), which has been growing and consequently opening a greater space for new e-commerce stores. the project intends to put into practice the existing technological potential, using the extraction of information from the internet, through web scraping, which will provide the indexing of product information, with this information being displayed on a price comparison website. the final opinion of the application will highlight the difference in prices and possible promotions.

**Keywords:** Web Crawler, Web Scraping. Price Comparator. Search Services. Counter Strike Global Offensive.

## LISTA DE ILUSTRAÇÕES

Ilustração 1 – Web Crawler vs Web Scraping	17
Ilustração 2 – Tela de comparação de preço CS:GO Stash	23
Ilustração 3 – Protótipo da tela de pesquisa	25
Ilustração 4 – Protótipo da tela de listagem de produtos	26
Ilustração 5 – Página inicial BatSkins	28
Ilustração 6 – Modelo entidade relacional	29
Ilustração 7 – Importação das bibliotecas	29
Ilustração 8 – Estrutura HTML da Batskins	29
Ilustração 9 – Tags e Classes dos produtos da loja Batskins	30
Ilustração 10– Função Scraping	31
Ilustração 11 – Input do para buscas de dados	32
Ilustração 12 – Script SQL para criação e preenchimento da tabela	32
Ilustração 13 – Tela de comparação de preços	33

**LISTA DE TABELAS**

Tabela 1 – Requisito Funcional 001 .....	24
Tabela 2 – Requisito Funcional 002 .....	24
Tabela 3 – Casos de teste .....	27

## LISTA DE ABREVIATURAS E SIGLAS

<b>Siglas</b>	<b>Descrição</b>
CS:GO	Counter Strike Global Offensive

## SUMÁRIO

<b>INTRODUÇÃO</b>	<b>10</b>
Problema	10
Objetivos	12
Objetivo Geral	12
Objetivos Específicos	12
Justificativa	13
<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>14</b>
Skins	14
E-commerce	14
Steam	14
Counter-Strike: Global Offensive	14
Valve Corporation	14
Web Scraping	15
Web Crawling	15
Scrapy	16
Beautiful Soup	16
Web Scraping vs Web Crawling	16
Casos de Teste	18
Google Charts	19
Html	19
Tag	19
Classes	19
Python	20
Angular	20
PostgreSQL	20
<b>METODOLOGIA</b>	<b>21</b>
Estudo sobre o funcionamento do mercado de skins	21
Aprofundamento na técnica de web crawling e web scraping	21
Levantamento de requisitos	21
Estudo sobre as ferramentas e técnicas	21
Implementação dos requisitos	22
Inspeção e teste	22
<b>DESENVOLVIMENTO</b>	<b>23</b>
<b>CONCLUSÃO E CONSIDERAÇÕES FINAIS</b>	<b>35</b>
<b>REFERÊNCIAS</b>	<b>36</b>



## 1. INTRODUÇÃO

Os comparadores de preços trazem alguns benefícios aos seus usuários, “Os comparadores de preço são sites que pesquisam e exibem, em um único lugar, lojas que oferecem o produto que o visitante está buscando. Isso permite que o possível comprador tenha acesso a várias opções de preço para um mesmo produto” (E-GOI, 2018).

Tendo como exemplos os vários comparadores de preços que são usados no Brasil foi proposto o desenvolvimento de uma ferramenta de comparação de preços para prover auxílio para o mercado de skins da comunidade de CSGO, hoje o mercado de skins CSGO ainda não possui um comparador de preços.

Para o desenvolvimento da ferramenta será realizado todo um estudo sobre o mercado de skins e os e-commerce que trabalham com venda de skins no Brasil, também será realizado um estudo sobre as técnicas e as tecnologias para o desenvolvimento da ferramenta.

## 2. Problema

O segmento de games vem crescendo cada vez mais ao longo dos anos, “De acordo com a consultoria do setor Newzoo, o setor ultrapassará os 200 bilhões de dólares em receitas em 2023. Atualmente os jogos eletrônicos vêm crescendo a 11% ao ano, um ritmo muito mais acelerado que o das indústrias de cinema e música juntas” (PURCHIO, 2021).

E o mercado de games do Brasil não fica para trás “Segundo dados da revista Valor, mesmo com a alta do dólar encarecendo os produtos, o faturamento do varejo com a venda de consoles cresceu 34,1% e a venda de computadores específicos para o público gamer aumentou em 75%” (SOUZA, 2020).

Com o crescimento do mercado de games, conseqüentemente, o mercado de skins também cresce exponencialmente, como exemplo: “Na população online urbana da China (cidades de Nível 1 e Nível 2) a grande maioria (94%) dos jogadores pagantes gastou dinheiro em itens de jogos ou bens virtuais nos últimos seis meses, com 55% dos homens e 60% das mulheres comprando *skins*” (NEWZOO, 2018).

Focando um pouco mais no mercado de *skins* do *Counter Strike Global Offensive*, que é um jogo online desenvolvido pela Valve Corporation e pela Hidden Path Entertainment,

possuem números impressionantes: “o CS:GO movimenta um mercado de skins em média de 10 bilhões de dólares por ano” (GLOBE COMMERCE, 2021). O CS:GO consegue movimentar todo esse dinheiro devido sua variedade de *skins* e a grande quantidade de jogadores. “Em 13 de abril de 2020 o CS:GO atingiu seu recorde de jogadores simultâneos que foi de 1.308.963 e diariamente o CS:GO consegue atingir o pico de um milhão de jogadores simultâneos” (STEAM DATABASE, 2021).

Atualmente encontra-se várias opções para compra *skins* de CS:GO, o próprio mercado da Steam, que não é o mais recomendado por apresentar elevado valor de taxa de transações. “A Taxa do Counter-Strike: Global Offensive é uma taxa específica do jogo que é determinada e recolhida pela editora do jogo. A taxa do Counter-Strike: Global Offensive é atualmente de 10%.” (STEAM SUPPORT, 2021). Por consequência dessas taxas criou-se um mercado paralelo de e-commerce de skins, para não terem que pagar essa taxa cobrada pela Steam, um meio de compra com os preços mais acessíveis. Mesmo com a intenção de diminuir o preço das skins, o objetivo central dos e-commerce é o lucro, com um mercado baseado em oferta, encontra-se brechas para produtos com valores abusivos, muitas vezes por falta de ferramentas para a comparação do preço desses produtos digitais. Tendo em vista o crescente mercado de skins e o surgimento de variadas formas de adquirir esses produtos, seria possível diminuir o tempo de busca por esses produtos entre essa variedade de lojas

### **3. Objetivos**

#### **3.1. Objetivo Geral**

Prover auxílio para o mercado de skins da comunidade de CSGO através do desenvolvimento de uma ferramenta para comparação de preços desses produtos digitais.

#### **3.2. Objetivos Específicos**

- Aprofundar conhecimento em algoritmo de Web Crawling;
- Aprofundar conhecimento em algoritmo de Web Scraping;
- Otimizar o tempo de busca por produtos no mercado lojas virtuais de skins;
- Mostrar custo benefícios de itens por meio de histórico de preço;
- Identificar e-commerces para serem adicionados na ferramenta;
- Facilitar acesso a possíveis descontos e promoções.

#### 4. Justificativa

*Steam* já oferece um mercado de comunidade bastante vasto, com milhares de opções de *skins* para serem compradas. A plataforma também é abrangente nos preços, que podem variar de pouquíssimos centavos a mais de R\$7 mil. Essas quantias altas não podem ser retiradas de sua conta do *Steam*, e devem ser gastas dentro da própria plataforma com outros itens ou jogos. Para solucionar esse impasse, começaram a surgir lojas alternativas, que compram itens dos jogadores e os revendem online por um preço abaixo do encontrado na loja da Valve, transformando aquele montante em dinheiro real, em um novo negócio (GLOBE ECOMMERCE, 2019).

Comparar preços é essencial antes de fazer uma compra, pois ajuda os consumidores a encontrar boas ofertas no mesmo item ou encontrar itens semelhantes a preços com desconto. A pesquisa mostra que cerca de 59% dos clientes preferem comparar preços online em todas as categorias antes de fazer uma compra (Business Case Studies, 2020).

Tendo como exemplo os diversos aplicativos que comparam os preços de produtos analisando as lojas virtuais ou e-commerce, buscamos trazer uma aplicação com este foco para o mercado de skins de Counter Strike Global Offensive, sendo uma ajuda para vários usuários na hora de realizar suas compras de skins, desenvolvendo uma ferramenta que terá um comparador de preços para as skins de CS:GO. Ao desenvolver essa ferramenta os usuários vão ter alguns benefícios como otimizar o tempo de busca por produtos, análise de custo benefício dos produtos e um padrão justo de preços fornecidos pelo mercado.

## **5. FUNDAMENTAÇÃO TEÓRICA**

### **5.1.Skins**

O termo Skin vem do inglês e se traduz como peles, no contexto dos jogos ele é inserido como o visual e estilos dos itens ou bonecos presentes nos jogos. O Site Parentzone define Skin como: “Uma skin é um download gráfico ou de áudio que altera a aparência dos personagens nos videogames. Eles são puramente estéticos - eles não aumentam as habilidades do personagem ou afetam o resultado do jogo” (PARENTZONE, s.d).

### **5.2.E-commerce**

E-commerce é a abreviação em inglês de comércio eletrônico, ou seja, toda transação comercial (compra e venda) feita através da internet com o auxílio de um equipamento eletrônico (CLIMBA COMMERCE, 2021).

### **5.3.Steam**

O Steam foi lançado em 2003 pela produtora Valve, como um serviço para vender jogos digitais, dos mais variados – basicamente, uma loja online. Ao longo dos anos, evoluiu e se tornou muito mais, sendo hoje uma plataforma completa, com chat entre amigos, coleções, lista de desejos, eventos e promoções temáticas, entre outros incrementos que recebeu (VINHA, 2019).

### **5.4.Counter-Strike: Global Offensive**

Counter Strike: Global Offensive (CS:GO) é um reboot do clássico Counter Strike de 1999, mas com data de lançamento em 2012. O game está disponível para download grátis no PC (Steam), Playstation 3 (PS3) e Xbox 360, e traz tudo o que seu antecessor tinha, mas agora com a adição de vários elementos mais modernos (VELLOSO, 2020).

### **5.5.Valve Corporation**

A Valve Corporation é uma empresa norte-americana que produz e publica jogos eletrônicos. Embora tenha feito fama pela série Half Life, a Valve é conhecida pelo software Steam, programa usado para dificultar a falsificação e venda de jogos usados e que atualmente funciona como um administrador de games (PUREBREAK, 2014).

## 5.6. Web Scraping

O Web Scraping se define como uma técnica utilizada para a extração de dados, com o objetivo de ser utilizada em um site ou análise para tomada de decisões. Normalmente, esses dados são extraídos em formatos de arquivos.

Teoricamente, web scraping é a prática de coletar dados por qualquer meio que não seja um programa interagindo com uma API (ou, obviamente, por um ser humano usando um navegador web). Isso é comumente feito escrevendo um programa automatizado que consulta um servidor web, requisita dados (em geral, na forma de HTML e de outros arquivos que compõem as páginas web) e então faz parse desses dados para extrair as informações necessárias (MITCHELL, 2015).

## 5.7. Web Crawling

O Web Crawling é uma técnica de indexação de conteúdo de sites que pode percorrer a web de forma automatizada, sendo utilizada por vários buscadores famosos como o Google e o Bing. O autor Ryan Mitchell descreve Web Crawling no seu livro “Web Scraping com Python” da seguinte maneira:

Os web crawlers (rastreadores da web) recebem esse nome porque rastreiam (crawl) a web. Em seu núcleo, encontra-se um elemento de recursão. Eles devem obter o conteúdo da página de um URL, analisar essa página em busca de outro URL e obter essa página, ad infinitum (MITCHELL, 2015).

As ferramentas existentes não possuem a mesma capacidade disponibilizada pelos algoritmos web crawling, que por sua vez é utilizado na grande maioria das pesquisas e projetos que envolvem indexação na *web*. Segue um exemplo de utilização no cenário acadêmico:

Entre as diferenças do Web Crawler em relação às demais ferramentas existentes está sua capacidade de poder coletar e realizar testes de forma automática em todos os sites existentes na Internet. Para isso, o Web Crawler recebe como entrada um determinado site e, ao encontrar links externos no site de entrada, cria uma lista de novas varreduras, que é executada em sequência após o término do processo de análise para cada site encontrado através desses links. Além disso, outro diferencial são as projeções dos resultados das análises, que são encaminhados para os possíveis responsáveis de cada site (MACHADO, 2015).

## 5.8.Scrapy

Por ser um Framework, o Scrapy disponibiliza diversas funcionalidades que facilitam o processo de crawler. Desde o controle de navegação na web, bibliotecas de parse em HTML, representação de dados e pipelines para filtragem e tratamento de dados (SANTANA, 2017).

Scrapy é uma estrutura de aplicativo para rastrear sites da web e extrair dados estruturados que podem ser usadas para uma ampla gama de aplicativos úteis, como mineração de dados, processamento de informações ou arquivamento de histórico. Embora o Scrapy tenha sido originalmente projetado para web scraping , ele também pode ser usado para extrair dados usando APIs (como Amazon Associates Web Services ) ou como um rastreador da web de propósito geral (SCRAPY DEVELOPERS, 2021).

## 5.9.Beautiful Soup

Beautiful Soup 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 (CRUMMY, s.d).

É uma ferramenta incrível para arrancar informações de uma página web. Você pode usá-lo para extrair tabelas, listas, parágrafos e você também pode colocar filtros para extrair informações de páginas da web (VOOO – INSIGHTS, 2016).

## 5.10. Web Scraping vs Web Crawling

O Web Scraping é o ato de “raspar” dados da internet, ou seja, baixar um conteúdo, podendo ser feito de forma manual. Já o Web Crawling tem um conceito completamente diferente, ele apenas “rastrea” os dados pela internet, sendo feito por um bot ou agente rastreamento de forma automatizada. Sendo o Crawling geralmente acompanhado pelo Scraping para fazer a filtragem das informações desnecessárias.

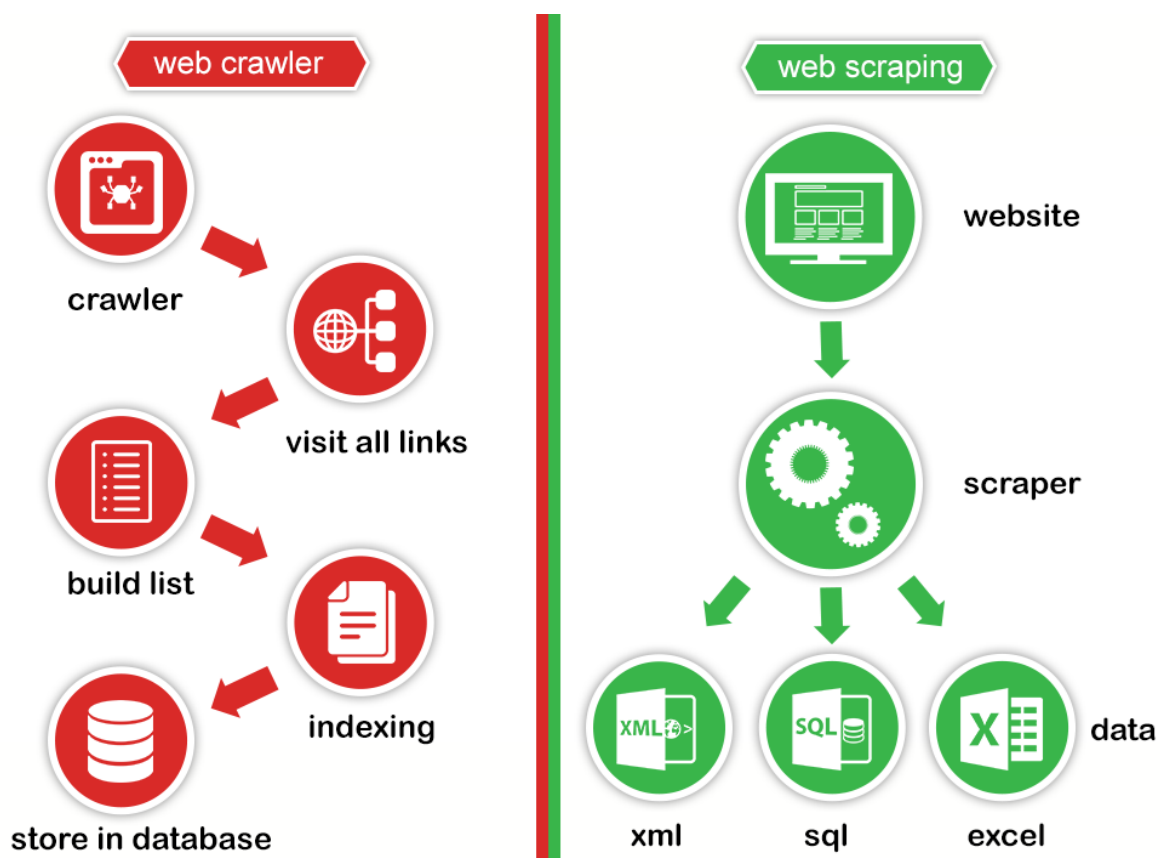
É importante compreender as principais diferenças entre o web crawling e o web scraping, mas também, na maioria dos casos, o web scraping anda de mãos dadas com o web crawling. Ao fazer o web scraping, você baixa informações prontamente disponíveis online. O

web crawling é usado para extração de dados de mecanismos de pesquisa e sites de comércio eletrônico e, posteriormente, você filtra as informações desnecessárias e seleciona apenas aquelas de que precisa, raspando-as (Fatenaite, 2021).

O conceito de Web Scraping e Web Crawling pode ser às vezes confundido, mesmo tendo objetivos diferentes, pois na maioria das vezes esses dois processos andam de mãos dadas, no entanto, apresentam uma diferença no funcionamento como apresentado na Ilustração 1.



**Ilustração 1 - Web Crawler vs Web Scraping**



Fonte: (ProWebScraping)

Conforme apresentado na Ilustração acima, o Web Crawling é um processo de localização das informações na World Wide Web (WWW), visitando todos os hiperlinks relacionados, listando-os e em seguida indexando todas as palavras em um documento e em seguida adicionando em um banco de dados. Já o Web Scraping como apresentado, é o processo de solicitar automaticamente um documento da web e coletar informações dele.

### 5.11. Casos de Teste

Um Caso de Teste mostra os caminhos percorridos por um módulo, Caso de Uso ou funcionalidade dentro do projeto. Serve como base para que os testadores possam executar os testes manualmente, mas pode ser criado, também, com o intuito de automatizar os testes. Além disso, os Casos de Teste devem cobrir o máximo de situações possíveis. Resumindo, um Caso de Teste é um conjunto de ações e os resultados esperados para elas (ENDLER, 2018).

### **5.12. Google Charts**

O Google Charts oferece uma maneira perfeita de visualizar dados em seu site. De gráficos de linhas simples a mapas de árvore hierárquica complexos, a galeria de gráficos oferece um grande número de tipos de gráficos prontos para uso. A maneira mais comum de usar o Google Charts é com JavaScript simples que você incorpora em sua página da web. Você carrega algumas bibliotecas do Google Chart, lista os dados a serem mapeados, seleciona opções para personalizar seu gráfico e, finalmente, cria um objeto de gráfico com um id de sua escolha (GOOGLE DEVELOPERS, 2019).

### **5.13. Html**

Criada pelo britânico Tim Berners-Lee, o acrônimo HTML significa HiperText Markup Language, traduzindo ao português: Linguagem de Marcação de Hipertexto. O HTML é o componente básico da web, ele permite inserir o conteúdo e estabelecer a estrutura básica de um website. Portanto, ele serve para dar significado e organizar as informações de uma página na web. Sem isso, o navegador não saberia exibir textos como elementos ou carregar imagens e outros conteúdos (HOME HOST, s.d).

### **5.14. Tag**

Cada página HTML consiste em uma série de tags (também chamadas de elementos), que podem ser consideradas os blocos de construção das páginas. Elas criam uma hierarquia que estrutura o conteúdo entre seções, parágrafos, cabeçalhos e outros blocos de conteúdo (HOTTINGER TUTORIAIS, 2021).

### **5.15. Classes**

Classes proporcionam uma forma de organizar dados e funcionalidades juntos. Criar uma nova classe cria um novo “tipo” de objeto, permitindo que novas “instâncias” desse tipo sejam produzidas. Cada instância da classe pode ter atributos anexados a ela, para manter seu

estado. Instâncias da classe também podem ter métodos (definidos pela classe) para modificar seu estado (DOCUMENTAÇÃO PYTHON, 2021).

### **5.16. Python**

Python é uma excelente linguagem de programação orientada a objetos, interpretada e interativa. Python combina poder notável com sintaxe muito clara. Possui módulos, classes, exceções, tipos de dados dinâmicos de nível muito alto e tipagem dinâmica. Existem interfaces para muitas chamadas de sistema e bibliotecas, bem como para vários sistemas de janelas. Novos módulos integrados são facilmente escritos em C ou C ++ (ou outras linguagens, dependendo da implementação escolhida). Python também pode ser usado como uma linguagem de extensão para aplicativos escritos em outras linguagens que precisam ser fáceis de usar scripts ou interfaces de automação (PYTHON WIKI, 2018).

### **5.17. Angular**

Angular é uma plataforma de desenvolvimento, construída em TypeScript. Como plataforma, o Angular inclui: uma estrutura baseada em componentes para a construção de aplicativos da web escaláveis; uma coleção de bibliotecas bem integradas que cobrem uma ampla variedade de recursos, incluindo roteamento, gerenciamento de formulários, comunicação cliente-servidor e muito mais; um conjunto de ferramentas de desenvolvedor para ajudá-lo a desenvolver, construir, testar e atualizar seu código (ANGULAR, 2021).

### **5.18. PostgreSQL**

O PostgreSQL é um SGBD (Sistema Gerenciador de Banco de Dados) objeto-relacional de código aberto, com mais de 15 anos de desenvolvimento. É extremamente robusto e confiável, além de ser extremamente flexível e rico em recursos. Ele é considerado objeto-relacional por implementar, além das características de um SGBD relacional, algumas características de orientação a objetos, como herança e tipos personalizados (PORTAL EDUCAÇÃO s,d).

## **6. METODOLOGIA**

### **6.1. Estudo sobre o funcionamento do mercado de skins**

Tendo como objetivo levantar os principais pontos do mercado, analisando a estrutura e todo o funcionamento dos sites que vendem skins através de sites web, por meio de pesquisas nos sites das lojas será levantado uma estrutura do modelo de negócio, focando nos seguintes pontos: estilos de usuários, gráficos do mercado de acordo com a precificação da *Skins* e Sites de comércio.

### **6.2. Aprofundamento na técnica de web crawling e web scraping**

Os algoritmos de web crawling são amplamente utilizados para indexar sites, sendo uma ferramenta para capturar informações em links de sites juntamente com o web scraping. Será dedicado um tempo na aprendizagem de toda a sua forma de funcionamento e implementação, seguindo os passos do livro “Web Scraping com Python: Coletando Mais Dados da Web Moderna” (MITCHELL, 2015). O estudo dessas técnicas será fundamental para a construção de toda a aplicação, pois se baseiam em dados de lojas virtuais de terceiros e com essas técnicas podemos extrair e manipular os dados relacionados aos produtos.

### **6.3. Levantamento de requisitos**

Com todos os pacotes de funcionalidades será elaborado as telas de mockup, e toda a documentação aprofundada que envolva essas telas, as regras de negócios e os critérios de aceitação. Oferecendo maior facilidade e objetividade a implementação desses requisitos.

### **6.4. Estudo sobre as ferramentas e técnicas**

O estudo sobre as ferramentas envolverá todo conteúdo que ajudará a desenvolver essa aplicação web. Sendo dividido em etapas, primeiro o Script de coleta de dados desenvolvido em Python, como banco o postgresSQL, o back-end em Node.js e o front-end com Angular na versão 9.

## **6.5.Implementação dos requisitos**

O desenvolvimento das aplicações é o objetivo final desse trabalho, e uma equipe pequena só pode entregar todos os seus objetivos no final do ciclo de vida do projeto. Por isso, nas primeiras semanas dedicadas ao trabalho, o tempo será dedicado à documentação e ao teste de inspeção desses requisitos, por meio de um processo incremental.

## **6.6.Inspeção e teste**

Nesta etapa, será realizada uma inspeção em toda a documentação produzida a fim de encontrar erros ou inconsistência na documentação, em seguida será produzida uma tabela de casos de teste de acordo com os requisitos que será realizado mais um tipo de teste, que é o teste de inspeção onde o testador sem ter acesso ao código fonte irá testar o'que foi documentado nos casos de teste a fim de verificar se está tudo de acordo com o que foi documentado nos requisitos.

## 7. DESENVOLVIMENTO

Na inicialização do trabalho foi realizada uma pesquisa sobre o mercado *skins* para identificar os e-commerce que trabalham com venda de skins dentro do Brasil e assim levantarmos os principais e-commerce brasileiros, nesta pesquisa encontramos uma enorme variedade de e-commerce. Dessa quantidade de e-commerces foi identificado que trabalham com um estoque bem menor e com mais rotação que o mercado da Steam. Também foi realizada uma pesquisa para identificar se já existia uma ferramenta parecida com a que o trabalho propõe. Depois dessas pesquisas foi encontrado o CS:GO Stash, um site que lista os preços das *skins* utilizando os itens anunciados na Steam e em uma loja paralela, a BitSkins.

O CS:GO Stash funciona da seguinte maneira: ao selecionar o produto desejado será listado o preço dos mesmos, sendo dividido em duas partes, a primeira lista apresenta o preço das *skins* StatTrack e todos os seus níveis de desgaste, já a segunda lista apresenta o preço das *skins* normais e todos os seus níveis de desgaste, também possível selecionar se esses preços listados estão vindo de itens anunciados na steam ou da loja BitSkins, sendo esse a única loja de terceiros que está disponível no CS:GO Stash, assim como é mostrado na ilustração 2.

Ilustração 2 - Tela de comparação de preço CS:GO Stash

**AK-47 | Leet Museo**

Covert Rifle

StatTrak Available

Inspect (FN)   Inspect (MW)   Inspect (FT)   Inspect (WW)   Inspect (BS)

Operation Riptide Case   The Operation Riptide Collection

210 Steam Market Listings	
StatTrak Factory New	R\$ 903,33
StatTrak Minimal Wear	R\$ 496,57
StatTrak Field-Tested	R\$ 299,52
StatTrak Well-Worn	R\$ 422,20
StatTrak Battle-Scarred	R\$ 290,03
Prices updated 12 seconds ago	

0,00      0,65

FN   MW   FT   WW   BS

**Description:** It has been custom painted with a Cubism depiction of Elite Crew Ts sneaking up A-long on Dust II.

**Flavor Text:** "Everything you can imagine is real"

**Finish Style:** Gunsmith

**Finish Catalog:** 1087

**Added:** 22 September 2021

**Creator:** Workshop Submission

**Update:** Operation Riptide

Fonte: CS:GO Stash

Após os resultados das pesquisas viabilizou-se a proposta desse trabalho, pois apesar do mercado já possuir uma ferramenta de amostra de preços, ela não tem o foco de fazer a comparação do preço do mercado de ecommerce. Identifica-se que o objetivo desta ferramenta é apenas listar os preços dos itens anunciados na steam e em uma loja paralela, a BitSkins, sendo assim o mercado de skins ainda sofre uma carência neste ponto.

Os estudos das técnicas aplicadas neste trabalho foram baseados no livro “Web Scraping com Python: coletando mais dados na web moderna” e com sua ajuda chegamos a algumas conclusões sobre as melhores práticas para o desenvolvimento do nosso projeto. Mesmo que o web scraping possa ser escrito de forma completamente manual utilizando Python, não é necessariamente a forma mais produtiva para fazê-lo, tendo em vista os frameworks Scrapy e o BeautifulSoup. Mesmo sendo importante conhecer os padrões básicos

para escrever do zero o código, a biblioteca Beautiful Soup cuida de muitos detalhes, tornando a codificação menos repetitiva.

Através de uma busca por e-commerce encontramos uma variedade gigantesca no mercado. Para a nossa pesquisa baseamos nossos dados em três e-commerces brasileiros, sendo essas referências no mercado: Arsenal Skins, Batskins e CSGOStore.

Durante os estudos, também foram levantados os principais requisitos do sistema. A Tabela 1 e 2, apresenta os Requisitos Funcionais. Todos os requisitos levantados até o momento seguem o padrão apresentado.

**Tabela 1 - Requisito Funcional 001**

<b>RF001</b>	Pesquisa de produtos		
<b>Descrição</b>	Possibilita que um usuário possa buscar por skins específicas com palavras chaves dentro do catálogo de lojas.		
<b>Descrição do risco</b>		<b>Risco</b>	<b>Prioridade</b>
		Nenhum	Essencial

Fonte: Os Autores

**Tabela 2 - Requisito Funcional 002**

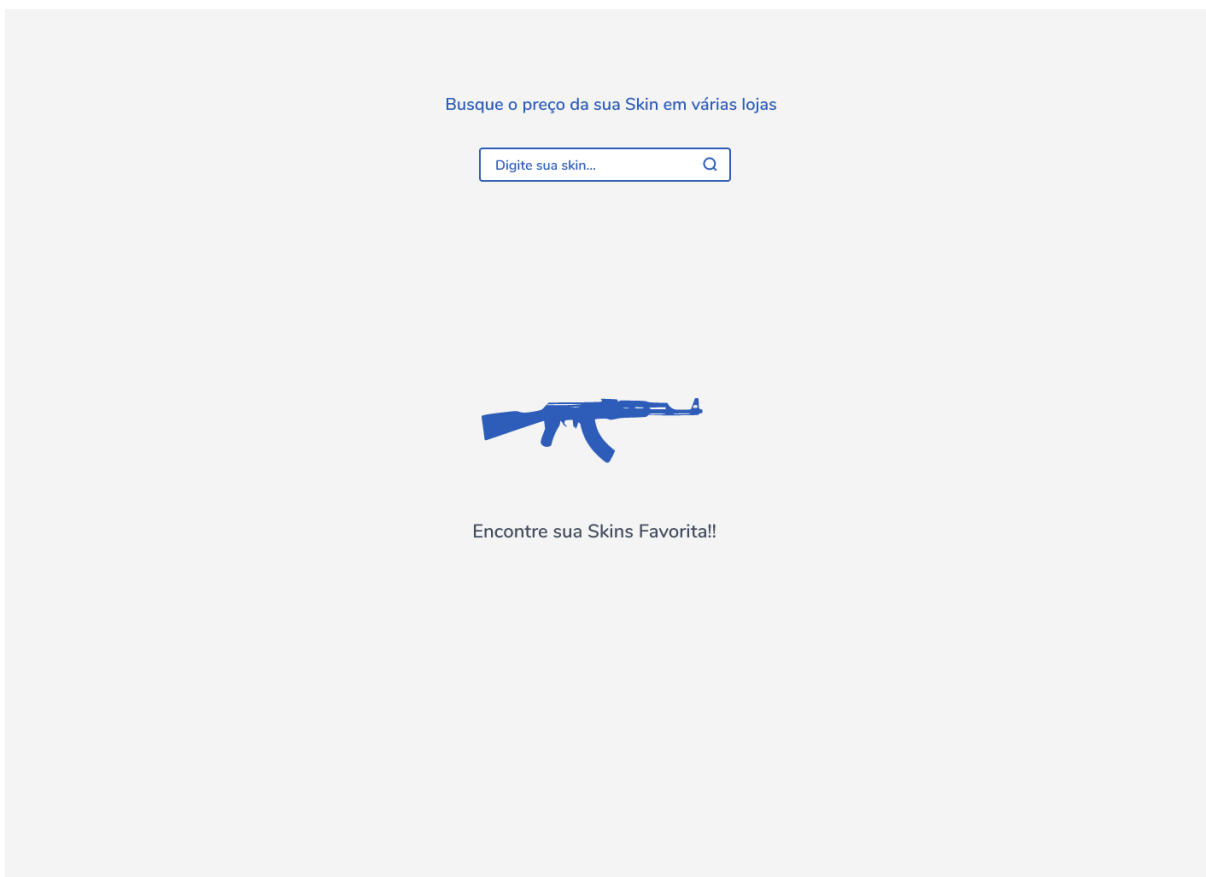
<b>RF002</b>	Filtragem de produtos		
<b>Descrição</b>	Possibilita que um usuário filtre a skin de acordo com sua escolha, podendo ordenar de forma crescente e decrescente		
<b>Descrição do risco</b>		<b>Risco</b>	<b>Prioridade</b>
		Nenhum	Importante

Fonte: Os Autores



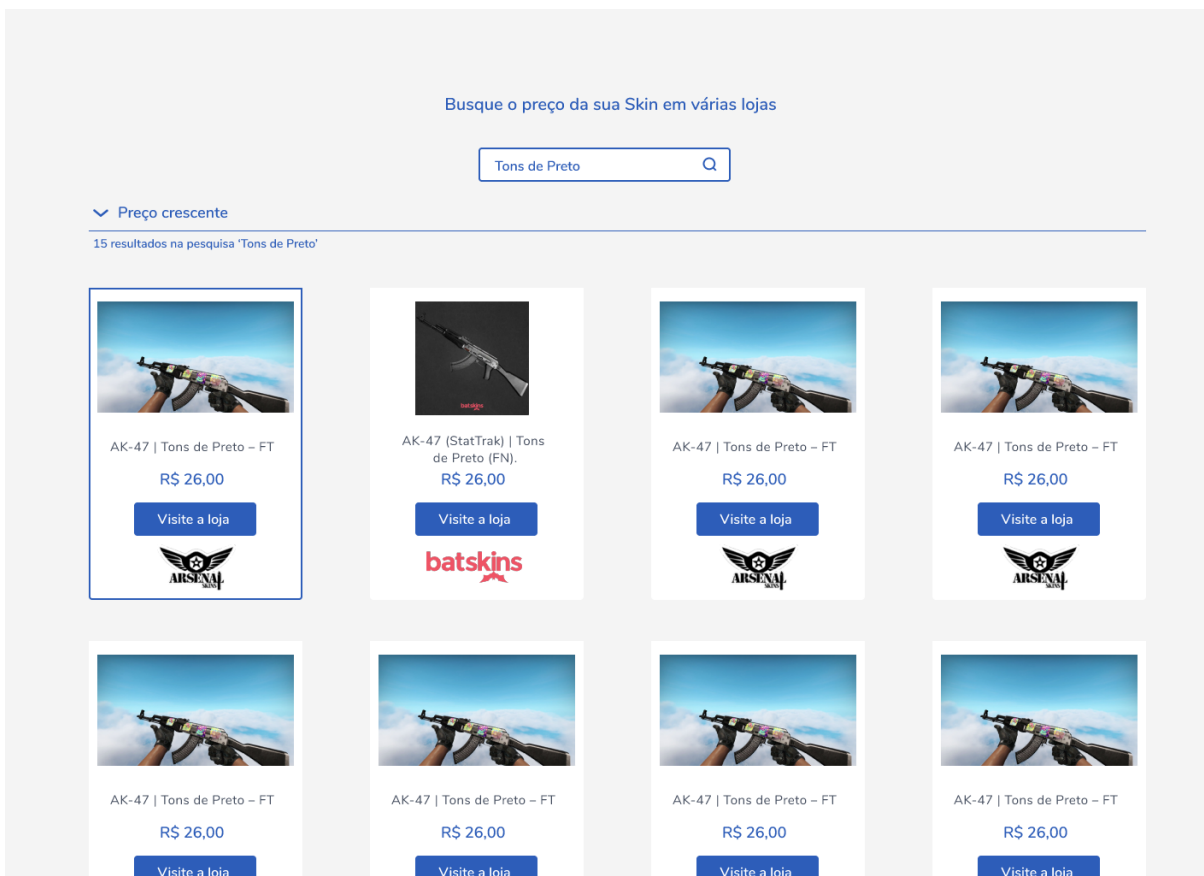
Antes do requisitos serem desenvolvidos foram desenvolvidos alguns protótipos de telas para deixar mais claro como seria feito todo o design da aplicação web, seguem alguns exemplos nas ilustrações 3 e 4.

### Ilustração 3 - Protótipo da tela de pesquisa



Fonte: Os Autores

## Ilustração 4 - Protótipo da tela de listagem de produtos



Fonte: Os Autores

Durante o desenvolvimento das funcionalidades do sistema foi desenvolvido uma tabela de caso de teste, a tabela de caso de testes e onde será documentado o passo a passo que o testador deverá percorrer durante a realização dos testes, observando a Tabela 3 é possível identificar qual funcionalidade será testada o seu título o passo a passo do teste e o seu resultado esperado e o status onde é mostrado qual foi o resultado do teste realizado.

**Tabela 3** - Casos de teste

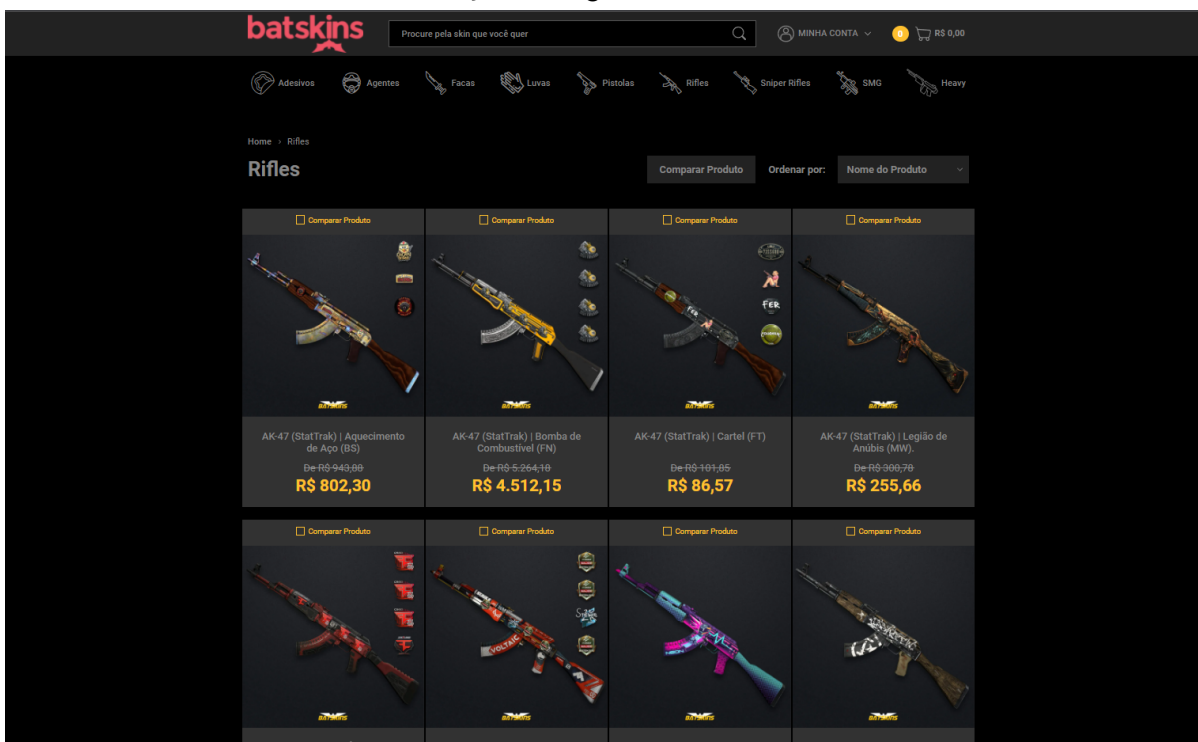
ID	Funcionalidade/Componente	Título	Passo-a-passo para execução	Resultado esperado	Status
1	Pesquisar produtos	Pesquisa de produtos	1- Acessar o site; 2- Digite na barra de pesquisa o produto que está procurando	1- Verifique se a pesquisa foi retornada corretamente.	Ok
2	Listar produtos	Listagem de produtos	1- Acessar o site; 2- Digite um produto na barra de pesquisa.	1- Verificar se foi retornada uma lista com o produto que foi pesquisado.	Ok
3	Filtrar produtos	Filtragem de produtos	1- Acessar o site; 2- Aplicar um filtro nos produtos.	1- Verificar se a filtragem está ocorrendo de maneira correta.	Ok
4	Descrição do produto	Detalhes do produto	1- Acessar o site; 2- Selecionar um produto.	1- Verificar se foi retornado a descrição do produto selecionado e o seu histórico de preço.	Ok
5	Exibir os preços do produto	Exibir comparação entre os preços dos produtos	1- Acessar o site; 2- Selecionar um produto.	1- Verificar se foi retornada uma lista com os preços e as lojas que vendem esse produto.	Ok

Fonte: Os autore

Os testes foram realizados de forma local, onde projeto é executado na máquina do testador para ser testado todas as funcionalidades que foram desenvolvida para identificar se está tudo de acordo com o que foi documentado no caso de teste, o teste foi feito de forma manual com o testador navegando pelo o projeto é realizado os testes, como o histórico de de preço não foi desenvolvido não foi possível realizar o teste nessa funcionalidade, mas as demais funcionalidade foram testadas.

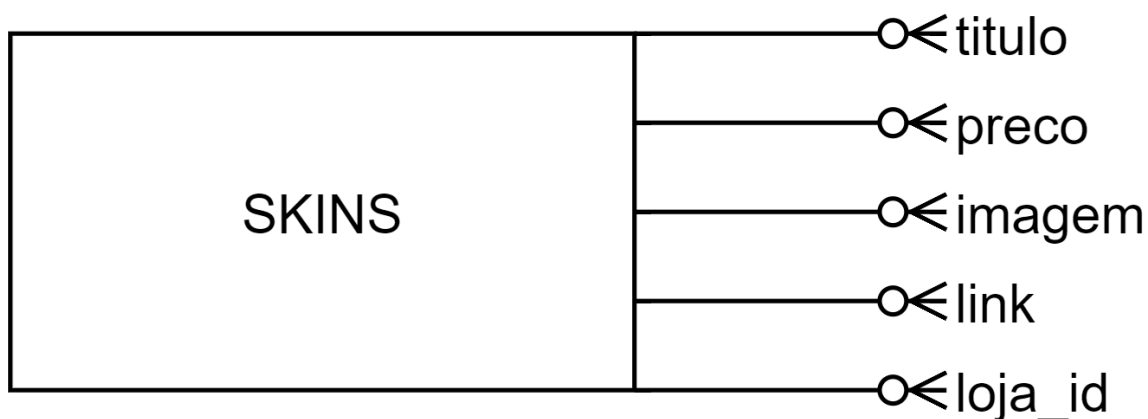
O padrão estrutural das lojas pesquisadas, como mostra a ilustração 5, sempre é uma listagem de itens, no caso skins, que apresentam, uma imagem do produto, nome e preço. Para fazer a varredura do HTML da página e armazenar os dados, primeiramente foi definido quais seriam estes dados necessários para a composição do banco de dados. Como visto na ilustração 6 a estrutura do banco é muito simples, apenas o título, o preço, a imagem do produto e o link que redireciona para a página do mesmo.

Ilustração 5 - Página inicial BatSkins



Fonte: BatSkins

Ilustração 6 - Modelo entidade relacional



Fonte: Os autores

Com todos os dados definidos foi criado um Web Scraping que coleta toda essa informação de acordo com a listagem. O Scraping em questão foi desenvolvido em Python, juntamente com duas bibliotecas, o pandas e o BeautifulSoup, que são importados como apresenta a ilustração 7. O pandas é utilizado para fornecer estruturas e operações para manipular tabelas, já o BeautifulSoup é a grande figura por trás do Web Scraping, pois ele fornece um pacote Python para análise de documentos HTML. A ilustração 8 apresenta a estrutura HTML da listagem apresentada na ilustração 5 do e-commerce BatSkins, ali encontra-se o padrão para ser buscado pelo BeautifulSoup, que se baseia em tags e classes do HTML para encontrar os dados.

**Ilustração 7 - Importação das bibliotecas**

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

Fonte: Os autores

**Ilustração 8 - Estrutura HTML da Batskins**

```
<ul class="list flex f-wrap row"> flex
  <li class="item flex"> flex
    <div class="product nb show-down"> flex
      <div class="compare-buttons">...</div>
      <div class="image">
        <a href="https://www.batskins.com.br/rifles/ak-47/ak-47-stattrak-aquecimento-de-aco-bs" class="space-image">...</a>
      </div>
      <a href="https://www.batskins.com.br/rifles/ak-47/ak-47-stattrak-aquecimento-de-aco-bs" class="info-product"> flex
        <div class="cronometro-data clock" data-end="2021-12-31">...</div> flex
        <div class="product-name">AK-47 (StatTrak) | Aquecimento de Aço (BS)
        </div>
        <div class="down-line">
          <div class="box-price"> flex
            <div class="price">
              <div class="product-price other_price"> flex
                <div class="price-before ">...</div>
                <div class="price-off">
                  " R$ 802,30 "
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
```

Fonte: Os autores

Tendo a estrutura da Batskins em vista, levantou-se um documento contendo todas as tags e classes que seriam necessárias para o BeautifulSoup localizar os dados no texto HTML, apresentado na ilustração 9. Um exemplo é a tag `div` que contém a classe “price-off”, com ela conseguimos acessar o valor do produto.

### Ilustração 9 - Tags e Classes dos produtos da loja Batskins

```
1 BATSSKINS:
2   ID_LOJA: 2
3   URL: https://www.batskins.com.br/loja/catalogo.php?loja=741009&categoria=11&pg={i}
4
5   TAG PRINCIPAL: li
6   CLASSE PRINCIPAL: item flex
7
8   TAG TITULO: div
9   CLASSE TITULO: product-name
10
11  ELEMENTO DA IMAGEM: data-src
12
13  TAG PREÇO: div
14  CLASSE PREÇO: price-off
15
16  CLASSE LINK: woocommerce-LoopProduct-link woocommerce-loop-product__link
```

Fonte: Os autores

Para passar esses dados para o Scraping a forma mais inteligente seria através de uma função, pois seria reutilizado para todas as lojas. Sendo assim foi criado essa função que pede todas as informações listadas na ilustração 8. Essa função apresentada na ilustração 10 engloba dois *loopings*, sendo o primeiro feito para percorrer toda a paginação da lista de produtos e o segundo para percorrer todos os produtos da lista. No final do Script foi utilizado o pandas que exporta todos os dados coletados na página para um arquivo CSV com a formatação perfeita para ser inserido no banco de dados.

### Ilustração 10 - Função Scraping

```

def webscraping(url, tag_principal, class_principal, tag_titulo,
class_titulo, tag_preco, class_preco, elemento_img, num_paginacao, loja_id ):
for i in range(1, int(num_paginacao)):
url_page = f'{url}'
site = requests.get(url_page)
soup = BeautifulSoup(site.content, 'html.parser')
skins = soup.find_all(f'{tag_principal}', class_= f'{class_principal}' )

for skin in skins:
titulo = skin.find(f'{tag_titulo}', class_= f'{class_titulo}').get_text().strip()

img = skin.find('img')
img_src = img[elemento_img]

link = skin.find('a')
link_url = link['href']

try:
preco = skin.find(f'{tag_preco}', class_= f'{class_preco}').get_text().strip()
num_preco = preco[2:]

except:
num_preco = "0"

lista_skins.append([titulo, num_preco, img_src, link_url, loja_id])

linha = titulo + ";" + num_preco + ";" + img_src + ";" + "\n"
print(linha)

news = pd.DataFrame(lista_skins, columns=['titulo', 'preco', 'imagem', 'link', 'loja_id'])

```

Fonte: Os autores

A ativação do Scraping é acionada apenas com a execução do arquivo Python, após executá-lo basta passar os dados que são solicitados no terminal que o BeautifulSoup junto com o pandas faz todo o trabalho automático definido na ilustração 9. Para passar os dados basta digitá-los no terminal como apresenta a ilustração 11.

### Ilustração 11 - Input do para buscas de dados

```

----- CASO NÃO TENHA O QUE FOI PEDIDO DIGITE: "vazio" -----
Digite o id da loja: 2
Digite a url da página: https://www.batskins.com.br/loja/catalogo.php?loja=741009&categoria=11&pg={i}
Digite a tag Principal: li
Digite a classe Principal: item flex
Digite a Tag do título: div
Digite a Classe do título: product-name
Digite o Elemento da imagem: dat-src
Digite a Tag do preço: div
Digite a Classe do preço: price-off
Digite a Classe do link: woocommerce-LoopProduct-link woocommerce-loop-product__link

```

Fonte: Os autores

Com toda a coleta de dados feita foi executado um *script* SQL, como a Ilustração 12 apresenta, que cria a tabela Skins e faz o preenchimento dela com os dados coletados a partir do caminho do CSV. O banco criado em PostgreSQL foi preenchido com toda a informação coletada nos três e-commerces selecionados para a pesquisa.

### Ilustração 12 - Script SQL para criação e preenchimento da tabela

```

create table
  skins(titulo varchar(255),
        preco decimal(20,2),
        imagem varchar(999),
        link varchar(999),
        loja_id int)

copy skins(titulo, preco, imagem, link, loja_id)
from 'C:\Users\pedro\Desktop\skins\skins-arsenal.csv'
with delimiter as ',' CSV HEADER

```

Fonte:Os autores

Para disponibilizar a comparação de preço foi estruturado um front-end desenvolvido com Angular 9, que lista todas as skins de forma ordenada de acordo com a pesquisa do usuário. Apesar de simplista, ele foi desenvolvido para atingir os objetivos do projeto. Feita uma busca simples na aplicação pode notar variação de preço do mesmo produto de acordo com as lojas como é apresentado na Ilustração 13.













### Ilustração 13 - Tela de comparação de preços

Busque o preço da sua Skin em várias lojas

Patrocínios

Preço crescente

6 resultados na pesquisa 'Patrocínios'

 <p>AK-47   Patrocínios (WW)</p> <p>R\$ 202,30</p> <p>Visite a loja</p> 	 <p>AK-47   Patrocínios (FN)</p> <p>R\$ 301,84</p> <p>Visite a loja</p> 	 <p>AK-47 (StatTrak)   Patrocínios (WW)</p> <p>R\$ 515,98</p> <p>Visite a loja</p> 	 <p>AK-47 (StatTrak)   Bloodsport   Patrocínios (WW)</p> <p>R\$ 515,98</p> <p>Visite a loja</p> 
 <p>AK-47 (StatTrak)   Patrocínios (WW)</p> <p>R\$ 515,98</p>	 <p>AK-47 (StatTrak)   Patrocínios (MW)</p> <p>R\$ 575,00</p>		

Fonte: Os autores

## 8. CONCLUSÃO E CONSIDERAÇÕES FINAIS

A realização do projeto foi de grande auxílio para o crescimento pessoal e profissional dos autores, possibilitando o conhecimento de novos conceitos e aplicando algumas teorias que foram adquiridas durante a graduação, para conseguir alcançar o objetivo proposto de desenvolver uma ferramenta para comparação de preços de produtos digitais foi desenvolvido uma pesquisa de mercado e uma pesquisa de ferramentas semelhantes.

Os resultados apurados nas pesquisas serviram para viabilizar o desenvolvimento da ferramenta proposta, assim o foco foi voltado para os estudos das técnicas e ferramentas que foram utilizadas para o desenvolvimento do projeto e também para o desenvolvimento da documentação do projeto.

A aplicação desenvolvida consegue ser um caminho viável para o problema da variação de preços entre as variedades de lojas online, disponibilizando um catálogo de preços para os clientes fazerem a busca e comparação entre esses e-commerces.

Dentre todos os objetivos propostos pelo trabalho, quase todos foram alcançados. Tendo agora uma ferramenta que provê o auxílio para o mercado de skins, fazendo uma comparação de preço, e conseqüentemente otimizando o tempo de busca.

Para futuros trabalhos seria interessante a aplicação do Web Crawling no script python, fazendo uma melhoria no rastreamento de produtos automatizados e aplicação de gráfico com o histórico de preço dos produtos utilizando o google Charts, que estavam entre alguns objetivos que não puderam ser alcançados com o tempo de desenvolvimento desse projeto. Acrescentar outros estilos de jogos, que também possuem skins e um mercado movimentado, a nossa ferramenta também seria de grande valia para trabalhos futuros.

## REFERÊNCIAS

AGOSTINI, Marciel. **Como colocar Stattrak em armas no CS:GO? Veja como skins funcionam**, TechTudo, 12 de mai. 2021. Disponível em:

<<https://www.techtudo.com.br/dicas-e-tutoriais/2021/05/como-colocar-stattrak-em-armas-no-csgo-veja-como-skins-funcionam-esports.ghtml>> Acesso em 13 de mai. de 2021.

Business Case Studies. **Why you always need to compare prices before shopping**

Disponível em:

<<https://businesscasestudies.co.uk/why-you-always-need-to-compare-prices-before-shopping/#:~:text=Comparing%20prices%20is%20essential%20before,similar%20items%20at%20discounted%20prices.&text=This%20means%20any%20product%20you,and%20when%20you%20purchase%20it>>. Acesso em 10 de mai. de 2021.

Clima Commerce. **O que é e-commerce?** Disponível em:

<<https://www.climba.com.br/blog/o-que-e-e-commerce-loja-virtual/>>. Acesso em: 04 de abr. 2021.

Crummy, **Documentação Beautiful Soup** Disponível em:

<<https://www.crummy.com/software/BeautifulSoup/bs4/doc.ptbr/#>>. Acesso em 18 de nov. de 2021.

Documentação Python. **Classes** Disponível em:

<<https://docs.python.org/pt-br/3/tutorial/classes.html>>. Acesso em 30 de out de 2021.

E-goí, **Aprenda a usar o comparador de preços a favor de seu negócio**. Disponível em:

<<https://blog.e-goí.com.br/comparador-de-precos>>. Acesso em 18 de dez de 2021.

ENDLER, Daniele. **Dicas para escrita de casos de teste**, Medium, 04 de jan. de 2018.

Disponível em:

<<https://medium.com/@daniele.e.s/dicas-para-escrita-de-casos-de-teste-ccea14a7fdd9>>.

Acesso em 24 de out. de 2021.

FATENAITE, Gabija. **Web Scraping vs Web Crawling: The Differences**, Oxylabs, 04 de

mai. de 2021. Disponível em: <<https://oxylabs.io/blog/crawling-vs-scraping>>. Acesso em 11 de mai. de 2021.

Google Developers. **Google Charts**. Disponível em:

<<https://developers.google.com/chart/interactive/docs>>. Acesso em 30 de out. de 2021.

GLOBE ECOMMERCE. **O bilionário mercado de skins de CS:GO e como ganhar dinheiro com isso**. Disponível em:

<<https://globecommerce.com.br/o-bilionario-mercado-de-skins-de-csgo-e-como-ganhar-dinheiro-com-isso/>>. Acesso em: 12 fev. 2021.

L, Andrei. **O Que é HTML? Guia Básico Para Iniciantes**, HOSTINGER TUTORIAIS, 17 de ago. 2021. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos>>. Acesso em 30 de out. de 2021.

MITCHELL, R. **Web Scraping com Python**. 2. ed. [s.l: s.n.].

MACHADO, Cristian Cleder et al. **Um Web Crawler para Projeções e Análise de Vulnerabilidades de Segurança e Consistência Estrutural de Páginas Web**. Revista de Empreendedorismo, Inovação e Tecnologia, Passo Fundo, v. 2, n. 2, p. 3-12, mar. 2016. ISSN 2359-3539. Disponível em: <<http://seer.imed.edu.br/index.php/revistas/article/view/869>>. Acesso em: 16 jun. 2021.

MARQUES, Rafael. **O que é HTML? Entenda de forma descomplicada**, Home Host. Disponível em: <<https://www.homehost.com.br/blog/tutoriais/o-que-e-html/>>. Acesso em de 30 de out. 2021.

Newzoo. **China games market** Disponível em: <<https://newzoo.com/insights/infographics/china-games-market-2018/>>. Acesso em 16 de dez. 2020.

Portal educação, **O que é o PostgreSQL?** Disponível em: <<https://siteantigo.portaleducacao.com.br/conteudo/artigos/informatica/o-que-e-o-postgresql/4844>>. Acesso em 13 de nov. de 2021.

Parentzone. **What is skin gambling?** Disponível em: <<https://parentzone.org.uk/article/what-skin-gambling#:~:text=A%20skin%20is%20a%20graphic,the%20outcome%20of%20the%20game>>. Acesso em 08 de dez. 2020

PURCHIO, Luisa. **Rumo aos US\$ 200 bi: estratégias da indústria de games para crescer mais**, Veja, 1 de fev. de 2021. Disponível em: <<https://veja.abril.com.br/economia/rumo-aos-us200bi-as-estrategias-da-industria-de-games-para-crescer-mais/>>. Acesso em 05 de abr. 2021.

Purebreak. **Valve Corporation** Disponível em: <[https://www.purebreak.com.br/empresa/valve-corporation\\_e538872](https://www.purebreak.com.br/empresa/valve-corporation_e538872)>. Acesso em 09 de mai. de 2021.

ProWebScraping. **Web Scraping Vs Web Crawling** Disponível em: <<http://prowebscraping.com/web-scraping-vs-web-crawling/>>. Acesso em 10 de mai. de 2021.

Python Wiki. **Python** Disponível em: <<https://wiki.python.org/moin/>>. Acesso em 14 de nov. de 2021.

SANTANA, Marlesson. **Utilizando o Scrapy do Python para monitoramento em sites de notícias (Web Crawler)**, Scrapy Developers, 24 de set. de 2017. Disponível em:

<<https://medium.com/@marlessonsantana/utilizando-o-scrapy-do-python-para-monitoramento-em-sites-de-noticias-web-crawler-ebdf7f1e4966>>. Acesso em 11 mai. de 2021.

SOUZA, Ramon de. **Impulsionado pela pandemia, mercado de games segue em crescimento recorde**, CanalTech, 28 de out. de 2020. Disponível em: <<https://canaltech.com.br/games/impulsionado-pela-pandemia-mercado-de-games-segue-em-crescimento-recorde-173720/>>. Acesso em: 04 de abr. 2021.

Steam Database, **Steam**. Disponível em: <<https://steamdb.info/app/730/graphs/>>. Acesso em 03 de abr. 2021.

Steam Support. **Mercado da comunidade - Perguntas frequentes** Disponível em: <[https://support.steampowered.com/kb\\_article.php?ref=6088-UDXM-7214&l=portuguese#:~:text=A%20Taxa%20de%20Transa%C3%A7%C3%A3o%20do,aumentada%20ou%20reduzida%20no%20futuro.](https://support.steampowered.com/kb_article.php?ref=6088-UDXM-7214&l=portuguese#:~:text=A%20Taxa%20de%20Transa%C3%A7%C3%A3o%20do,aumentada%20ou%20reduzida%20no%20futuro.)> Acesso em: 06 de mai. 2021.

VELLOSO, Felipe. **Counter Strike: Global Offensive traz o clássico com uma nova roupagem**, TechTudo, 15 de dez. 2020. Disponível em: <<https://www.techtudo.com.br/tudo-sobre/counter-strike-global-offensive.html>>. Acesso em 09 de mai. de 2021.

VINHA, Felipe. **O que é Steam? Conheça a loja de jogos para PC da Valve**, TechTudo, 18 de fev. 2019. Disponível em: <<https://www.techtudo.com.br/noticias/2019/02/o-que-e-steam-conheca-a-loja-de-jogos-para-pc-da-valve.ghtml>>. Acesso em 06 de mai. de 2021.

Vooo – Insights, **Guia para iniciantes de Web Scraping em Python (usando BeautifulSoup)** Disponível em: <<https://www.vooo.pro/insights/guia-para-iniciantes-de-web-scraping-em-python-usando-beautifulsoup/>>. Acesso em: 18 de nov. de 2021.

Angular, **What is Angular?** Disponível em: <<https://angular.io/guide/what-is-angular>>. Acesso em 13 de nov. de 2021.