

## ***Question's Advisor: um aplicativo para auxiliar estudantes de programação a escreverem melhores perguntas***

**José Remígio<sup>1</sup>, Franck Aragão<sup>1</sup>, Cleyton Souza<sup>2</sup>**

<sup>1</sup>Instituto Federal de Educação, Ciência e tecnologia da Paraíba (IFPB) – Monteiro – PB – Brasil.

<sup>2</sup>Universidade Federal de Campina Grande (UFCG) – Campina Grande – PB – Brasil.

{jose.rafael.feitosa, fjunior.aragao, cleyton.caetano.souza}@gmail.com

**Resumo.** *Disciplinas de programação costumam exigir um grau elevado de esforço por parte dos estudantes, pois envolvem a compreensão e resolução de problemas com a construção de algoritmos utilizando alguma linguagem de programação. Muitos estudantes acabam recorrendo a Sites de Perguntas e Respostas a fim de encontrar alguma ajuda. No entanto, esses ambientes apresentam alguns desafios no que se refere à obtenção ou não de ajuda. Alguns estudos identificaram que as características da pergunta podem influenciar no potencial dela receber resposta ou não. Desta forma, foi desenvolvido um aplicativo para auxiliar estudantes novatos em programação a escreverem perguntas melhores.*

### **1. Cenário de uso**

Disciplinas de programação são requisitos obrigatórios em cursos da área de computação (SBC, 2003). Estas disciplinas costumam exigir um grau elevado de esforço por parte dos estudantes, pois envolve a compreensão e resolução de problemas com a construção de algoritmos utilizando alguma linguagem de programação (VAHLICK, 2007 & MOTA *et al.*, 2013).

Comumente, os estudantes de programação recorrem a Sites de Perguntas e Respostas, como, por exemplo, o G.U.J ou o Stack Overflow, para obter ajuda quando se deparam com algum problema (REMIGIO *et al.*, 2016). Estes ambientes de compartilhamento de perguntas e respostas se tornaram bastante populares. O Stack Overflow, por exemplo, conta com a presença de mais de 6.5 milhões de usuários. Além disso, em 2016, ele recebeu cerca de 13 milhões de perguntas e 21 milhões de respostas (STACKEXCHANGE, 2016).

No entanto, estes locais também apresentam alguns desafios no que diz respeito à formulação de perguntas e obtenção de respostas. Um dos desafios deste processo é a dificuldade que os alunos têm em expressar seus problemas na forma de perguntas (SULLINS *et al.*, 2015).

Alguns trabalhos recentes identificaram uma correlação entre as características da pergunta e as chances dela ser respondida (ASADUZZAMAN *et al.*, 2013; CHUA & BANERJEE, 2015 & HAO, SHU & IRAWAN 2014). A adição dessas características (consideradas “boas”) na pergunta pode torná-la mais “atrativa”, aumentando as chances dos estudantes de encontrar ajuda mais rapidamente, além de possivelmente, aumentar a qualidade das perguntas.

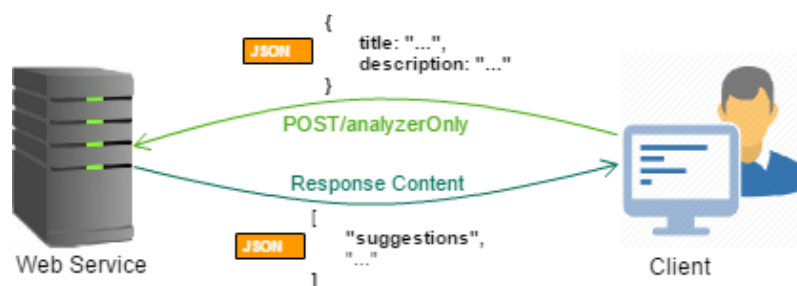
Nesse contexto, é proposta uma ferramenta para auxiliar iniciantes em programação a publicar, nos sites de perguntas e respostas sobre programação, perguntas que incluam as características consideradas como essenciais pela comunidade.

## 2. Desenvolvimento

O desenvolvimento do software, ora apresentado, pode ser dividido em duas etapas. A primeira foi o desenvolvimento de um Webservice, cuja principal função é buscar a presença ou ausência de “boas” características nas perguntas e oferecer sugestões que possam melhorá-las. Este deve ser útil para outros desenvolvedores possam implementar diferentes aplicações clientes para seu uso. A segunda etapa diz respeito ao desenvolvimento de um cliente para este Webservice, um aplicativo multiplataforma que tem como principal objetivo ajudar estudantes novatos a postarem perguntas com maior qualidade em Site de Perguntas e Respostas sobre programação.

### 2.1. Webservice

O Webservice foi desenvolvido usando a arquitetura REST (*Representation State Transfer*). De acordo com Cholia, Skinner & Boverhof (2010), o REST é um modelo arquitetural comum, usado para acessar recursos na web. Neste modelo, é formalizado um conjunto de boas práticas denominadas *constraints*; entre elas estão: a definição de um modelo cliente-servidor, um protocolo de comunicação sem estado e uma interface uniforme para todo o sistema. Em adicional, foi utilizado o Spring Framework<sup>1</sup> para criação do Webservice. O Spring é um *framework* de código aberto para criação de aplicações comerciais em Java. O Spring oferece suporte para criação de aplicações RESTful de forma ágil e fácil, reduzindo a complexidade em sua construção. A Figura 1 demonstra como o Webservice funciona.



**Figura 1 - Funcionamento do Webservice**

O Webservice “recebe” uma pergunta como entrada e retorna, como resultado, uma lista de sugestões para “melhorar” a pergunta. Dentro do Webservice, foi criada uma entidade com alguns métodos analisadores que recebem o título e corpo da pergunta. Cada analisador é responsável pela identificação de uma característica na pergunta, como, por exemplo, verificar se a pergunta está escrita de acordo com norma

<sup>1</sup> <https://projects.spring.io>

culta da língua portuguesa, se seu título é claro, objetivo e etc. Este modelo segue a arquitetura apresentada em (SOUZA *et al.*, 2015).

No que se refere à comunicação com o Webservice, para uma aplicação cliente realizar uma requisição via HTTP POST, ela deve enviar uma pergunta com o título e sua descrição no formato JSON (*Javascript Object Notation*) para o recurso.

Ao receber uma pergunta, o programa enfileira os “analísadores” que checam a presença ou ausência das características. Após este processo, o Webservice retorna uma lista de sugestões com possíveis melhorias para a pergunta, também no formato JSON.

Para a implementação dos analisadores que verificam a presença ou ausência das características foi tomado como base na lista de “boas” características apresentada em (REMIGIO *et al.* 2017). Nesta pesquisa, os autores identificaram uma lista contendo 13 características que aumentam a atratividade e qualidade de uma pergunta de programação. Essas características foram obtidas por meio de uma revisão da literatura e a aplicação de um questionário, onde se perguntou aos usuários de Sites de Perguntas e Respostas quais características eles esperavam em uma pergunta para que ela tivesse alta atrativa e qualidade. Este questionário obteve 400 respostas. Com base neste estudo, foram desenvolvidos 13 analisadores, cada um responsável por averiguar a presença ou ausências de uma dessas características.

Para verificar a acurácia dos analisadores, foi criado usado um *dataset* contendo 200 perguntas, extraídas do Stack Overflow em português<sup>2</sup>, metade com perguntas respondidas e metade com perguntas não respondidas. Em seguida, foram identificadas, por dois juizes, a presença e ausência de cada característica em cada pergunta. Na tabela 1, é apresentado os resultados da acurácia dos analisadores, com base no confronto das avaliações dos juizes (*ground truth*) e os analisadores.

**Tabela 1 - Acurácia dos analisadores**

<b>Característica</b>	<b>Perguntas Respondidas</b>	<b>Perguntas Não Respondidas</b>
Objetividade	88%	93%
Priorizar descrição curta	100%	99%
Clareza	82%	86%
Coerência entre título e descrição	97%	96%
Título bem definido	95%	97%
Descrição bem definida	82%	85%
Incluir exemplo ou trecho de código	95%	95%
Incluir links relacionado a pergunta	99%	99%
Combinar links com parte do conteúdo	100%	92%
Ser Educado	82%	91%
Evitar criar perguntas duplicadas	90%	92%
Evitar criar perguntas sobre trabalhos acadêmicos	100%	100%
Usar norma culta da língua	63%	26%

Na Tabela 1, fica evidente que a “pior” performance foi na característica *Usar norma culta da língua*. Para implementação deste analisador, foram utilizadas APIs de Processamento de Linguagem Natural (PLN) que verificam se a pergunta respeita as

<sup>2</sup> <https://pt.stackoverflow.com/>

normas de ortografia e concordância da língua portuguesa. Essas APIs possuem alta precisão e acabaram identificando erros que passaram despercebidos pelos juízes. Por essa razão, a discordância elevada entre os juízes e os analisadores automáticos. Contudo, foi considerado que o desempenho dos analisadores foi bastante satisfatório.

## 2.2. Web App

Com o objetivo de tornar o aplicativo acessível em qualquer plataforma, além de tornar a experiência de uso mais simples e engajável, foi desenvolvido um aplicativo Web, utilizando o conceito de *Progressive Web Apps*<sup>3</sup> (PWA). O aplicativo foi nomeado *Question's Advisor*.

Para construção do aplicativo foi utilizado o *framework JavaScript AngularJS*<sup>4</sup> e uma biblioteca de componentes Web denominada *Materialize*<sup>5</sup> para construção das páginas e acesso a recursos do Webservice. A escolha de tais tecnologias foi essencial para proporcionar melhor navegação em dispositivos móveis e *desktop*, por meio de uma *Single Page Application* e para melhor apresentação dos componentes nas páginas utilizando a especificação de componentes *Material Design* da Google. Isso torna a visualização do software mais semelhante entre os diferentes dispositivos utilizados no dia a dia, além de permitir a sua utilização por dispositivos diversos, com diferentes resoluções de telas e sistemas operacionais.

## 3. Apresentação do Software

O protótipo desenvolvido oferece, pelo menos, duas maneiras de utilização: via dispositivo móvel ou *desktop*, ambas acessíveis via browser. O sistema pode ser acessado no endereço <https://appif.herokuapp.com>. Para sua utilização, basta acessar a URL e realizar a autenticação por meio do Login com uma conta do Stack Overflow. Outra opção é se autenticar usando as credenciais do Facebook ou uma conta do Google. Este processo é necessário para que o usuário possa publicar perguntas no Stack Overflow usando o aplicativo. Na Figura 2, é apresentado o processo de primeira autenticação, no qual o usuário autoriza o aplicativo a publicar no Stack Overflow, usando sua conta na plataforma.

---

<sup>3</sup> <https://developers.google.com/web/progressive-web-apps/>

<sup>4</sup> <https://angularjs.org/>

<sup>5</sup> <http://materializecss.com/>

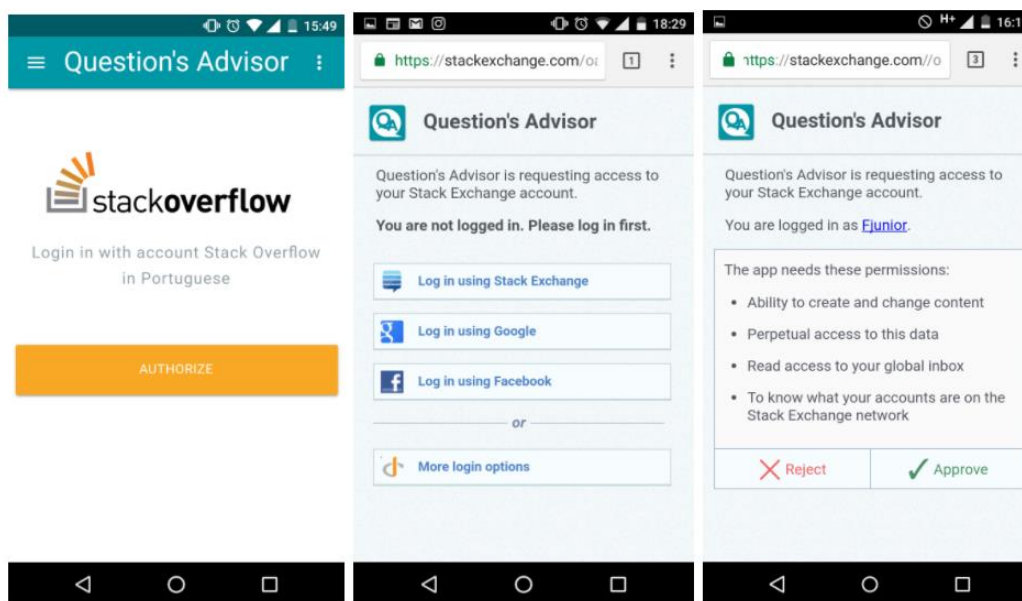


Figura 2 - Autorização no aplicativo

Após o processo de autenticação, o usuário é redirecionado para página inicial do aplicativo, onde poderá acompanhar todas as perguntas publicadas, tendo acesso ao número de respostas, número de visualizações e ao link de cada pergunta postada no Stack Overflow. Na Figura 3, é mostrada essa página, contendo nenhuma pergunta e lista (esquerda) e algumas perguntas listadas (direita).

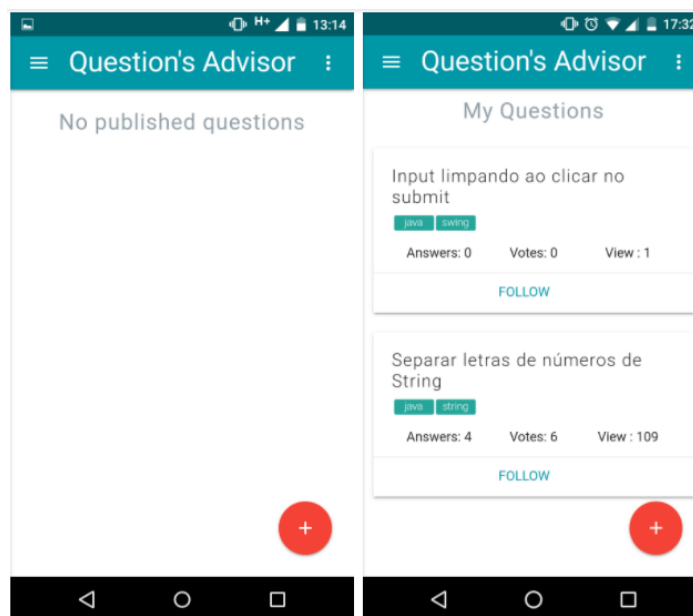


Figura 3 - Página Principal do aplicativo

Ao clicar no botão flutuante (no canto inferior), o usuário é direcionado para página de criação de uma nova pergunta. Durante a formulação da pergunta, o usuário pode adicionar trechos de código ao corpo da descrição, os quais serão mostrados com um estilo diferente. Além disso, o usuário deve adicionar *tags* relacionadas ao assunto

da pergunta, para publicação no Stack Overflow. Na Figura 4, é apresentada a página de criação de uma nova pergunta.

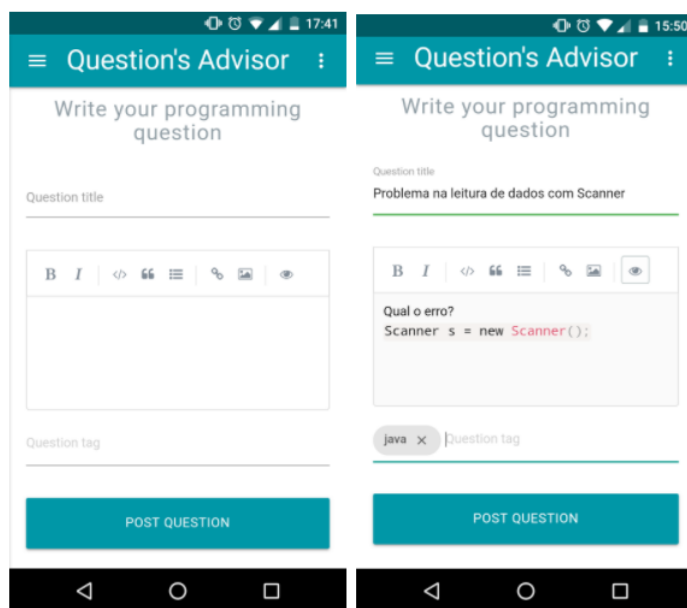


Figura 4 - Página para escrever uma nova pergunta

Após escrever a pergunta e clicar em postar, será exibida uma caixa de diálogo com sugestões de como melhorar a pergunta. Caso o usuário ache necessário, o mesmo terá a chance de melhorar sua pergunta com as dicas apresentadas ou ainda ignorá-las e postar. Entretanto, esse processo é opcional e cabe a ele escolher aplicar ou não as sugestões. Na Figura 5, está representado o processo em que a pergunta é analisada (esquerda) e as dicas são apresentadas (centro), além da confirmação de publicação da pergunta (direita).

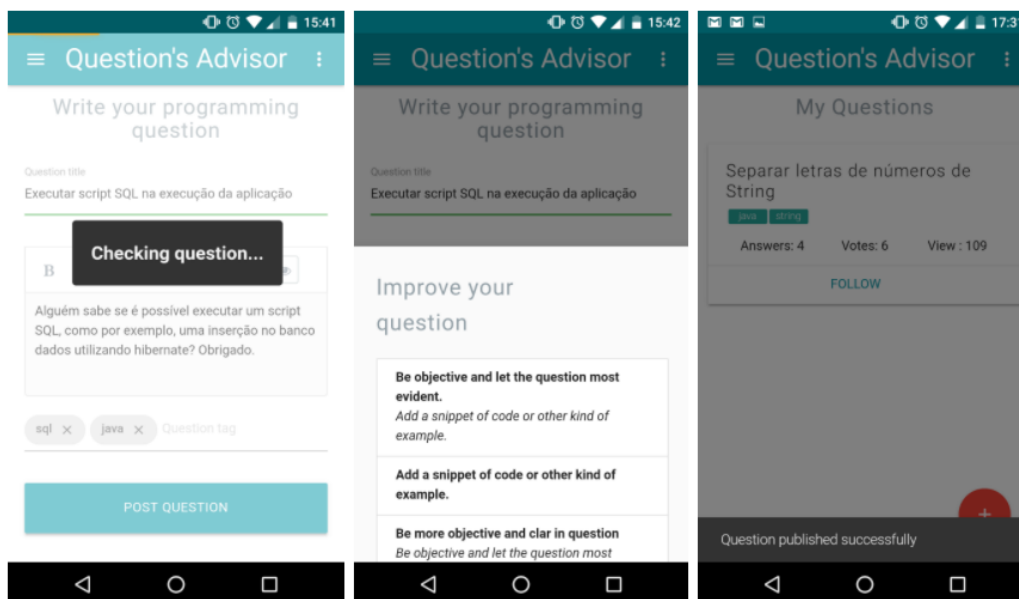


Figura 5 - Publicação de uma pergunta

As sugestões mostradas para usuário são processadas pelo o Webservice utilizando Processamento de Linguagem Natural (PLN) sobre a pergunta, que identifica a presença ou ausência de “boas” características. Foram utilizadas as seguintes bibliotecas de PLN: CoGroo<sup>6</sup>, OpenNLP<sup>7</sup> e Language Tool<sup>8</sup>.

#### 4. Considerações finais

Alguns trabalhos já haviam citado como proposta de trabalhos futuros uma ferramenta que ofereça sugestões de reescrita para a pergunta de programação (DROR *et al.*, 2013 & SHAH *et al.*, 2014).

Neste trabalho, foi apresentada uma ferramenta que assiste ao usuário na formulação de sua pergunta, oferecendo *feedback* qualitativo e ajudando-o a “melhorá-la”. Com isso, os novatos em programação poderão escrever perguntas com qualidade e atratividade maiores, e conseqüentemente aumentar suas chances de obter resposta.

Para projetos futuros, planeja-se finalizar o aplicativo para compartilhamento de pergunta em mais Sites de Perguntas e Respostas, aprimorar os analisadores do Webservice e avaliar o uso da ferramenta utilizando algum estudo de caso. Além disso, pretende-se ampliar o projeto para mais áreas do conhecimento.

#### Agradecimentos

Queremos agradecer ao IFPB – Campus Monteiro pelo apoio a essa pesquisa.

#### Referências

Sociedade Brasileira de Computação - SBC (2003) “Currículo de Referência da Sociedade Brasileira de Computação para Cursos de Graduação em Computação e Informática”. Disponível Em: <<http://www.sbc.org.br/documentos-da-sbc/category/131-curriculos-de-referencia>>. Acesso: Dez. 2016.

Vahldick, A. (2007) “Uma experiência lúdica no ensino de programação orientada a objetos. In: I Workshop de Ambientes de Apoio à Aprendizagem de Algoritmos e Programação”. Simpósio Brasileiro de Informática na Educação.

Mota, M. (2009) “Ambiente integrado a plataforma moodle para apoio ao desenvolvimento das habilidades iniciais de programação”. Anais do Simpósio Brasileiro de Informática na Educação.

Remígio, J. and Aragão, F. and Souza, C. (2016) “Investigando as Características Desejáveis em Perguntas de Programação”. XI Congresso Norte e Nordeste de Pesquisa e Inovação.

---

<sup>6</sup> <http://cogroo.sourceforge.net/>

<sup>7</sup> <https://opennlp.apache.org/>

<sup>8</sup> <https://www.languagetool.org/>

Stackexchange; 2016. Stack Exchange Network Statistics. Disponível em:  
<<http://stackexchange.com/sites>>. Acesso em: 12 dez. 2016.

Sullins, J., McNamara, D. S., Acuff, S., Neely, D., Hildebrand, E., Stewart, G., and Hu, X. (2015) "Are You Asking the Right Questions: The Use of Animated Agents to Teach Learners to Become Better Question Askers." FLAIRS Conference.

Asaduzzaman, M., Mashiyat, A., Roy, C. & Schneider, K. (2013) "Answering questions about unanswered questions of Stack Overflow". 10th Working Conference on Mining Software Repositories (MSR), p. 97–100.

Chua, A., Banerjee, S. (2015) "Answers or no answers: Studying question answerability in Stack Overflow". *Journal of Information Science*, 41(5), 720–731.

Hao, G., Shu, Z., Irawan, J. (2014) "Good or Bad Question? A Study of Programming CQA in Stack Overflow." Nanyang Technological University.

Cholia, S., Skinner, D., and Boverhof, J. (2010) "NEWT: A RESTful service for building High Performance Computing web applications". *Proc. Gateway Computing Environments Workshop (GCE)*, pp. 1-11.

Souza, C. Maia, J., Magalhães, J., Barros, H., Costa, E., Fechine, J. (2015) "A Meta-Information Extractor for Interrogative Sentences". *Proc. of the International Conference on Computational Science and Its Applications (ICCSA)*, v. 9155, p. 345-354.

Remígio, J. Aragão, F., Souza, C., Costa, E., Fechine, J. (2017) "Question's Advisor - A Wizard Interface to Teach Novice Programmers How to Post "Better" Questions in Stack Overflow". In *Proceedings of the 19th International Conference on Enterprise Information Systems - Volume 1: ICEIS*, p. 471-478.

Dror, G., Maarek, Y. SzpektorWill, I., (2013) My Question Be Answered? Predicting "Question Answerability" in Community Question-Answering Sites. *Machine Learning and Knowledge Discovery in Databases*, Prague, v. 8190, p. 499–514.

Shah, C., Kitzie, V. Choi, E. (2014) "Questioning the question - Addressing the answerability of questions in community question-answering." *Proceedings of the Annual Hawaii International Conference on System Sciences*, 1386–1395.