Consulta Automatizada para Pesquisa de Preços de Revistas em Quadrinhos

Pedro Gomes Marins[1], Sandro Roberto Fernandes[2]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais – *Campus* Juiz de Fora

[2]Núcleo de Informática - Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais – *Campus* Juiz de Fora

pedrogm.jf@gmail.com, sandro.fernandes@ifsudestemg.edu.br

Abstract. *This article aims to introduce an automated comic book pricing query tool, HqFinderWeb. The developed tool was made available on a website, which uses a Crawler to capture data on third party websites, using tools to navigate and capture the necessary information. Data is stored and later displayed to the user quickly and simply.*

Resumo. *Este artigo tem como objetivo apresentar uma ferramenta de consulta automatizada de preços de revistas em quadrinhos, o HqFinderWeb. A ferramenta desenvolvida foi disponibilizada em um website, que utiliza um Crawler para capturar dados em sites de terceiros, utilizando ferramentas para realizar a navegação e a captura das informações necessárias. Os dados são armazenados e posteriormente exibidos ao usuário de forma rápida e simples.*

1. Introdução

A utilização das tecnologias atuais proporciona maior disponibilidade de acesso a informações, contribuindo para que todos os serviços que as utilizam sejam beneficiados, assim como sites de vendas de passagens aéreas, hotéis e lojas virtuais. A globalização impulsiona a realização de tarefas online, diminuindo o tempo gasto e melhorando a produtividade do cotidiano de uma pessoa ou empresa. Portanto, compras de produtos através de websites é uma tendência em expansão, motivando os profissionais da área ao aprimoramento de ferramentas para facilitar o acesso e pesquisa pelos usuários.

Crawlers são softwares desenvolvidos com a finalidade de realizar uma varredura na internet de maneira automatizada. Também conhecidos como *Spiders* ou *Bots* (robô), um *Crawler* captura informações nos textos dos sites ou até mesmo em seus HTML.

Sites que apresentam motores de busca, em particular usam dos *Crawlers* para manter sua base de dados atualizada. Os *Web Crawlers* são utilizados principalmente para criar uma cópia de todas as páginas visitadas para um pós-processamento a ser realizado por um "motor de busca" que irá indexar as páginas baixadas para prover buscas mais rápidas. Também são utilizados para manutenção automatizada de um website, como checar textos, links e o código HTML.

Infelizmente, há *Crawlers* nocivos, utilizados por criminosos para roubar conteúdos protegidos e cometer fraudes, repassando informações de produtos e serviços de uma empresa para a concorrência, o que pode causar grandes prejuízos aos negócios. Assim, inúmeros sites apresentam ferramentas que servem como obstáculos aos *Crawlers*, a mais famosa delas é a *Captcha*: teste de desafio cognitivo, utilizado como ferramenta antispam. Um tipo comum requer que o usuário identifique as letras de uma imagem distorcida, às vezes com a adição de uma sequência obscurecida das letras ou dos dígitos que apareça na tela.

A utilização de um *Crawler* é um processo comum de ser utilizado quando se tem um trabalho mecânico e repetitivo a ser praticado na web. Assim como um colecionador de revistas em quadrinhos (HQs) ao procurar por preços e disponibilidade de produtos nos websites, desperdiça um longo período de seu dia em busca de um preço ideal e tem dificuldade em encontrar o produto que deseja.

Dado essa complexidade, viu-se a necessidade de uma ferramenta que pudesse realizar essa pesquisa de forma automática e simples, facilitando a tarefa do colecionador. Assim, a ferramenta *HqFinderWeb* foi desenvolvida.

2. Metodologia

Para automatizar uma pesquisa realizada em sites que realizam vendas de HQs, foi necessário entender o passo a passo da pesquisa feita pelo usuário em cada website, para que o *HqFinderWeb* use os parâmetros de pesquisa corretos. Em geral, toda HQ tem um título e volume, como por exemplo: "*Dragon Ball - Volume 4*". Entretanto, há a possibilidade de editoras diferentes publicarem o mesmo quadrinho. Assim, houve a necessidade de adicionar o termo Editora no parâmetro de pesquisa (Figura 1).

Figura 1. Página inicial do *HqFinderWeb*. Fonte: O autor.

O *HqFinderWeb* navega em 4 (quatro) sites diferentes (Banca do Gibi, Comix, Panini e Excelsior). Visto que, são sites que não apresentam grandes barreiras para o *Crawler*, como a presença de *javascript* e ao mesmo tempo são sites confiáveis ao consumidor. Todos estes sites seguem um padrão para sites de compras online, existe a página inicial com um campo de pesquisa com um botão pesquisar ao lado. Que resulta numa página mostrando uma lista dos resultados encontrados. Em seguida, ao clicar em um produto, leva para a página mais detalhada do mesmo. Assim o código do software foi dividido em duas superclasses, uma que realiza toda a navegação e outra que realiza a extração das informações. Para cada site foi criado uma classe que descende das superclasses. As subclasses têm o objetivo de serem bastante genéricas, para que assim futuramente possa acrescentar novos sites de pesquisa no *HqFinderWeb* e que tenha uma manutenção simples e rápida.

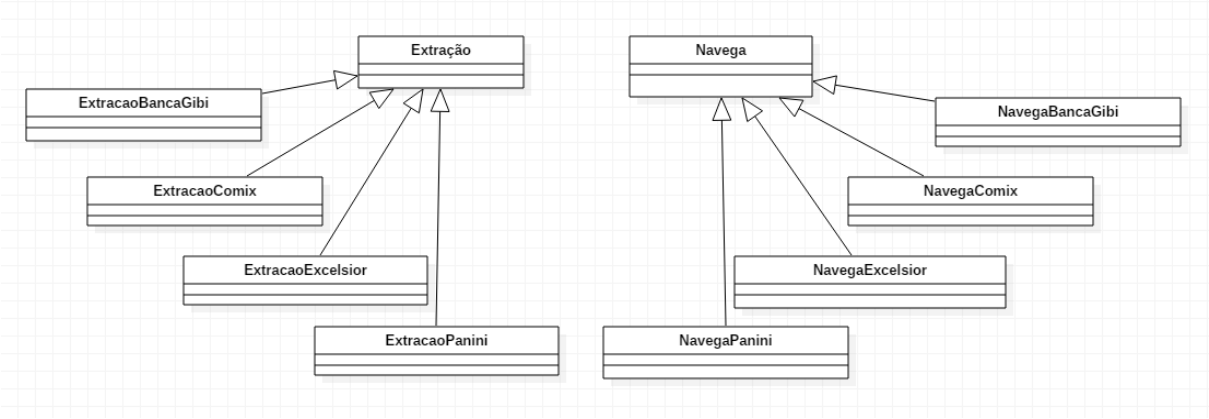


Figura 2. Diagrama de Classes do HqFinderWeb. Fonte: O Autor.

O *Crawler* navega para a página principal do site, onde por meio da extração do HTML identifica a localização da barra de pesquisa possibilitando a inserção do título e o volume do quadrinho procurado. Da mesma forma, encontra no HTML o botão pesquisar que é selecionado e direciona o *Crawler* para a próxima página, que contém uma lista dos resultados encontrados (Figura 3). Entretanto essa lista vai possuir muitos quadrinhos que não são desejados pelo usuário, como acontece na maioria dos sites.

Assim, o *Crawler* recebe o HTML desta nova página e após extrair uma lista de todos os resultados encontrados é utilizado um filtro para excluir todos os quadrinhos indesejados, restando apenas um. Porém haverá casos de o software encontrar mais de um quadrinho desejado, como foi explicado anteriormente. Assim sendo, o *Crawler* navega para a página detalhada do(s) produto(s) encontrado(s) (Figura 4), e utilizará um novo filtro pela editora. Após os filtros utilizados e restando somente um produto, é preciso conferir a disponibilidade do estoque referente ao quadrinho, em seguida finalmente é extraído o valor e o link do quadrinho.

Todas a informações são extraídas de forma automática e simples para o usuário, que irá recebê-las em questão de minutos. Excluindo a necessidade do cliente ser obrigado a ir de site em

site manualmente gastando o seu tempo, o *HqFinderWeb* se torna uma ferramenta essencial para aqueles que desejam mais praticidade e rapidez na hora de executar compras de quadrinhos online.

Figura 3. Exemplo de um site. Há a barra de pesquisa e a lista dos resultados encontrados.
Fonte: http://www.comix.com.br/

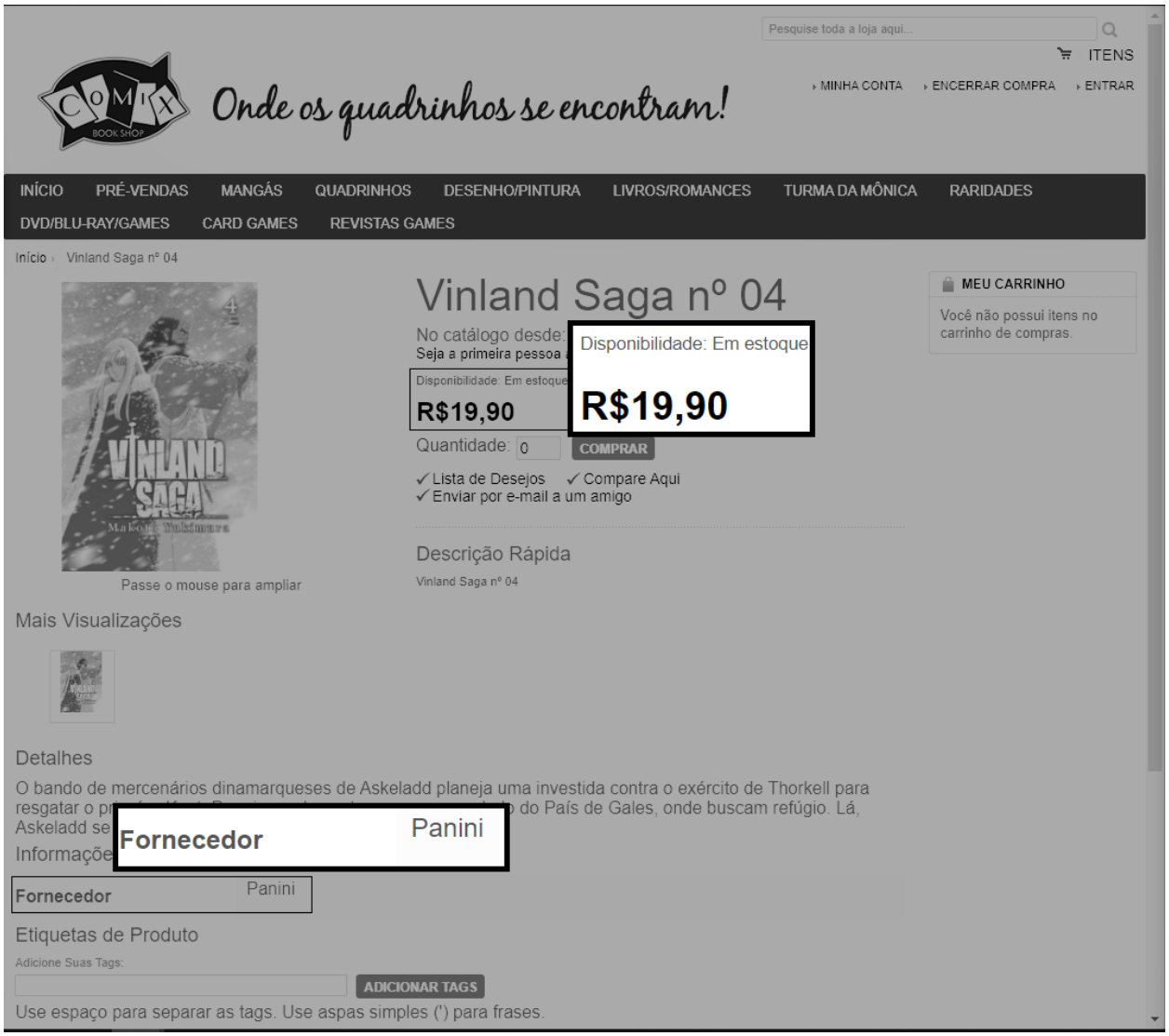


Figura 4. Exemplo de um site. Página detalhada do produto desejado. Há a verificação da disponibilidade, preço e fornecedor (editora). Fonte: http://www.comix.com.br/

Para que o *HqFinderWeb* realize às navegações necessárias foi usada o *Selenium*, uma biblioteca para testes de *softwares Web* que realiza a automação de navegadores. Podendo automatizar HTML, CSS e até *XPath* (linguagem de consulta para selecionar nós de um documento XML). Dentre várias funcionalidades contidas na biblioteca a mais interessante para o projeto é a possibilidade do *Selenium* de realizar funcionalidades de navegação, como acessar links, preencher formulários, colocar imagens, fazer *downloads*, localizar elementos, entre outras ações possíveis. Apesar de ser uma biblioteca de testes, se tornou viável o uso dela para a construção do *HqFinderWeb*. Com ela o software pode navegar e extrair as informações contidas nos sites de quadrinhos. Ao navegar, o *Selenium* utiliza a ferramenta *WebDriver*, que cria uma emulação de um navegador real.

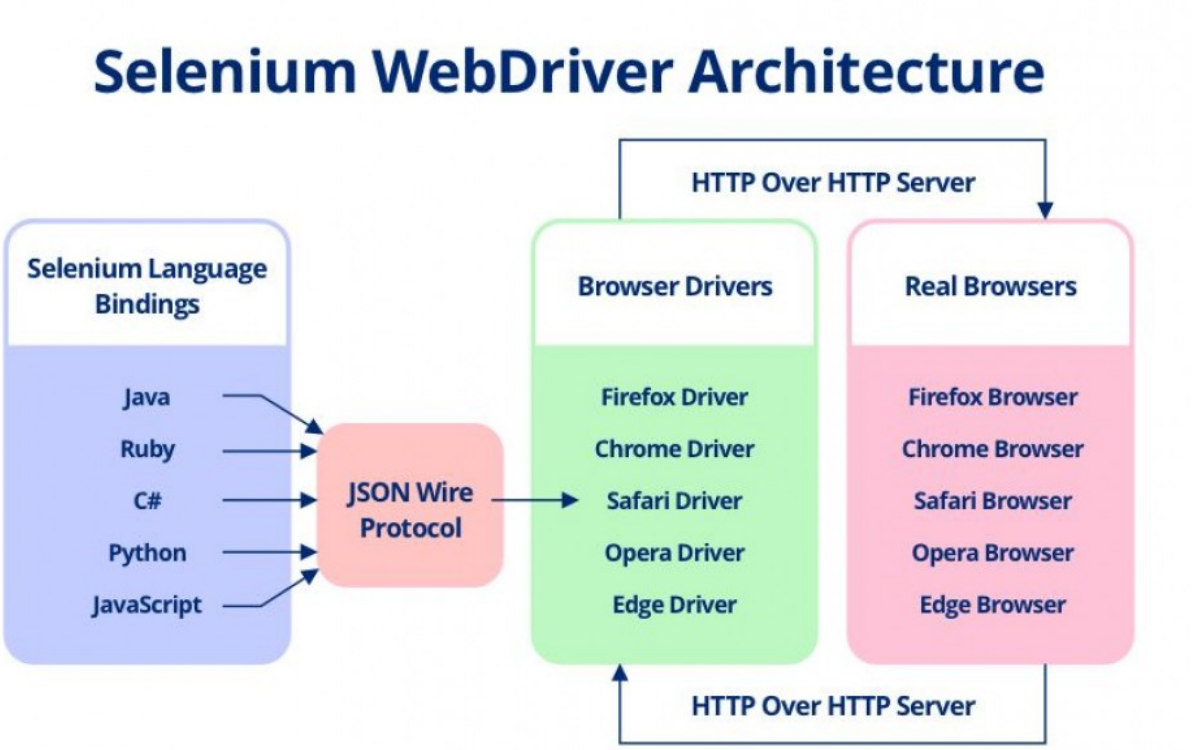


Figura 5. Arquitetura do Selenium WebDriver. Fonte: Hackr.io (https://hackr.io/blog/complete-guide-selenium-webdriver)

A navegação é divida em três etapas. Primeiramente o driver (emulando um navegador) inicializa e navega para a página inicial do site escolhido. Posteriormente, seguindo a lógica de um usuário realizando a pesquisa de forma manual, deve-se, através do driver localizar na página web a barra de pesquisa e o botão pesquisar. Essa localização é feita através da elaboração de um *XPath* que levará o driver para o nó desejado (Figura 6). Podemos ver que o código *XPath* procura por todo HTML uma *tag* chamada "*label*" que contenha o texto escrito "Pesquisa", após achar essa *tag* procura pela *tag* seguinte a essa que tenha o nome de "*input*".

Pode-se dizer que o código *XPath* é um passo a passo de como encontrar uma certa *tag* em todo o HTML de um site. Por essa razão, torna-se de grande importância a generalização do código, no sentido em que se o site realizar alguma mudança em sua estrutura, ou seja, algum nome de *tag*, *id*, *class*, etc ser mudado o *XPath* não irá encontrar a *tag* desejada. Certos sites mudam suas estruturas com certa frequência, por isso esse fator foi levado em conta para a realização deste projeto.

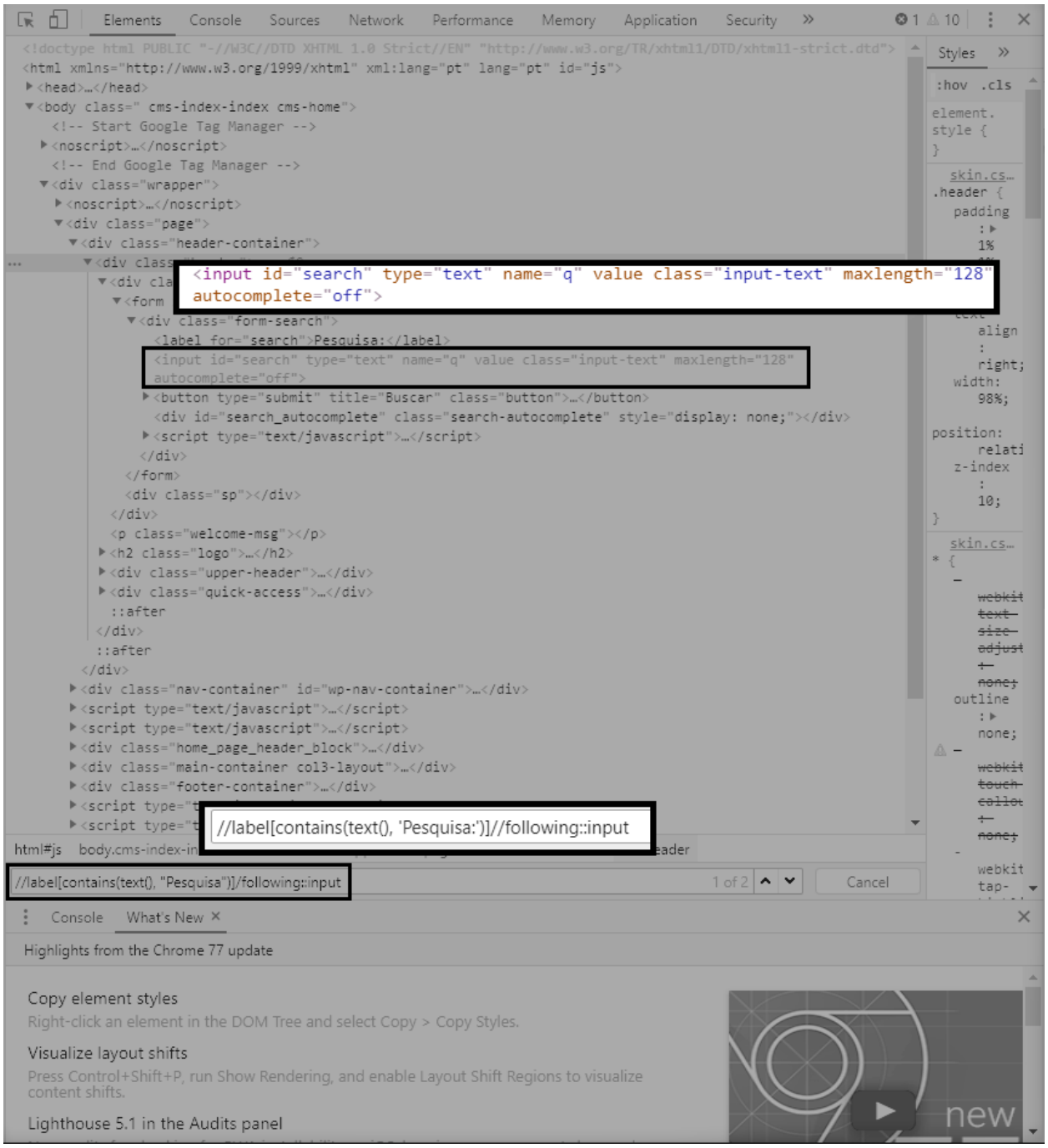


Figura 6. Página inicial. Código do XPath que leva diretamente para a barra de pesquisa. Fonte: O Autor.

Após a identificação da localização da barra de pesquisa e do botão de pesquisa, o driver insere o parâmetro passado pelo *HqFinderWeb* e executa a ação de clicar em pesquisar, levando-o para a próxima página do site e segunda etapa da navegação.

Assim que o driver for para a página de resultados, poderão acontecer duas coisas, uma lista de produtos ser exposta na página, ou uma mensagem do site dizendo que não houve nenhum resultado encontrado. O software é capaz de lidar com ambas às situações. No caso de não retornar nenhum resultado, o driver tenta localizar no HTML, usando *XPath*, a frase padrão do site que indica ao usuário que não houve resultados encontrados (Figura 7). Se o driver encontrar o aviso, o software encerra às atividades do driver com o site em questão e se dirige para o próximo.

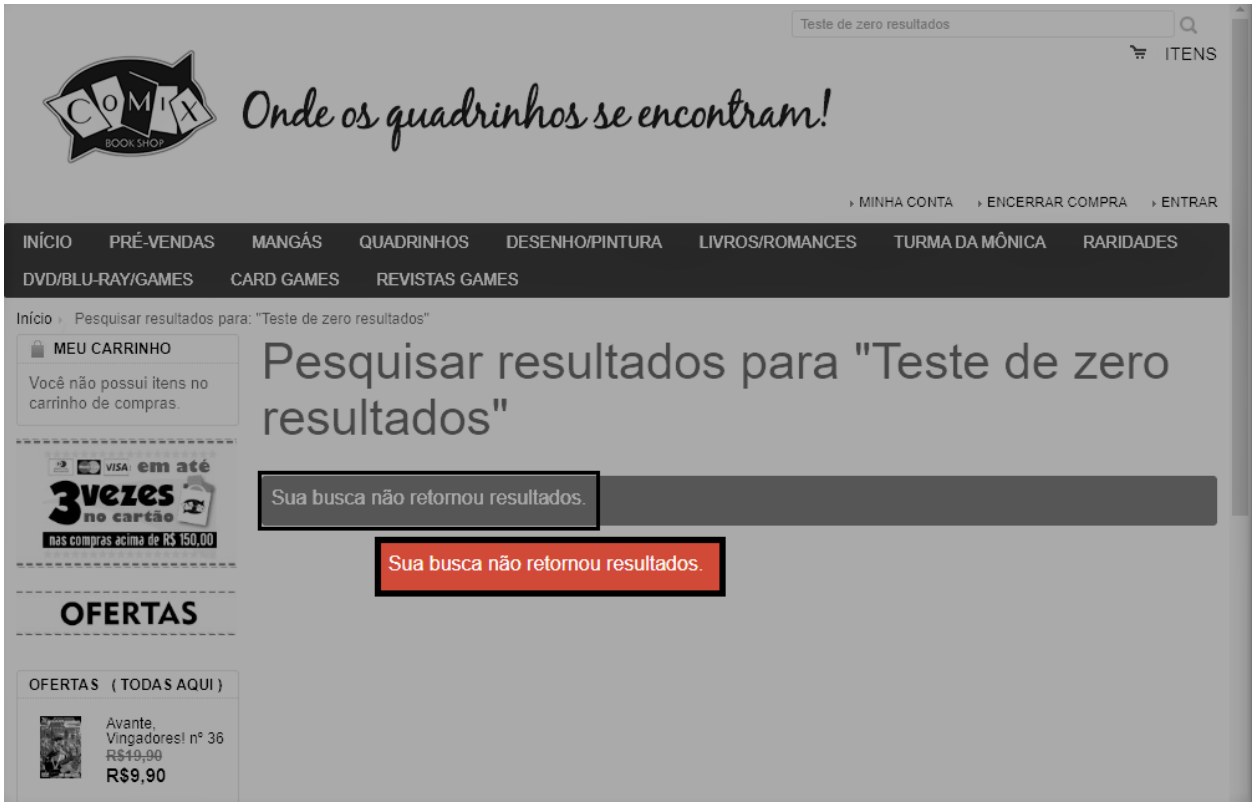


Figura 7. Página de Resultados. Nenhum produto foi encontrado. O Xpath localiza essa frase no html. Fonte: Comix (http://www.comix.com.br).

Quando o site retorna uma lista de resultados, deverá ser realizada a identificação de cada quadrinho que foi encontrado. Portanto, precisa-se encontrar no HTML, através do código *XPath*, todas as *tags* correspondentes necessárias, formando uma lista de todos os produtos encontrados.

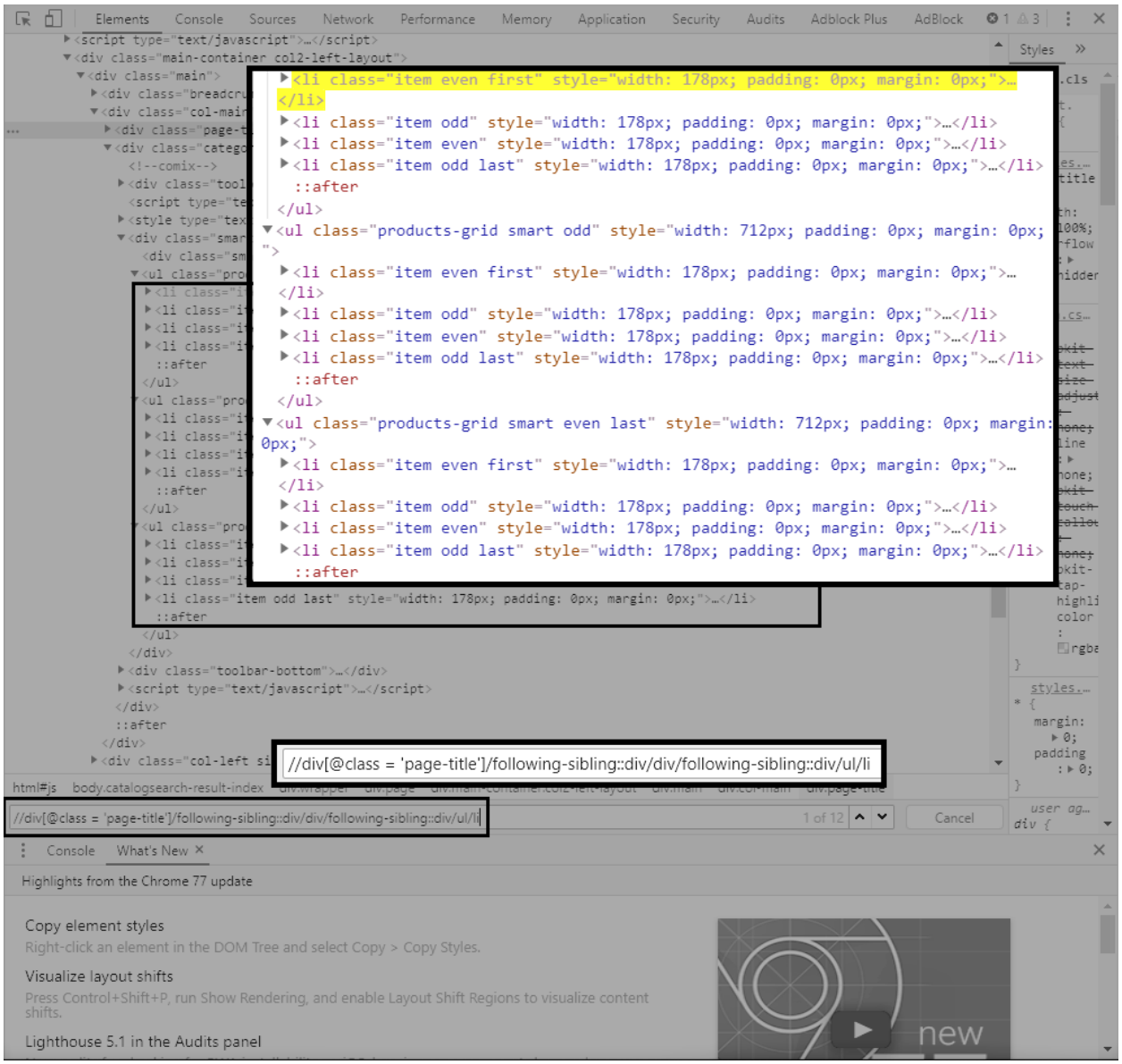


Figura 8. Página de Resultados. Lista de produtos encontrados através do XPath. Fonte: O Autor.

Cada *tag* "*li*" (Figura 8), possui informações sobre o quadrinho, como o preço, título e a URL que levará para a página detalhada do produto. Porém, essas informações estão contidas em *sub-tags* presentes em cada uma dessas "*li*". Novamente um novo *XPath* é feito para que seja possível realizar a localização dessas informações (Figura 9).

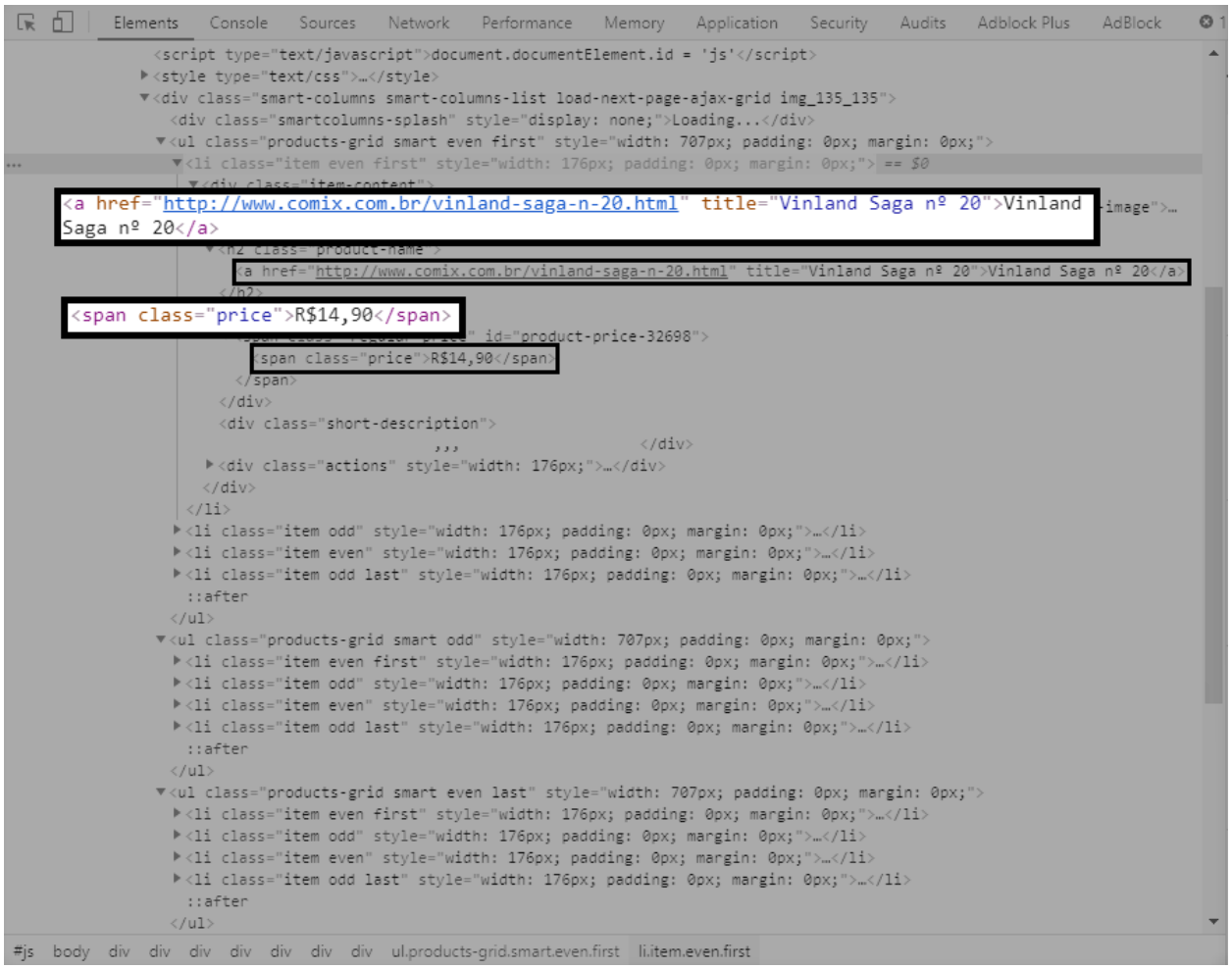
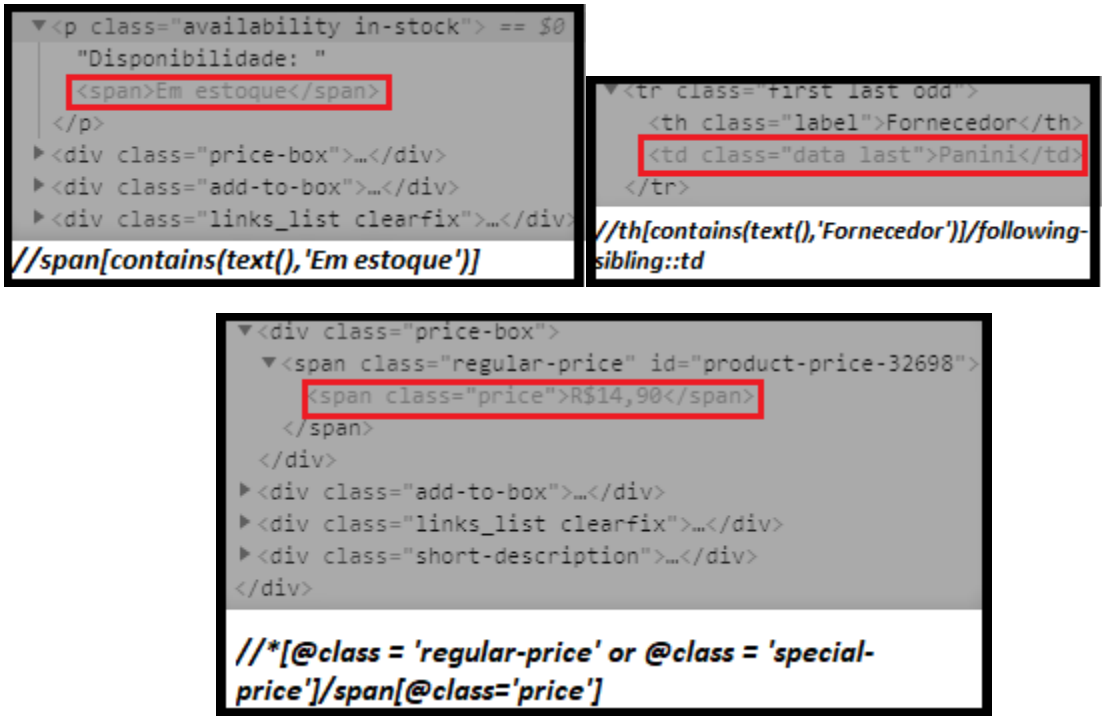


Figura 9. Página de Resultados. Detalhes de uma *tag* presente na lista de produtos. Na *sub-tag* "*a*" há a *url* que levará para a página detalhada do produto, assim como, possui o título do quadrinho. A *tag* "*span*" informa o preço. Fonte: O Autor.

Após cada uma das tag "*li*" serem varridas e terem suas informações extraídas, é feita a comparação do título de cada quadrinho com o parâmetro apresentado pelo *HqFinderWeb,* são excluídos aqueles produtos que não se encaixam. Assim, tem-se uma nova lista de *tags* chamada de *listaDeResultados* (Lista de variáveis do tipo *HtmlNode*, presente na biblioteca *HtmlAgilityPack* da linguagem C#), uma que contém somente os quadrinhos com o mesmo Título e Volume desejados pelo usuário. Usando as *urls* extraídas anteriormente, o driver é direcionado para a navegação da página detalhada de cada quadrinho dessa nova lista e assim é feito. Nessa página é possível identificar e extrair o valor, disponibilidade e editora referente ao produto (Figuras 10). Por último, usando um novo *XPath*, o driver captura essas informações, e os quadrinhos que não estiverem disponíveis para a compra são excluídos da *listaDeResultados* assim como aqueles produtos que não forem da mesma editora desejada pelo o usuário do *HqFinderWeb*. Por fim, a lista *listaDeResultados* está pronta, ela contém apenas os quadrinhos que se encaixam em todos os parâmetros passados. Em seguida o driver se dirige para o próximo site, onde irá realizar o mesmo processo e a extração da informação solicitada.



Figuras 10. Página detalhada do produto. Localização das informações com os seus respectivos XPath. Fonte: O Autor.

Assim que a navegação e a extração das informações finalizam, o software exibe para o usuário todos os resultados encontrados pelos sites. Os resultados exibidos pelo *HqFinderWeb* apresentam o valor das HQs, em ordem crescente, juntamente com o link que direciona o usuário para a página detalhada do produto onde poderá realizar a compra do produto (Figura 11).

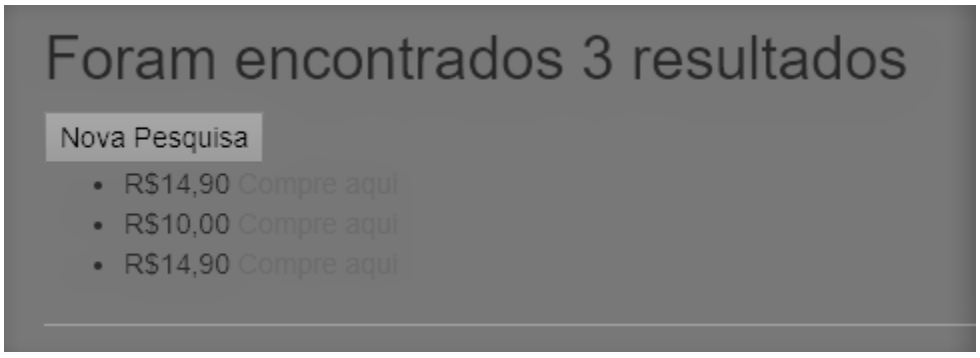


Figura 11. *HqFinderWeb,* Página dos resultados encontrados. Fonte: O Autor

3. Resultados e Discussões

Ao pesquisar por uma *HQ* nos sites navegáveis pelo *HqFinderWeb* de forma manual, o usuário tem que digitar na barra de pesquisa o quadrinho desejado. Em seguida, procurar entre a lista de

resultados encontrados a *HQ* esperada. Finalmente, entrar na página de detalhes do produto e conferir a disponibilidade e o preço do mesmo. Além do mais, esse passo a passo é repetido para cada site que o usuário deseja navegar (Figura 12).

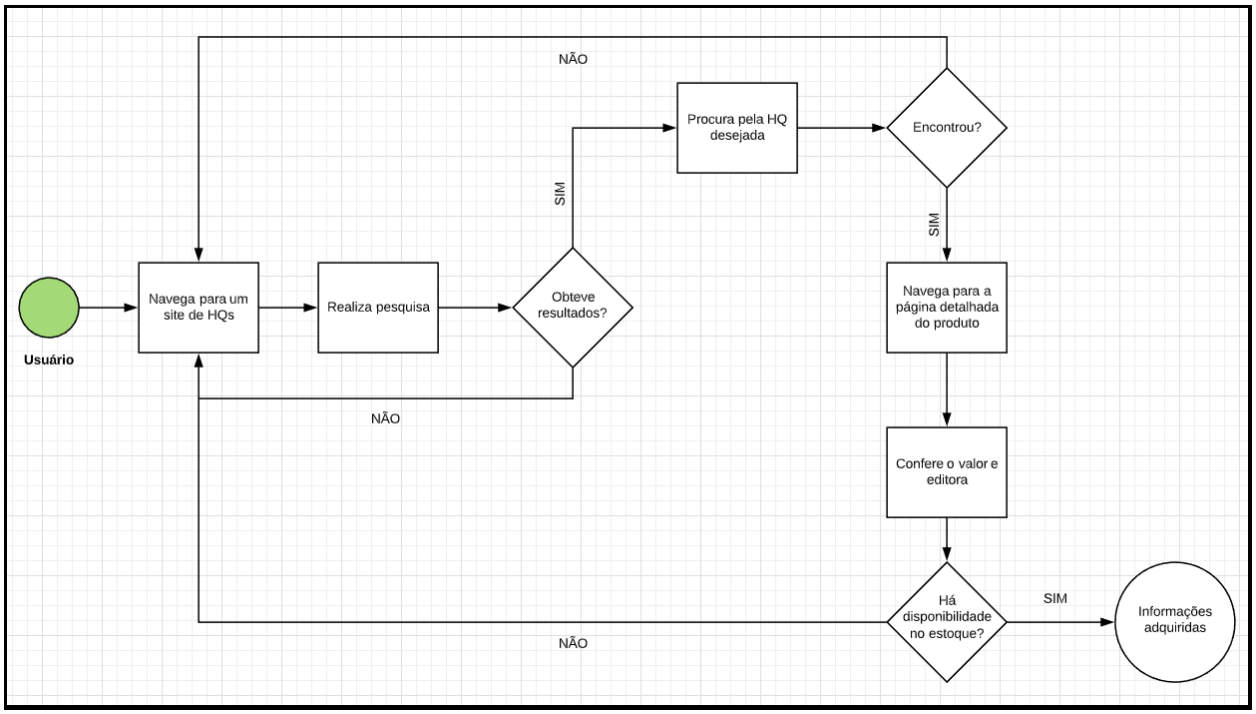


Figura 12. Fluxograma de processo. Pesquisar manualmente uma *HQ*. Fonte: O Autor.

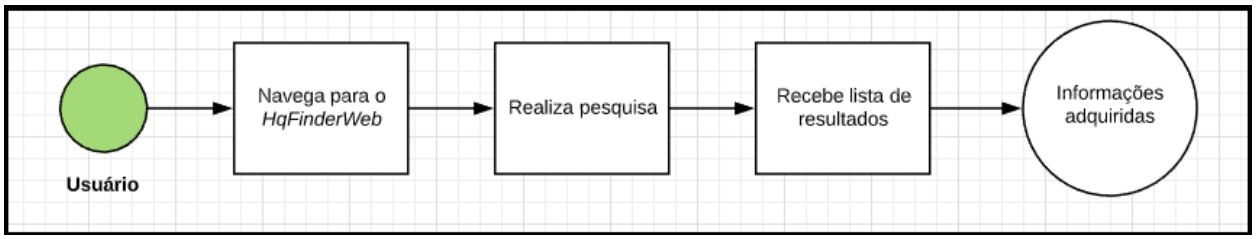


Figura 13. Fluxograma de processo. Pesquisar *HQ* pelo *HqFinderWeb*. Fonte: O Autor.

O *HqFinderWeb* conseguiu entregar para o usuário de forma rápida e simples, informações de preço, assim como, a disponibilidade dos quadrinhos. O uso do *Selenium* facilitou muito para que o software pudesse realizar suas ações de navegação e extração. Pois, usando o *WebDriver* como ferramenta foi possível que o *HqFinderWeb* tenha todos os privilégios e comportamentos de um browser web, como o Google Chrome.

A informação, sendo um valioso patrimônio empresarial, intensifica a importância da segurança da informação. Consequentemente, sites apresentam inúmeros mecanismos de defesa

contra possíveis invasores. Um desses mecanismos é a utilização de cookies, principalmente para identificar e armazenar informações sobre os visitantes. Eles são pequenos arquivos de texto que ficam gravados no computador do usuário e podem ser recuperados pelo site que o enviou durante a navegação. O *Selenium WebDriver*, com poucas linhas de código é possível de enviar os cookies necessários para a navegação.

Há também sites que apresentam em seu HTML conteúdos gerados por códigos *JavaScript*, que é tipicamente executado no nosso navegador. O *Selenium WebDriver* consegue executar esses scripts, mesmo sendo uma tarefa complicada ainda sim é uma possibilidade presente na biblioteca.

Contudo, tais benefícios acabam que prejudicam no desempenho do *HqFinderWeb,* visto que, para realizar uma pesquisa no software pode levar cerca de 3 á 4 minutos. Todavia, ainda é mais rápido e menos trabalhoso do que realizar a pesquisa manualmente.

4. Conclusões

O *HqFinderWeb* apresenta uma grande oportunidade de crescimento, pensando no número de sites navegáveis. Como o objetivo do software é atender os compradores de quadrinhos, há a grande necessidade de se ter o maior leque de sites possíveis. Deste modo, estrutura do software foi construída em favor de que isso aconteça, visto que são super-classes genéricas que realizam a navegação e extração.

Ainda sim, é importante levar em conta de que quanto mais sites o *HqFinderWeb* navegar mais processamentos será gasto. Talvez seja prudente a verificação de outras ferramentas ou maneiras de realizar a navegação nos sites. Finalizando, o *HqFinderWeb* é uma ferramenta muito eficiente para o seu estado atual (realizando a navegação em apenas 4 sites). Podendo no futuro crescer e se aprimorar cada vez mais.

Referências

PEIXOTO, Rafael. Selenium WebDriver Descomplicando testes automatizados com Java. São Paulo: Casa do Código, 2018.

Microsoft Visual Studio Community 2017: Construção de Programas na linguagem C#. Version 15.6.3. [S.l.]: Microsoft Corporation, 2017. Disponível em: <https://visualstudio.microsoft.com>. Acesso em: 21 de março de 2019.

Microsoft .NET Framework: Plataforma para desenvolvimento e execução de aplicações. Version 4.8.03752. [S.l.]: Microsoft Corporation, 2017.

Microsoft .NET Framework: Plataforma para desenvolvimento e execução de aplicações. Version 4.8.03752. [S.l.]: Microsoft Corporation, 2017.

Selenium WebDriver: Ferramenta que oferece uma API que permite a escrita de forma mais produtiva e organizada de scripts de testes. Version 78.0.03904.70000. [S.l.]: Apache License, 2004.

HtmlAgilityPack: Parser HTML que constrói uma leitura a partir dos dados do DOM e suporta XPATH simples ou XSLT. Version 1.11.16. [S.l.]: MIT License, 2017.