

# Maraboo - Um agente de recomendação pessoal auxiliar de compras para e-commerce de vestuário

Trabalho de Conclusão do Curso de  
Tecnologia em Sistemas para Internet

Matheus Alves Barbosa

Orientadora: Marcia Häfele Islabão Franco

<sup>1</sup>Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)  
Campus Porto Alegre  
Av Cel Vicente, 281, Porto Alegre – RS – Brasil

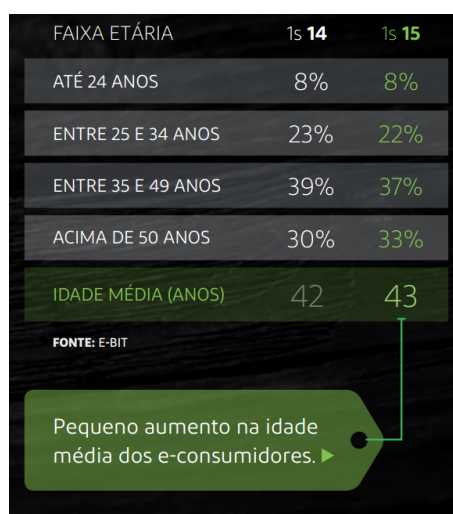
mafeus.barbosa@gmail.com, marcia.franco@poa.ifrs.edu.br

**Resumo.** *Tem sido crescente o consumo por meio de comércio eletrônico no Brasil. Entretanto, tendo em vista que o foco das lojas geralmente está no marketing e não na exposição de dados históricos sobre variações de preço ou estoque de produtos, há casos em que o consumidor não é capaz de saber se adquirir um produto é vantajoso ou não a ele. Este artigo tem como objetivos descrever brevemente as ferramentas atuais de comparação de preço entre lojas assim como expor uma solução em desenvolvimento aplicada a uma única loja virtual - a Dafiti - em forma de sistema para web, através de análise de variações de preço e estoque e da existência de promoções para perceber tendências por meio de técnicas de web crawling e scraping. Espera-se, dessa forma, que os usuários recebam recomendações com maior precisão, que a eles seja proporcionada maior transparência e que seja facilitada a percepção de vantagens em compras por meio da representação gráfica de informações pertinentes a produtos, levando assim, por meio de uma ferramenta especializada e automatizada, à economia de tempo.*

## 1. Introdução

O número de usuários de sistemas de comércio eletrônico tem sido crescente, de aproximadamente 36% a.a., conforme apontado por pesquisas realizadas entre os anos de 2001 a 2014 eCommerceOrg 2014. Se por um lado esse fato facilita a aquisição de produtos por quantidade maior de pessoas, por outro, alguns módulos de notificações de sistemas tem se mostrado menos eficientes. Há possíveis causas para essa ineficiência, sendo uma delas o aumento de potenciais clientes buscando um mesmo produto em um mesmo sistema. Nesse caso, qualquer um desses usuários interessados pode tomar conhecimento de informações relevantes como promoções ou baixas no estoque de forma tardia. Tendo ciência disso, do ponto de vista desses usuários, o real problema é que eles venham a ser decepcionados.

No entanto, um usuário de e-commerce pode contornar essa dificuldade por meios manuais, ou seja, acessando o sistema com alta frequência e observando cada um de seus produtos de interesse. Porém, pela natureza dessa solução, surge um custo: o tempo investido.



**Figura 1. Percentual de e-consumidores por faixa etária, conforme o 32º WebShoppers, relatório sobre e-commerce da E-bit do 1º semestre de 2015 E-bit 2015**

Considerando que muitas pessoas também utilizam a internet para comparação de preço e que os maiores e-consumidores pertencem ao grupo de idade de 35 a 49 anos (ver Figura 1), grupo que está contido na faixa mais significativa de pessoas economicamente ativas (25 a 49 anos, cerca de 60%) IBGE 2015, pode-se supor que a parte majoritária dos usuários do comércio eletrônico dispõe de menos tempo livre. Tendo isso em vista, é possível concluir que há um grupo significativo de usuários que seriam potenciais beneficiários de uma solução automatizada do processo de monitoramento e recomendação de produtos.

Atualmente há ferramentas que resolvem o problema de comparação de preço, todavia, dentre as analisadas neste trabalho, constatou-se que nenhuma considera propriedades específicas de lojas tais como estoque e existência de cupons de desconto, nem contemplam em suas bases de dados produtos da Dafiti. Este artigo vem mostrar que a exploração dessas características pode trazer como benefício grande chance de existir economia financeira, através da percepção por parte do sistema da demanda por um produto e prever que seu preço poderia ser diminuído para incentivar o consumo. Indubitavelmente, outro benefício seria a economia de tempo, devido ao fato de o usuário não precisar comparar preços de produtos diversas vezes ao longo de dias, já que essa deixaria de ser sua responsabilidade.

Este artigo está organizado da seguinte forma: na seção 2 são apontados os conceitos envolvidos na solução, os trabalhos relacionados são apresentados na seção 3, na seção 4 a solução desenvolvida, a metodologia utilizada é descrita na seção 5, os resultados atingidos assim como problemas encontrados são detalhados na seção 6, os trabalhos futuros na seção 7 e enfim, na seção 8 são apresentadas as considerações finais do trabalho.

## 2. Conceitos envolvidos

No tocante a comércio, não se pode negar que conhecer o perfil de um cliente permite que alguém (ou algo) lhe faça recomendações com maior probabilidade de aceitação, fato que

é benéfico a ambas as partes. No contexto desta proposta, é através de perguntas como “este produto que acabou de entrar em uma promoção pode interessar a alguém?” que os conceitos de agentes inteligentes e sistemas de recomendação tornam-se relevantes. Em simples termos, conhecendo o perfil de um usuário e agindo reativa e proativamente, a ele podem ser feitas recomendações tão precisas quanto possível, proporcionando melhor aproveitamento de seu tempo.

Nesta seção serão apresentados breves conceitos sobre Sistemas de Recomendação e as demais tecnologias envolvidas.

## 2.1. Sistemas de Recomendação

Segundo Schafer, Konstan e Riedl [2001], o termo “sistema de recomendação” evoluiu para substituir o termo “filtragem colaborativa”, pois este se refere a um algoritmo específico para recomendação, enquanto aquele é para sistemas que, dentre outras características, recomendam produtos com base na preferência de usuários e permitem que estes os avaliem. Pode-se complementar, ainda, com o entendimento de Weitzel et al.:

*“O objetivo dos Sistemas de Recomendação é inferir um conjunto de resultados relevantes ao interesse de um usuário ou um perfil ao qual se adéqüe. Essencialmente, um sistema de recomendação analisa as informações de um usuário em busca de padrões de interesse que possam ser utilizados para a recomendação de conteúdo”*

## 2.2. Tecnologias envolvidas

Há sistemas que permitem que algum usuário seu procure um certo item em diferentes lojas virtuais, permitindo-lhe comparar e/ou monitorar preços, além de receber avisos sobre suas baixas, utilizando técnicas de *web crawling* Thurow 2003 e *web scraping*, também conhecido como *web harvesting* ou *data extraction* Schrenk 2012 . Em suma, *web crawling* é uma técnica na qual é lançado um conjunto de URLs que são acessadas, seguido recursivamente por um processo de busca e acesso a outras encontradas no DOM (Document Object Model). Juntamente com isso, *web scraping* consiste no manuseio do conteúdo de elementos de hipertexto, considerados não estruturados com a finalidade de extraí-los e estruturá-los em alguma base de dados para posterior análise. Essas e outras técnicas permitem o funcionamento de um *web crawler* - ou robô, programa tipicamente executado por uma tarefa agendada (Cron) do sistema operacional - com base em regras pré-estabelecidas por seu criador.

Enquanto tarefas agendadas são consideradas rotineiras, há um conceito de propósito similar: *worker* [HEROKU 2016]. Este, entretanto, funciona de forma reativa a eventos. Um exemplo de aplicação seria o envio de notificações: se um usuário visitasse a página de um produto e fosse percebida alguma alteração de propriedade deste, esta tarefa poderia ser enfileirada para que um *worker* a executasse, permitindo que o tempo de resposta da requisição não fosse afetado pelo envio de notificações sobre a alteração.

No tocante a envio de notificações, é comum que especialmente redes sociais utilizem formas de recebimento de avisos assíncrona para que haja ganho na usabilidade do sistema. Há diferentes formas de realizar isso, dentre as quais HTTP *polling*, WebSocket e outros. Em suma, o *polling* consiste no envio constante de requisições HTTP a um servidor. Apesar de ser de fácil implementação, se o número de clientes aumenta, pode

surgir sobrecarga de rede. WebSockets, por outro lado, conforme descrito na RFC 6455 Fette e Melnikov 2011:

*“The WebSocket Protocol enables two-way communication between a client running untrusted code in a controlled environment to a remote host that has opted-in to communications from that code. [...] The goal of this technology is to provide a mechanism for browser-based applications that need two-way communication with servers that does not rely on opening multiple HTTP connections.”*

ou seja, através de uma única conexão é possível receber constantemente dados, eliminando o *overhead* gerado e recebendo com maior agilidade a resposta.

No contexto do e-commerce, *web crawlers* são responsáveis por, dentre outras atribuições, gravar informações de produtos, comparar preços, verificar estoque e gerar notificações. Para que eles sejam considerados eficientes nesse sentido, é indispensável que mantenham catalogados registros sempre mais próximos da atualidade. Isso concerne às ferramentas de comparação de preço, uma vez que estas consomem recursos das lojas virtuais às quais elas fazem consultas, e por esse motivo há políticas de educação de *web crawling* Mischel 2011, dentre as quais, a de evitação de sobrecarga de rede. Tais ferramentas devem, portanto, implementar essas políticas.

### **3. Trabalhos relacionados**

Há diversos sistemas que resolvem o problema de comparação de preços, entre eles, Bondfaro, Buscapé e Baixou. Feita a análise destes, pôde-se concluir que eles possuem características e propósitos similares: constatou-se que todos dão suporte a diversas lojas e permitem que a variação de preço conforme o tempo seja vista graficamente, e ainda indicam se um momento é adequado para a compra de forma mais vantajosa a um usuário.

Até o presente momento, por mais que essas soluções sejam úteis e aplicáveis a vários casos, pelo fato de serem genéricas, (1) não permitem que características específicas de lojas virtuais, como a gestão de estoque e a existência de promoções, sejam exploradas para trazer mais benefícios a seus usuários, e (2) nenhuma inclui o catálogo da Dafiti, a loja exclusivamente suportada pelo trabalho deste artigo.

É exemplo de não consideração dessas características a gestão de estoque, que difere entre as lojas. Uma gestão que trabalhe com estocagem limitada e rara reposição possibilita a previsão de baixa de estoque e diminuição de preço, cuja exploração consiste na recomendação precisa de momento vantajoso de compra. Outro exemplo é a existência de promoções que não são normalmente percebidas por *web crawlers*, que diminuem o valor total de um pedido em um momento posterior à busca por produtos - por serem aplicáveis sob condições definidas em qualquer tempo pelas próprias lojas, não podem ser previstas por um algoritmo de busca sem que haja intervenção humana.

Conforme apresentado no Quadro 1, a solução proposta considera dados de estoque de produtos, essencialmente para fins de recomendação. A importância dessa característica pode ser revelada através da percepção de problemas frequentes no negócio, por exemplo, um cliente comprar um produto específico antes de outro. Analisando as variações de estoque conforme o tempo, pode-se perceber uma tendência no consumo de

**Quadro 1. Comparativo dos Sistemas**

|                               | Bondfaro | Buscapé | Baixou | Maraboo |
|-------------------------------|----------|---------|--------|---------|
| Suporte a múltiplas lojas     | X        | X       | X      |         |
| Monitoramento de preço        | X        | X       | X      | X       |
| Notificação de baixa de preço | X        | X       | X      | X       |
| Suporte ao catálogo da Dafiti |          |         |        | X       |
| Monitoramento de estoque      |          |         |        | X       |

um determinado produto. Por exemplo, sendo a demanda baixa, o sistema pode recomendar que o usuário espere para realizar a compra, por haver a possibilidade de o preço diminuir no futuro. Se, por outro lado, o estoque estiver diminuindo com mais rapidez, pode-se recomendar que o usuário efetive brevemente sua compra.

É fundamental destacar que esta proposta, contrariamente aos trabalhos relacionados citados, não dará suporte a diversas lojas online. Essa característica é intencional, uma vez que o propósito da proposta é aproveitar ao máximo as características de apenas uma loja, a fim de otimizar a experiência de seus usuários, levando-os à fidelização. Tendo isso em vista, será dado suporte à Dafiti, pela qual também foi averiguado não haver suporte pelos demais sistemas analisados, até o presente momento.

#### **4. Maraboo**

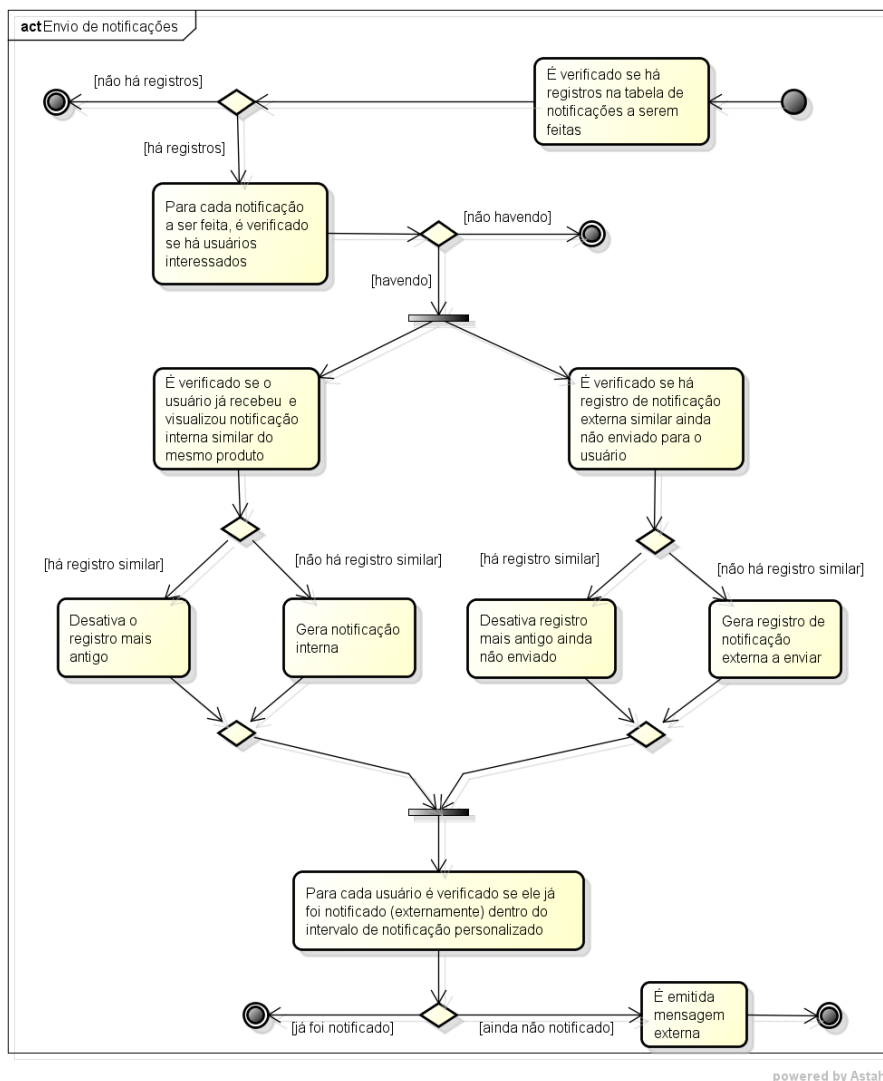
Para resolver os problemas mencionados na seção anterior, no segmento do vestuário online, esta seção apresenta o desenvolvimento do Maraboo, um sistema para a web destinado a usuários clientes da Dafiti. A Dafiti foi escolhida por ser uma das atuais líderes desse mercado, possuir políticas de pós-venda flexíveis, ter vasta quantidade de produtos e promoções frequentes. Da perspectiva do sistema em si, ela também foi escolhida por permitir a análise dos dados, essenciais ao seu funcionamento.

Considerando que um dos princípios do sistema proposto é a exclusividade, considera-se ideal que a experiência de uso do sistema seja similar à da própria loja explorada, trazendo assim maior familiaridade e facilidade no uso. Uma das formas de concretizar isso é através do uso do mesmo mecanismo de busca da loja para consultar produtos. Dessa forma, o Maraboo age como mediador: um parâmetro de busca do usuário é recebido por ele e remetido à Dafiti; o resultado da pesquisa é, então, reenviado ao usuário.

Uma vez que é recomendável que o número de acessos ao site da loja por parte do sistema seja o menor possível, essa interceptação de consultas pode ser explorada para propósitos invisíveis ao usuário. Por exemplo, os preços de produtos encontrados no catálogo podem ser usados para atualizar o histórico de variações de preço de produtos; este registro, por sua vez, pode gerar notificações a usuários. A execução dessas ações

geraria atraso considerável no processamento da requisição do usuário e para isso são utilizados *workers*. Dessa forma, o usuário vê com maior rapidez as informações que lhe interessam e a tarefa de atualizar produtos e gerar notificações é encaminhada a um *worker*. Havendo alguma notificação ao próprio usuário requerente, o uso de *websocket* poderá permitir que o recebimento de notificações ocorra assincronamente.

Após realizar uma pesquisa, é permitido ao usuário optar por solicitar monitoramento de um produto ou ver mais informações sobre ele. No caso da requisição de monitoramento, levando em conta que a forma de notificação é personalizável, surgem as seguintes opções: i) receber aviso somente quando o preço baixar de certo valor ii) receber avisos sobre variações de estoque iii) receber avisos sobre qualquer variação de preço. O sistema também permitirá marcar um produto ou categoria de produtos como indesejada. Pode-se observar que a economia de tempo é fator intrínseco à proposta e tal funcionalidade contribuiria para esse ponto, permitindo que o usuário visualize somente o que lhe interessar.



**Figura 2. Diagrama de atividades - Envio de notificações**

É comum que usuários tenham preferências diferentes quanto a frequência de re-

cebimento de alertas, e pesquisas realizadas pelo E-bit em 2015 apontam que eles se sentem incomodados quando as recebem sem permissão ou repetidamente. Tendo isso em vista, o sistema permite que o usuário informe a ele de que formas (e-mail, SMS, possivelmente por aplicativo para android) e com que regularidade quer ser notificado. Para isso, o sistema possui uma lista de agendamento de notificações no banco de dados, na qual há dados referentes a produtos que podem ser informado a usuários. Periodicamente é verificada a existência de registros nela e, havendo, para cada usuário, é conferido se este já não recebeu notificações em tempo inferior ao intervalo definido por ele, caso em que a notificação é efetivada.

No contexto do sistema, há duas formas de notificação: internas e externas (e-mail/SMS). Durante a execução da rotina de monitoramento, sempre que é encontrada alguma alteração, tanto emite-se uma notificação interna aos usuários interessados, como também gera-se um registro na tabela de notificações externas a serem enviadas. Para o funcionamento adequado da emissão de mensagens externas, não haverá duplicidade de notificação a usuários nem envios demasiadamente frequentes. Esse processo ocorrerá conforme ilustrado na Figura 4.

Para que ao usuário seja sempre mostrada a informação de forma atualizada, quando ele acessa o sistema ou está prestes a ver sua lista de interesses, é ativada a mesma funcionalidade da rotina de busca por alterações. Dessa forma, outros usuários também podem ser notificados quanto a alterações nesses produtos, caso estes estejam em suas listas de interesses.

Diversas vezes ao dia, com intervalos ainda não definidos, por meio de tarefas agendadas, serão feitas consultas ao sistema da loja para a coleta de dados, utilizando a técnica de *web scraping*. Com menor frequência haverá buscas ao catálogo a fim de realizar coleta rápida de preços e descobrir produtos novos, permitindo que possivelmente seja agendada uma notificação de recomendação.

---

**Algoritmo 1** Cálculo de tempo mínimo intervalar entre requisições

---

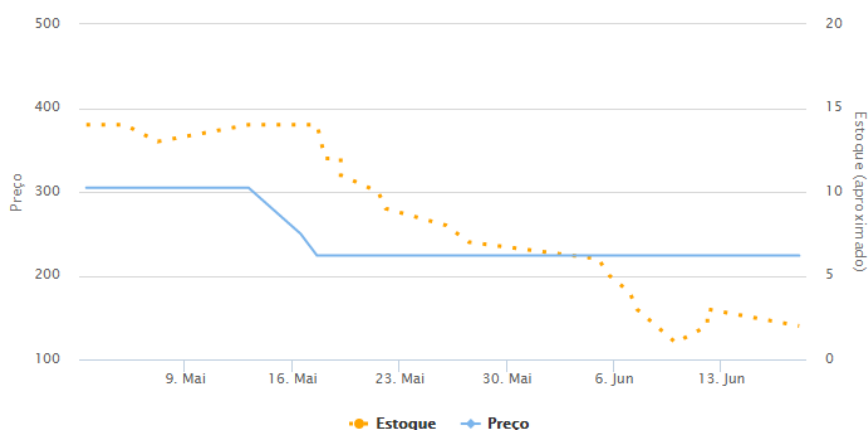
Sendo  $q$  a quantidade de um tamanho de interesse de um produto,  $t$  o tempo decorrido desde a última alteração registrada e  $i$  o intervalo mínimo a ser respeitado antes de uma nova requisição:

- 1: **se**  $q \geq 3$  e  $t < 30$  **então**
  - 2:      $i \leftarrow 24h$
  - 3: **se não se**  $q \in [1, 2]$  **então**
  - 4:      $i \leftarrow 6h$
  - 5: **se não se**  $q = 0$  e  $t < 30$  **então**
  - 6:     **se**  $t > 14d$  **então**
  - 7:          $i \leftarrow (t - (14 * 66/16))h$
  - 8:     **se não**
  - 9:          $i \leftarrow 6h$
  - 10: **se não se**  $30 \leq t \leq 45$  e  $q = 0$  **então**
  - 11:      $i \leftarrow 72h$
  - 12: **se não**
  - 13:      $i \leftarrow \infty$
  - 14: **retorna**  $i$
-

Com maior frequência é executada a rotina de monitoramento, sobre cujo processo é importante que haja detalhamento. Considerando que existe ética em termos de *web crawling*, foi estipulado um tempo mínimo intervalar entre requisições a páginas de produtos de interesse de usuários, gerando assim concentrações menores de tráfego de rede para os produtos cuja probabilidade de alteração de qualquer propriedade seja menor. Em outras palavras, considerou-se que alguns produtos merecem mais atenção que outros e para tal, foi criado o algoritmo ilustrado no pseudocódigo 1.

Assim sendo, feita uma consulta ao sistema da Dafiti, havendo alguma variação em preço ou estoque, esta é registrada no histórico de alterações do produto e são emitidas notificações aos usuários interessados.

Havendo a coleta periódica de dados, torna-se possível resolver o problema de descobrir se uma compra é vantajosa. Como pode ser vista na Figura 4, o registro dos dados, exibido de forma gráfica e textual, permite que ao usuário seja concedida maior transparência no que concerne ao valor de algum item. Exibindo a variação de estoque aproximado, pode-se perceber uma tendência de consumo, fato que pode determinar uma decisão de compra ou não; por exemplo, se o preço fosse constante e o estoque estivesse baixando, poderia-se concluir que há chance de o preço não ser reduzido e eminentemente haver indisponibilidade do produto por falta de estoque.



**Figura 3. Gráfico de produto no sistema: relação preço e estoque X tempo**

Não se pode deixar de mencionar as promoções, que de igual forma fazem parte da proposta de prover transparência. Apesar de atualmente não serem exibidos seus períodos de validade no gráfico, ao administrador do sistema é possível registrá-las. O sistema logo busca os produtos vinculados a elas, notifica usuários interessados de sua existência e finalmente, torna-se possível ver, na página de um produto, quais promoções estão ativas ou não, assim como os respectivos períodos de validade.

Para otimizar o processo evolutivo do sistema, foi adotada uma metodologia de desenvolvimento incremental e iterativa. Com o objetivo de alcançar a agilidade necessária ao processo de desenvolvimento, o sistema foi desenvolvido em PHP com o framework Yii2, por fazer uso de uma arquitetura MVC e resolver diversos problemas de validação, acesso a banco de dados, entre outros com facilidade.

A Figura 4 ilustra o diagrama de casos de uso do sistema na visão do usuário.



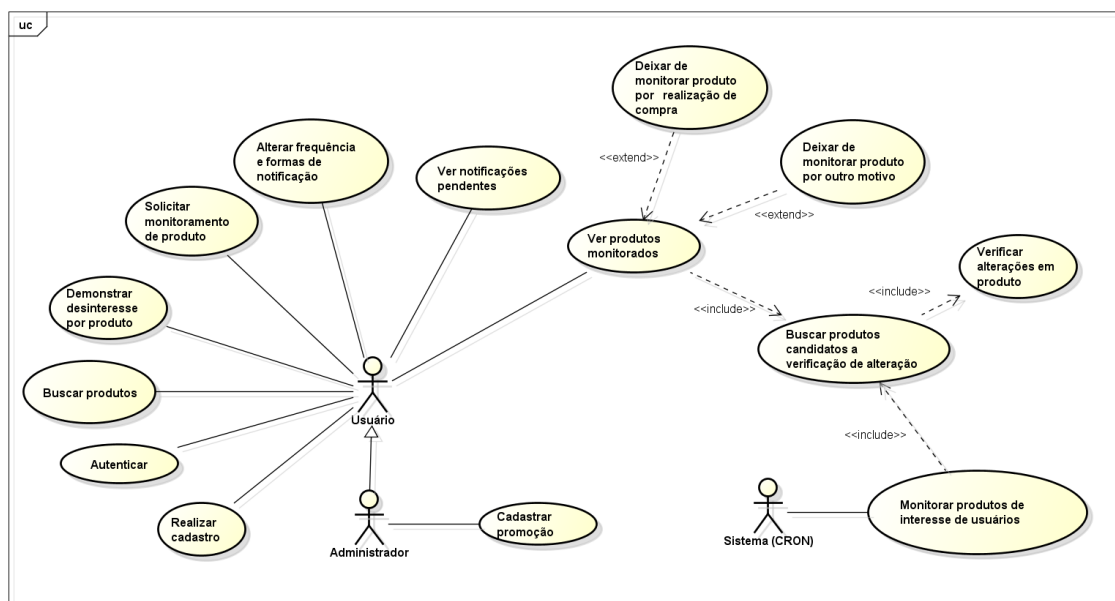


Figura 4. Diagrama de casos de uso - Ações do usuário

## 5. Metodologia

Primeiramente foi verificado se é possível consultar os dados de produtos, via API ou *web crawling*, para ter ciência da facticidade do sistema. Ao descobrir que não havia uma camada de web services disponível, foi necessário explorar as páginas de produtos e catálogo da Dafiti para encontrar padrões na estrutura da linguagem de marcação, o HTML. Além disso, procurou-se encontrar um ponto de acesso que retornasse apenas os dados necessários - estoques e preços - ao invés de uma página inteira contendo dados irrelevantes. Para isso, foi utilizada a barra de desenvolvimento do navegador Google Chrome com a qual é possível verificar, entre outras coisas, o tráfego de rede gerado por chamadas assíncronas de javascript, descobrindo assim como montar URLs de acesso a recursos.

Em seguida, foi realizado o estudo das ferramentas atuais que buscam resolver o problema já destacado, e pôde-se concluir que não há ferramenta com propósito equivalente, apenas similar.

Para fundamentar e realçar as contribuições do desenvolvimento de um novo sistema, foram consultadas pesquisas estatísticas recentes sobre o estado do e-commerce no Brasil ou relacionadas, assim como técnicas de sistemas de recomendação e os demais conceitos previamente destacados.

Concluída a revisão bibliográfica, foram criadas as estruturas de persistência de dados e listagem das funcionalidades. Decidiu-se usar um banco de dados relacional (MySQL) para persistir os dados básicos do negócio, a partir do qual foram produzidos os diagramas associados: entidade-relacionamento, casos de uso e atividades. Decidiu-se usar, também, um banco de dados não-relacional (Redis) para o agendamento de tarefas a serem realizadas por workers, caso do envio de notificações.

Logo após, foi realizado o desenvolvimento do sistema. Mostrando-se positivo o estudo de viabilidade, incluiu aprofundamento na examinação dos meios de coleta de dados (tendo em vista que constitui o núcleo do sistema, é indispensável que seja tão otimizado como possível), elaboração do projeto de interface, análise quanto à arquitetura (framework) adotada, estudo e definição da identificação do robô, efetiva programação e realização de testes.

Futuramente serão avaliados relatos de usuários testadores-alfa, que poderão incorporar novas demandas. Nesse caso, será mensurada a qualidade percebida por eles a fim de melhorar a experiência de uso do sistema nas versões subsequentes.

## **6. Resultados e discussão**

Até este momento, foi possível criar o sistema cumprindo os requisitos mínimos que motivaram sua existência. É possível cadastrar-se, buscar produtos, visualizar um produto assim como seu gráfico de variação de estoque e preço sobre o tempo, incluir produtos na lista de interesses e receber notificações sobre alterações referentes a preços, promoções e estoque. Naturalmente, surgiram problemas durante a etapa de desenvolvimento. Um deles foi associado à biblioteca utilizada para converter o DOM em um objeto PHP (Simple HTML Dom v1.5), que possui uma restrição arbitrária padrão de 600 kilobytes por arquivo HTML a ser analisado. O fato só foi descoberto após verificação de diversas propriedades de documentos diferentes, válidos e inválidos, e foi corrigido através da alteração da constante que definia essa limitação.

Sabendo que os dados de produtos são extraídos através da análise do HTML, considerou-se mapear as estruturas para classes do PHP, com vista à facilidade no reuso ao longo das interfaces do sistema. O processo levou tempo considerável e, mesmo meses após a primeira versão, defeitos continuaram a ser encontrados conforme diferentes testes eram feitos. O propósito de criar componentes reusáveis, contudo, foi atingido, e isso permitiu evolução acelerada dos módulos.

O registro de estoque real é um fator de grande relevância à proposta do sistema, conforme demonstrado neste artigo. Era possível obter tal dado por meios relativamente simples, porém por volta do fim do ano de 2015, ele passou a ser ocultado na loja explorada. Como não foi encontrada alternativa de baixo custo em termos de número de requisições HTTP sobre resultados obtidos, o problema permanece não resolvido.

## **7. Trabalhos futuros**

Apesar de funcionalidades terem sido implementadas, e o domínio “maraboo.com” ter sido registrado, ainda resta que o sistema seja publicado para que seja feito um teste de aceitação com os usuários testadores, permitindo assim que possíveis melhorias sejam feitas.

Além disso, há operações que devem ser aprimoradas. Considerando as políticas de web crawling e que há por volta de 150.000 produtos atualmente cadastrados na loja, é necessário que haja um algoritmo de seleção de produtos candidatos à busca por alterações de dados, ou seja, que na rotina de monitoramento sejam contemplados produtos que não explicitamente são de interesse de usuários, porém possivelmente venham a ser.

Tal proposta de melhoria está fortemente relacionada ao conceito de sistemas de

recomendação, ou seja: devem surgir mecanismos capazes de mapear perfis dos usuários e sejam capazes de fazer recomendações inteligente, confiável e proativamente.

## 8. Considerações finais

O comércio virtual, desde que começou a ganhar confiabilidade, passou a obter um espaço crescente no interesse das populações. Juntamente com as inovações tecnológicas, técnicas têm sido desenvolvidas objetivando melhorar a experiência de todas as partes envolvidas, possibilitando a criação de ferramentas auxiliares de recomendação, comparação de preço entre outras utilidades, claramente úteis aos consumidores finais. No entanto, respeitando delimitações éticas, ainda há benefícios a serem explorados, fato que motivou a elaboração desta proposta de solução.

Até este momento, foi concluída a fase de viabilidade inicial, na qual se pôde encontrar uma forma de buscar dados sobre preço, estoque e outras informações de produtos, assim como a modelagem inicial, incluindo diagramas de casos de uso e atividade. A partir destas etapas, foram desenvolvidos módulos do sistema a fim de cumprir os requisitos mínimos que motivam sua existência. É possível cadastrar-se, buscar produtos, visualizar um produto assim como seu gráfico de variação de estoque e preço sobre o tempo, incluir produtos na lista de interesses e receber notificações sobre alterações referentes a preços, promoções e estoque.

Futuramente, espera-se que o sistema seja publicado para que haja um teste mais intensivo de aceitação com testadores, permitindo assim que possíveis melhorias sejam feitas. Espera-se obter um sistema que irá permitir que os usuários, com poucos cliques, deleguem a ele a tarefa de estar constantemente observando um produto e avisá-los somente quando for o momento (verdadeiramente) ideal para realização de uma compra. Dessa maneira, será possível a eles poupar dinheiro, assim como investir seu tempo economizado naquilo que lhes for agradável ou útil.

## Referências

- E-bit [2015]. *Webshoppers. 32ª edição. 1º Semestre/2015*. Online: acessado em 2015-11-21. Disponível em [http://img.ebit.com.br/webshoppers/pdf/32\\_webshoppers.pdf](http://img.ebit.com.br/webshoppers/pdf/32_webshoppers.pdf).
- eCommerceOrg [2014]. *Pesquisas sobre o Consumidor on-Line*. Online: acessado em 2015-11-21. Disponível em <http://www.e-commerce.org.br/stats.php>. URL: <http://www.e-commerce.org.br/stats.php>.
- Fette, I. e A. Melnikov [2011]. *The WebSocket Protocol. RFC 6455*. <http://www.rfc-editor.org/rfc/rfc6455.txt>. RFC Editor. URL: <http://www.rfc-editor.org/rfc/rfc6455.txt>.
- HEROKU [2016]. *Background Jobs with Workers in PHP*. Online: acessado em 2016-05-20. Disponível em <https://devcenter.heroku.com/articles/php-workers>.
- IBGE [2015]. *Pesquisa Mensal de Emprego. Outubro de 2015*. Online: acessado em 2015-11-21. Disponível em <http://www.sidra.ibge.gov.br/bda/tabela/protabl.asp?c=2041&i=P&nome=on&opn110=2&notarodape=on&tab=2041&unit=0&pov=3&OpcTipoNivt=1&nivt=0&sec58=1142&sec58=2792&sec58=100052&sec58=100053&sec58=5502&>

orp=4&opv=1&pop=3&qtu110=1&orv=2&opc58=1&sev=891&sev=1000891&opp=1&poc58=2&cabec=on&orc58=3&opn7=2&decm=99&ascendente=on&sep=57176&sep=50036&orn=1&qtu7=6&pon=2&OpcCara=50&proc=1.

Mischel, Jim [2011]. *Jim's Random Notes. Writing a Web Crawler: Politeness*. Online: acessado em 2015-10-27. Disponível em <http://www.blog.mischel.com/2011/12/20/writing-a-web-crawler-politeness/>.

Schafer, J Ben, Joseph A Konstan e John Riedl [2001]. "E-commerce recommendation applications". Em: *Applications of Data Mining to Electronic Commerce*. Springer, pp. 115–153.

Schrenk, Michael [2012]. *Webbots, spiders, and screen scrapers: a guide to developing Internet agents with PHP/CURL*. No Starch Press.

Thurrow, S. [2003]. *Search Engine Visibility*. Voices Series. Disponível em <https://books.google.com.br/books?id=6pLgoklkpsIC>. New Riders. ISBN: 9780735712560.

Weitzel, Leila et al. "Proposta de métricas de avaliação da qualidade da informação médica para Sistemas de Recomendação baseados no perfil do usuário". Em: *CADERNOS de Informática 5.1*, pp. 23–48.