

# Proposta de um Ambiente Online para a Resolução e Correção Automática de Algoritmo

Idovaldo Cunha da Silva

Luís Carlos Costa Fonseca

Reinaldo de Jesus da Silva

Universidade Estadual do Maranhão  
Cidade Universitária Paulo VI, s/n -  
Tirirical, São Luís - MA, 65055-000  
+55 99 982344968  
idovaldo@gmail.com

Universidade Estadual do Maranhão  
Cidade Universitária Paulo VI, s/n -  
Tirirical, São Luís - MA, 65055-000  
+55 98 981112222  
lccfonseca@gmail.com

Universidade Estadual do Maranhão  
Cidade Universitária Paulo VI, s/n -  
Tirirical, São Luís - MA, 65055-000  
+55 98 981007302  
reinaldo.silvrrrb@gmail.com

## ABSTRACT

In order to contribute to the teaching and learning process in the program logic of discipline, this work aims to develop an authoring tool and application testing with automatic correction. To this end, we will integrate the tool with a visual editor code and BOCA environment, which is a system used for managing computer programming competitions. The development of media based search tool pedagogically provide the theoretical and methodological assumptions of the theory Historical-Cultural for the realization of programming logic of evidence in the replies to be algorithms that are automatically corrected, generating an immediate return to the student

## RESUMO

Com o intuito de contribuir no processo de ensino e aprendizagem na disciplina de lógica de programação, este trabalho visa desenvolver uma ferramenta de autoria e aplicação de prova com correção automática. Para tanto, integraremos a ferramenta a um editor visual de código ao ambiente BOCA, que é um sistema utilizado para gerenciar competições de programação de computadores. O desenvolvimento da ferramenta busca propiciar meios fundamentados pedagogicamente nos pressupostos teórico-metodológicos da Teoria Histórico-Cultural, para a realização de provas de lógica de programação em que as respostas sejam algoritmos que serão corrigidos automaticamente, gerando um retorno imediato para o aluno.

## Palavras Chaves

Lógica de programação. Ambiente de Avaliação. Integração. Teoria Histórico-Cultura.

## 1. INTRODUÇÃO

Hoje, as ciências e a tecnologia estão cada vez mais aliadas no processo de ensino e aprendizagem. Nessa perspectiva, podemos fazer uso de várias ferramentas tecnológicas como jogos e simuladores, por exemplo, nas mais diversas disciplinas, como: matemática, química, biologia, física, informática e etc, afim de

desenvolver o pensamento criativo e despertar processos psicológicos superiores dos seus usuários, com o intuito de aumentar seu grau de memorização, concentração e o desenvolvimento de seu raciocínio lógico.

Nesta perspectiva, o presente trabalho visa contribuir com o processo de ensino e aprendizagem na disciplina de lógica de programação, através da proposta de implementação de uma ferramenta de autoria e aplicação de prova a alunos dos cursos de computação, além da integração entre duas ferramentas distintas, o Blockly e BOCA, ao sistema que será desenvolvido.

A disciplina de lógica de programação é normalmente ensinada nos cursos da área de informática. Geralmente, nos cursos de computação, o ensino de lógica de programação se utiliza de técnicas, tais como: fluxograma ou a aplicação da teoria ensinada em sala de aula, através de uma linguagem de programação específica, como C, C++, Java, JavaScript, Pascal e etc, porém, mais recentemente surgiram as linguagens visuais que tem sido muito utilizada para desenvolvimento de aplicativos educacionais para o ensino de lógica de programação.

No ensino e aprendizagem de lógica de programação uma das dificuldades dos alunos é a execução da sua lógica depois de desenvolvidos seus algoritmos, tendo em vista que os mesmo são escritos em papel e que, segundo [5], o teste de mesa, que é aplicado nessa etapa da disciplina e que é um método utilizado para verificar se o algoritmo desenvolvido é eficaz, deve seguir um passo a passo, e isso além de desmotivar o aluno traz consigo um outro problema, que seria o ato de seguir o passo a passo para verificar a funcionalidade de sua solução.

E afim de alcançar o desenvolvimento de uma ferramenta que supra as necessidades pedagógicas referente ao processo de ensino e aprendizagem, utilizaremos em sua concepção, a Teoria Histórico-Cultural. Neste contexto, iremos buscar suporte teórico pedagógico nas categorias de desenvolvimento e aprendizagem, mediação e atividade, postuladas como sendo de fundamental importância para a apropriação do saber e para o desenvolvimento dos processos psicológicos superiores humano.

Frente ao exposto, o presente trabalho tem por objetivo desenvolver uma ferramenta de autoria e aplicação de prova, e a integração dos ambientes, Blockly e BOCA. O sistema como um todo, foi concebido pedagogicamente para realizar avaliações automáticas de lógica de programação, e possibilitar a criação, correção e reuso de questões e provas, que se encontram cadastrados na base de dados do sistema, pelo professor, e gerando um retorno imediato para o aluno a cada submissão de questão.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '15, Month 1–2, 2015, City, State, Country.

Copyright 2015 ACM 1-58113-000-0/00/0010 ...\$15.00.

## 2. REFERENCIAL TEÓRICO

Serão apresentados nessa seção alguns temas que darão fundamento teórico metodológico para o desenvolvimento do ambiente *online* para a resolução e correção automática de avaliação de algoritmo na disciplina de lógica de programação.

### 2.1. Dificuldade no ensino e aprendizagem de lógica de programação

Nos cursos da área de computação as disciplinas relacionadas com a programação e ensino e aprendizagem de algoritmos ocupam ponto de destaque e são essenciais para o bom desempenho dos alunos no decorrer dos cursos e na maioria das carreiras relacionadas com a informática.

Pode-se observar nos cursos que abordam disciplinas que trabalham com lógica de programação, uma certa falta de interesse dos alunos em relação ao seu aprendizado, e isso se deve ao fato dessa disciplina ter como ponto fundamental uma grande carga de conceitos abstratos que é encontrado em todo o processo de ensino e aprendizagem nas disciplinas relacionadas a área de programação de computadores, onde a sofisticação dos ambientes de programação, principalmente das linguagens de programação, e da máquina em si, tendem a atrapalhar a elaboração da lógica dos alunos. Assim, para minimizar a falta de interesse dos alunos uma boa alternativa seria colocar para os iniciantes nessa disciplina, ambientes com uma boa carga de simplicidade [1].

[7], destaca alguns fatores que contribuem para a dificuldade no processo de ensino e aprendizagem de lógica de programação, dentre outras:

- algoritmos e programas computacionais são processos dinâmicos, isso implica que os métodos tradicionais de ensino dessa disciplina não são mais adequados.
- a impossibilidade de o professor fazer um acompanhamento individualizado ao aluno, tendo em vista o grande número de egressos que formam o primeiro semestre dos cursos de computação e ao tempo curto do currículo escolar.
- as linguagens de programação usuais apresentam uma grande sobrecarga de sintaxes complexas e não têm representações visuais dos algoritmos, o que não contribui para uma melhor compreensão;

Nessa perspectiva, os fatores citados geram frequentemente desorientação e desinteresses por parte de muitos dos alunos, exigindo um acompanhamento e orientação que dificilmente o professor terá para disponibilizar e que os métodos tradicionais nem sempre dão conta de suprir [7].

Essas dificuldades têm levado educadores a procurar estratégias e matérias que possam diminuir as dificuldades sentidas por alunos e professores de programação. Para isso faz-se uso de programas educativos, tendo em vista que eles apresentam características dinâmicas e que sugere que as operações e interações dos algoritmos sejam, em geral, melhor descritas por meio de representações visuais dinâmicas [7].

Conforme [9], por as disciplinas de lógica de programação serem umas das primeiras a serem oferecidas nos cursos de computação, geralmente são formadas por alunos que vem de uma outra realidade, alunos oriundos do ensino médio, e com um histórico escolar de repetência e déficit de aprendizagem, dentre outros problemas. Assim, forma-se grande número de alunos

matriculados nessas disciplinas, dificultado um acompanhamento e atendimento individualizado por parte do professor.

### 2.2. Técnicas utilizadas para o ensino de lógica de programação

Lógica de programação pode ser definida como um conjunto de técnicas para encadear pensamentos a fim de atingir determinado objetivo. Para a construção de algoritmos a lógica de programação é essencial, pois as pessoas que desejam trabalhar com desenvolvimento de programas para computadores têm que se apropriar de tais conceitos e conhecimentos.

Para o desenvolvimento de algoritmos temos que usar essencialmente a lógica, para isso, faz-se usos de algumas linguagens, tais como: linguagem de programação de auto nível, fluxograma e linguagens de programação visual (bloco), por exemplo.

As linguagens de programação de auto nível fazem uso de uma infinidade de palavras adicionais, como “program”, “var”, “begin”, “end” que pode, muitas vezes, acarretar mais um problema para os alunos no aprendizado da disciplina. Como mostrado na Figura 1.

```

Program FATORIAL;
var
  I, N, FAT: integer;
Begin
  Writeln('Digite um número:');
  readln (N);
  FAT := N;
  For i:=1 to (N-1) do
    FAT := FAT * I;
  writeln('O fatorial de ', N, ' equivale a: ', FAT);
end;

```

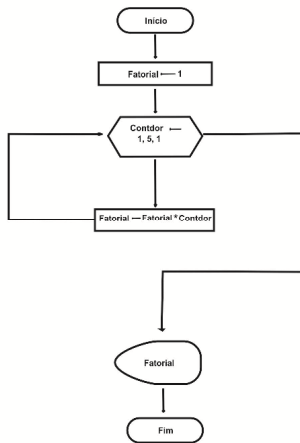
Figura 1. Código na linguagem pascal

Todos essas palavras e caracteres são necessários para que o programa seja entendido pelo compilador, que é um programa ou conjuntos de programas que tem por objetivo traduzir programa escrito em uma linguagem de alto nível, mais próximas da linguagem humana, programa fonte, como Pascal, Java, C, C++ e etc, para uma linguagem de máquina, ou baixo nível, como o Assembly. Esta tradução envolve a análise sintática, a qual tem por objetivo verificar se o programa está escrito dentro das regras da linguagem de programação do compilador [10].

Frente ao exposto, fica claro que todo esse excesso e sobrecarga de sintaxe das linguagens de programação tornam-se uma barreira a mais no processo de ensino e aprendizagem da disciplina de lógica de programação, pois, além de se preocupar com o desenvolvimento do raciocínio o aluno ainda tem que se preocupar com a assimilação da linguagem. Esse é outro ponto que tem que ser levado em consideração quando se fala sobre o ensino e aprendizagem na disciplina de lógica de programação.

O fluxograma, por outro lado, que é usado para mostrar de forma gráfica, a lógica para resolver programas, sendo que nesse processo destaca-se passos individuais e o fluxo de execução [10]. Assim, o fluxograma é nada mais do que uma representação gráfica do algoritmo, através de formas geométricas, facilitando a compreensão da lógica utilizada pelo aluno.

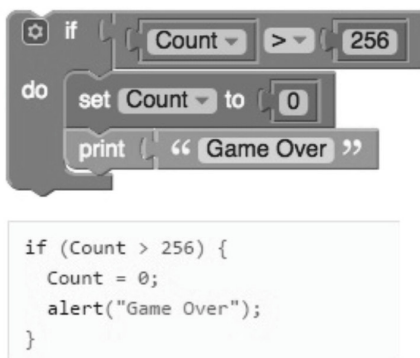
A Figura 2 ilustra os passos usados no desenvolvimento da lógica de programação para a resolução de problema para encontrar o fatorial de 5.



**Figura 2. Fluxograma: cálculo do fatorial**

Porém, computadores não são programados usando desenhos. Existem, no entanto, algumas linguagens gráficas, mas são raras e pouco utilizadas [10].

A Figura 3 mostra um código em linguagem visual e logo a baixo o corresponde, exportado em JavaScript.



**Figura 3. Código na linguagem visual em blocos**

A linguagem visual em blocos, que consiste na criação de códigos onde o aluno pode arrastar e soltar blocos para escrever programas, mostra-se mais interessante, atraente e eficiente quando a finalidade é o foco na lógica desenvolvida pelo aluno, pois o aluno não precisará se preocupar com a sintaxe da linguagem quando estiver resolvendo suas atividades. A linguagem visual consiste em que cada bloco corresponde a uma linha de código em C, C++, Java, Python dentre outras, que depois de desenvolvido o código, ele pode ser exportado em outra linguagem.

Tendo em vista as características da linguagem visual e o enfoque da disciplina de lógica de programação, que está mais interessada na forma conceitual de organização das ideias, do que no aprendizado de sintaxe de linguagem, que optamos para fazer uso da linguagem visual nesse trabalho.

### 2.3. Teoria histórico-cultural como aposte teórico metodológico

Com enfoque no processo de ensino e de aprendizagem humano, lançaremos mão dos pressupostos teóricos metodológicos da Escola de Vygotsky - Teoria Histórico-Cultural - e demais colaboradores, para fundamentar a proposta de desenvolvimento do ambiente *online* para a resolução e correção automática de avaliação de algoritmo, a fim de desenvolver uma sistema voltada pedagogicamente para o desenvolvimento humano e que seja capaz de auxiliar professor e aluno no processo de ensino e aprendizagem na disciplina de lógica de programação.

Vygotsky foi um psicólogo Bielo-Russo, com formação em medicina e direito, dentre outras, que se interessava por literatura e por entender o processo psicológico do desenvolvimento humano [8]. A sua influência é sentida até hoje em diversos campos de pesquisa, dentre eles a arte, a literatura, a psicologia, a antropologia, a neurologia, a linguística e a educação. Mesmo depois de quase um século da realização das suas pesquisas, seus estudos ainda servem como base para diversas trabalhos no campo da linguagem e educação, entre outros.

Dentre as categorias da Teoria Histórico-Cultural, que utilizaremos para concepção da proposta de desenvolvimento do ambiente *online* para a resolução e correção automática de avaliação de algoritmo, recorremos aos eixos norteadores: desenvolvimento e aprendizagem, mediação e atividade, vistas como pontos centrais dentro desta teoria e fundamentais para o processo de apropriação de conhecimento e desenvolvimento psicológico humano.

Conforme [8], o desenvolvimento humano, a aprendizagem e a relação entre eles enfatiza a existência de uma trajetória de desenvolvimento que ocorre em parte definida pelo processo de maturação do organismo do indivíduo, pertencente a espécie humana, mas que o aprendizado é quem possibilita o despertar de processos internos de desenvolvimento, que ocorrem através do contato do indivíduo em certo ambiente cultural.

Nesse sentido, entendemos que o desenvolvimento das atividades psicológico superiores humana ocorre em consequência da apropriação da cultura a partir de seu contato e interação com a realidade e com o meio que o rodeia. Segundo [3], através da apropriação desses conhecimentos o homem transforma a natureza e a constitui em objeto de conhecimento e, ao mesmo tempo, transforma a si mesmo em sujeito do conhecimento, assim entendemos como sendo essencial no desenvolvimento da ferramenta aspectos pedagógicos a fim de proporcionar maior aprendizado por parte dos alunos.

A mediação para [11] é a base dos processos psíquicos superiores. Portanto, é de suma importância para desenvolvimento humano e tem uma enorme relevância nos processos pedagógicos, tendo em vista a relação que ocorre entre o sujeito que ensina – o mediador –, e o sujeito que aprende – o receptor.

A relação entre o homem e o mundo é uma relação mediada, sendo essa relação essencial para o processo de transmissão das atividades humanas. [3] diz que, segundo a concepção de Vygotsky, o professor não ensina conceitos aos alunos, pode-se, no máximo, apresentar definições de conceitos que serão reproduzidos pelos alunos. Na realidade são os próprios alunos que formaram seus conceitos e o professor é apenas um mediador nesse processo.

A atividade não é qualquer coisa que a pessoa faça, mas apenas aquilo que faz sentido para ela. A tarefa realizada pelo ser humano sempre tem um objetivo e motivo, o objetivo é o resultado esperado no final da tarefa, que o mesmo já é previsto como uma ideia, o motivo é a necessidade que o levou a querer atingir aquele objetivo. O sentido é dado pela relação entre o motivo e o objetivo – ou resultado – previsto para a tarefa [6].

Assim, a relação entre estímulos e respostas é mediada pela atividade do indivíduo. Entendemos ainda que é necessário um certo grau de comprometimento do indivíduo no desenvolvimento de uma atividade. Para que ocorra o aprendizado, é necessário, por parte da pessoa que está realizando a tarefa, um interesse naquilo que está sendo feito, no motivo que a impulsionou a realizar a atividade e até mesmo o que se espera no final da tarefa.

### 3. AMBIENTES ENVOLVIDOS E TABALHOS REALACIONADOS

#### 3.1. Os ambientes Blockly e BOCA

O Blockly é uma ferramenta de programação que faz uso de uma linguagem visual, permitindo aos usuários escrever códigos conectando blocos. O *software* é uma API (*Application Programming Interface*) disponibilizada pela Google, gratuitamente, afim de facilitar a criação e desenvolvimento de apps (*application*), procurando evitar que o desenvolvedor precise decorar comandos, e se preocupar com uma sobrecarga de sintaxe específica de uma linguagem de programação. Esta ferramenta está sendo usada largamente para desenvolvimento de aplicações educacionais para auxiliar o processo de ensino e aprendizagem sobre lógica de programação, devido a sua interface amigável, intuitiva e lúdica.

O Blockly foi adotado por se tratar de um *software Open Source* e oferece permissões de cópia, modificação e uso, contanto que seja mantida a licença original, os direitos autorais e o fornecimento do código-fonte aos demais usuários.

O Ambiente BOCA é um sistema de administração de competições de programação e foi desenvolvido para ser usado nas maratonas de programação nos moldes das realizadas pela SBC (Sociedade Brasileira de Computação). O sistema BOCA foi criado por [2] da EACH-USP (Escola de Artes, Ciências e Humanidades da USP). O sistema foi desenvolvido em uma interface web e serve para gerenciar provas ou exercícios de programação, possui controle de tempo e disponibilização de resultados em tempo real durante as competições. O BOCA foi programado em linguagem PHP - (*Hypertext Preprocessor*), portanto, é portátil para todo sistema onde tal linguagem esteja disponível.

Porém, usuários que não possuem suficiente conhecimento sobre linguagem de programação ou mesmo do funcionamento do compilador, que é o caso da maioria dos alunos que estão iniciando nos cursos da área que envolve programação de computadores, não compreendem facilmente as informações apresentadas pelo sistema, sendo necessário reavaliar a forma e o conteúdo da apresentação dos erros aos alunos em função de seu nível de conhecimento.

#### 3.2. Trabalhos relacionados

Na busca por trabalhos que se utiliza da integração entre ambientes com a finalidade de dar suporte as atividades com

correção automática de exercícios de algoritmos, encontramos dois trabalhos que chamaram atenção e se destacaram, sendo um proposto por [4], e o outro trabalho realizado por [5], ambos propondo a integração do Ambiente Virtual Moodle, que é uma plataforma de ensino muito disseminada e utilizada no Brasil na modalidade de EAD (Educação a Distância), ao ambiente BOCA.

A proposta de [4] procurou contribuir com as condições de ensino e aprendizagem em laboratórios de disciplinas de programação, o trabalho apresentou um ambiente que integra o Moodle a ferramenta *Web BOCA*, utilizada no apoio a competições de programação. A ferramenta permite o registro do resultado da compilação e da execução de problemas propostos nas linguagens C, C++ e Java, rastreando as submissões dos alunos, e possibilita a avaliação do professor via interface do Moodle. Considerando que a compilação e execução de programas em um servidor remoto compartilhado pode requerer muitos recursos computacionais, o ambiente desenvolvido oferece suporte a balanceamento de carga.

E o segundo trabalho foi realizado por [5], que buscou integrar o ambiente Moodle ao ambiente BOCA para realizar a avaliação automática de questão enviadas por alunos. A solução apresentada buscou não alterar características do Moodle, permitindo uma implantação mais simples. A solução também foi desenvolvida para que os ambientes fossem implementados em servidores independentes já que a utilização do BOCA durante a avaliação dos algoritmos poderia comprometer o desempenho do ambiente Moodle.

Embora estes trabalhos citados tragam importantes contribuições para o ensino de algoritmos, eles se diferenciam da proposto aqui exposta, pois pretendemos implementar uma ferramenta de autoria e aplicação de prova e integrar os ambientes com a finalidade de ensino e aprendizagem na disciplina de lógica de programação, sem a pretensão de se trabalhar com uma linguagem de programação que possui um enorme sobrecarga de caracteres, preocupando-se somente com o desenvolvimento das ideias de forma lógica do alunos, e assim trabalhar em um outro contexto, tendo em vista que iremos utilizar nessa perspectiva uma linguagem visual de blocos para codificação.

Entendemos que para o ensino de lógica de programação o uso de métodos de uma linguagem de programação já existente torna-se uma barreira a mais, tendo em vista que as linguagens de programação estão mais focadas em aspectos computacionais, enquanto que a lógica de programação está mais voltar para a forma conceitual de organização de ideias sobre a realização de determinadas tarefas. Fazendo disso um dos motivos suficientes para que não sejam trabalhadas juntas.

#### 3.3. Arquitetura de desenvolvimento

O presente trabalho tem como principal finalidade o apoiar ao processo de ensino e aprendizagem na disciplina de lógica de programação. Com esse intuito, a proposta de desenvolvimento do ambiente *online* para a resolução e correção automática de avaliação de algoritmo e a integração entre os *softwares* Blockly e BOCA, foi pensada com o objetivo de potenciar uma participação ativa do aluno no processo de ensino e aprendizagem, permitindo a prática e fixação do conteúdo trabalhado em aula, e da correção automática dos algoritmos durante essa resolução.

É também objetivo deste sistema dar ao professor a possibilidade de gerar avaliações ou atividades - que poderão ser geradas escolhendo-se por categoria ou ser geradas aleatoriamente -



cadastrear questões, excluir questões, visualizar os resultados dos problemas resolvidos e alterar nota no sistema. O professor poderá, ainda, fazer o devido acompanhamento de seus alunos individualmente, onde terá a possibilidade de gerar relatórios de desempenho individual dos alunos, pois será criada uma base de dados para armazenar as informações geradas pelos alunos no decorrer da resolução das atividades, a fim de fazer comparações e observar o rendimento individual e da turma. Vale ressaltar que o professor terá acesso aos códigos submetidos pelos alunos, e as atividades ficam armazenadas para reutilização posterior.

A seguir a Figura 4 mostra a arquitetura de integração interna entre o ambiente *online* para a resolução e correção automática de avaliação de algoritmos, o editor visual de código e o sistema BOCA:

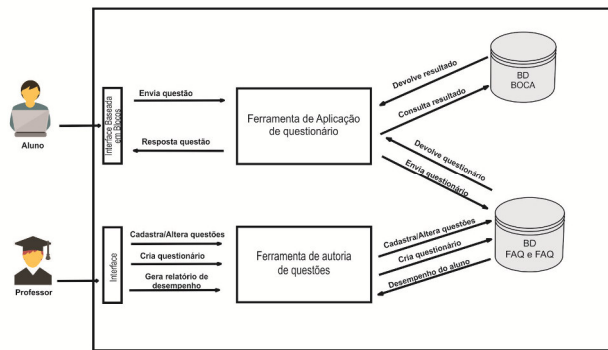


Figura 4. Arquitetura interna do Sistema

Para a criação das atividades, o professor cadastra na base de dados, através de uma interface, que será desenvolvida especificamente para tal finalidade, o título da questão, o enunciado do problema, um conjunto de entradas esperadas e um conjunto de saídas específicas, não sendo necessário cadastrar uma linguagem de programação específica, pois o editor trabalha com uma linguagem visual, porém o mesmo gerar um código corresponde a lógica desenvolvida em uma outra linguagem específica.

Contudo, destacamos uma dificuldade, pois o sistema BOCA dá suporte a, somente, códigos em C, C++ e Java, enquanto a ferramenta Blockly gera códigos em JavaScript, Payton, PHP e Dart, assim será necessário fazermos uma conversão da linguagem visual para uma das linguagens correspondentes aceitas pelo sistema BOCA, afim de que seja realizada as correções dos problemas, a princípio implementaremos a ferramenta para geração de códigos em linguagem C++.

Para a prática dos alunos, suas lógicas serão desenvolvidas na linguagem visual utilizada pelo editor Blockly e o ambiente *online* para a resolução e correção automática de avaliação de algoritmo, enviará o algoritmo para a avaliação do sistema BOCA, em linguagem C++, exportada pelo sistema, possibilitando aos alunos um retorno automático sobre a resolução de sua atividade. O sistema BOCA dá um retorno podendo ser positivo ou negativo.

Para um melhor entendimento, podemos observar na figura 5 o fluxo do processo entre o ambiente *online* e os ambientes Blockly e BOCA, tais processos servem para o envio de questões e recebimentos das possíveis respostas.

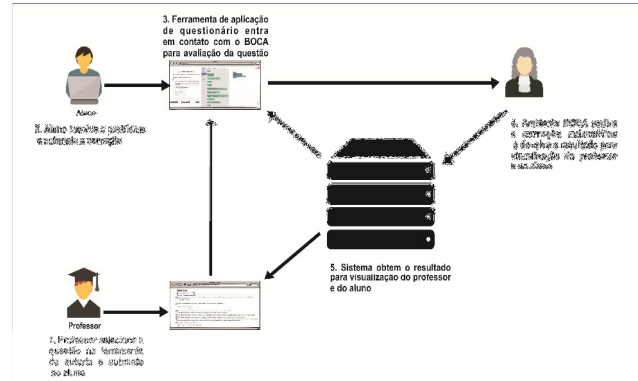


Figura 5. Fluxo do processo do sistema com os ambientes

Processos: 1 – o professor, logado no ambiente, define a avaliação ou atividade e a submete para resolução; 2 – o aluno visualiza a avaliação ou atividade definida pelo professor, em uma interface desenvolvida para o acesso do aluno, desenvolve e submete uma solução para a avaliação; 3 – o sistema editor de código entra em contato com o ambiente BOCA e envia o código-fonte, em C++, para avaliação; 4 – o ambiente BOCA realiza os devidos processos de avaliação automática para a solução proposta e devolve o resultado; e 5 – o sistema obtém o resultado da avaliação e o disponibiliza para visualização pelo aluno e pelo professor.

O sistema será composto por duas interfaces, uma exclusiva para o uso do professor e outra desenvolvida especificamente para os alunos, cada uma com funções para a finalidade ao qual se propõe.

### 3.3.1. Tela do professor

Devidamente logado no sistema o professor poderá criar avaliações e ou atividades e, posteriormente, editar ou remove-las. Na tela do professor, será mostrado também, uma lista com opções de tipo de questão para o professor escolher qual o nível de questões que será submetido em cada avaliação ou atividade, onde o mesmo terá também a opção de gerar avaliações e ou atividades aleatoriamente com questões já cadastradas na base de dados do ambiente. Após a escolha do tipo de questão será mostrado para o professor outra tela com os campos necessários para cada tipo de questão.

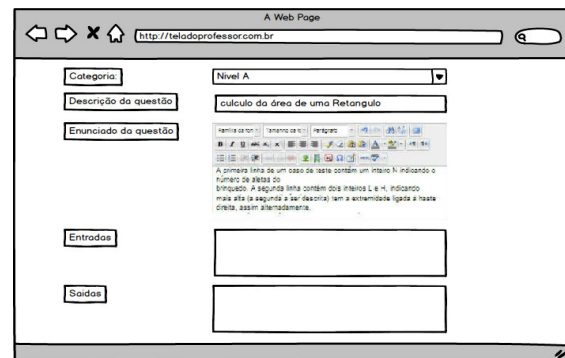


Figura 6. A Tela de cadastro de questões do professor

A Figura 6, mostra abaixo a tela para criação de questões, nessa interface o professor deve inserir o título da questão, o enunciado

da questão, um conjunto de entrada e um conjunto de saída padrão. Para julgar as questões, o sistema BOCA necessita de um conjunto de entradas e um conjunto de saída padrão, em formato .zip, contendo também algumas informações referente as questões.

### 3.3.2. Tela do aluno

A tela do aluno apresenta o problema, mostrando o título do problema, o corpo do enunciado contendo exemplos de entrada e saída que o programa deverá gerar, além disso existirá uma área para que o aluno possa arrastar os blocos para desenvolver seu algoritmo.

Para que o sistema BOCA possa julgar a solução submetida pelo aluno, o sistema espera um arquivo contendo o código fonte gerado pelo editor de código em linguagem visual. Deixamos aqui especificado que o sistema possibilitará ao aluno que ele escolha a ordem de resolução das questões presentes no questionário, onde o questionário terá um tempo específico para ser resolvido pelo aluno. A Figura 7 mostra a tela do aluno para resolução de questão.

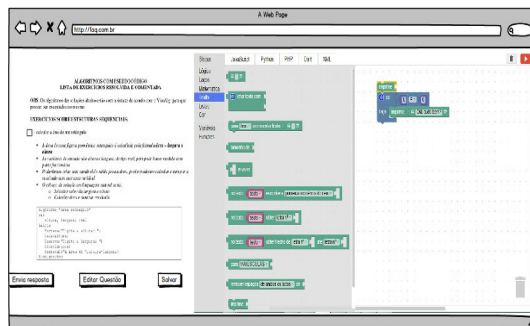


Figura 7. Tela de resolução de questionário do aluno.

## 4. RESULTADOS ESPERADOS

Durante ao longo do tempo têm sido pensadas e criadas diversas ferramentas para o auxiliar professores e alunos no processo de mediação do conhecimento. Nesta concepção, o presente trabalho visa contribuir para o processo de ensino e aprendizagem na disciplina de lógica de programação dos cursos da área de informática. Nesse sentido, propomos um ambiente *online* para a resolução e correção automática de avaliação de algoritmos, apoiado em princípios pedagógicos postulados pela teoria histórico-cultural como essenciais para o processo de desenvolvimento humano, afim de contribuir de forma significativa para o processo de mediação do conhecimento.

Tendo em vista tais dificuldades no processo de ensino e aprendizagem dessa disciplina, a proposta de desenvolvimento do ambiente apresenta-se neste trabalho como sendo de fundamental importância para uma estruturação fundamentada pedagogicamente para o processo de desenvolvimento das máximas potencialidade do psicológico humano. A Proposta de desenvolvimento do ambiente *online* para a resolução e correção automática de avaliação de algoritmo será desenvolvida e, posteriormente, será realizar testes para verificar o seu uso junto ao seu público alvo (professores e alunos da disciplina de lógica de programação). Com o fim dos testes, pretende-se, para uma melhor avaliação do desempenho da ferramenta junto a professores e alunos, disponibilizar a ferramenta em uma turma

regular na disciplina de lógica de programação, com o objetivo de verificar a aplicação prática da ferramenta e possíveis ajustes a serem realizados.

## 5. REFERÊNCIAS

- [1] ALMEIDA, Eliane. S., Costa, Evandro B., Silva, Klebson dos S., Paes, Rodrigo B., Almeida, André Atanasio M.; Braga, Julian D. Herrera. AMBAP: Um Ambiente de Apoio ao Aprendizado de Programação. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2002/006.pdf>>. Acesso em: 02 mar. De 2015
- [2] CAMPOS, Cássio P de; Ferreira, Carlos E. BOCA: um sistema de apoio a competições de programação. Disponível em: <<http://www.ime.usp.br/~cassio/boca/campos-ferreira-wei2004.pdf>>. Acesso em: 03 abr. 2015.
- [3] CAVALCANTI, Lana de Sousa. Cotidiano, mediação pedagógica e formação de conceitos: uma contribuição de Vygotsky ao ensino de Geografia. Cadernos CEDES, Campinas, v. 25, n.66, p. 185-208, 2005. Disponível em <<http://www.scielo.br/pdf/ccedes/v25n66/a04v2566.pdf>>. Acesso em: 05/02/1015.
- [4] FRANÇA, Allyson Bonetti; Soares, José Marque. Sistema de apoio a atividades de laboratório de programação via Moodle com suporte ao balanceamento de carga. In Simpósio Brasileiro de Informática na Educação, XXII SBIE - XVII WIE, 21 a 25 de nov. de 2011, Aracaju, Anais. Disponível em: <<http://cbie2014.ufgd.edu.br/workshops/>>. Acesso em: 22 abr. 2015.
- [5] GALASSO, Rafael Hernandez. Integração de um juiz online de algoritmos com o ambiente virtual de aprendizagem Moodle. Itajaí, 2013. 72 f. Trabalho Técnico científico de Conclusão de Curso (Graduação em Ciência da Computação) – Centro de Ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí, Itajaí, 2013.
- [6] MELLO, Suely do Amaral. A Escola da Vygotsky. In: CARRARA, Kester. (Org.). Introdução à Psicologia da Educação: seis abordagens. São Paulo: AVERCAMP, 2004. p. 135 a 155.
- [7] MENDES, A. J. N. Software educativo para apoio à aprendizagem de programação. In: Taller International de Software Educativo, Santiago - Chile, 2001. Disponível em <[http://www.c5.cl/ieinvestiga/actas/tise01/pags/charlas/charla\\_mendes.htm](http://www.c5.cl/ieinvestiga/actas/tise01/pags/charlas/charla_mendes.htm)>.
- [8] OLIVEIRA, Marta Kohl de. Aprendizagem e Desenvolvimento: um processo sócio-histórico. São Paulo: Scipione, 2009.
- [9] SIEBRA, Sandra de Albuquerque; Silva, Danielle Rousy D. da. Prática de Ensino de Algoritmos. Disponível em: <<http://pt.scribd.com/doc/27483736/Pratica-de-Ensino-de-Algoritmo-volume-1e2#scribd>>. Acesso em: 20 abr. 2015.
- [10] VALENTIN, Henryethe, Koscianski, André. Um Estudo sobre o Ensino-aprendizagem de Lógica de Programação. Disponível em: <<http://posgrad.fae.ufmg.br/posgrad/viiienpec/pdfs/137.pdf>>. Acesso em: 05 abr. 2015.
- [11] VYGOTSKI, Lev Semenovich. A Formação Social da Mente. 4.ed. São Paulo: Martins Fontes: 1991.