

Ferramenta para Auxílio na Aprendizagem de Lógica de Programação em Sistemas Informatizados

Ruan M. Carvalho¹, Pedro A. Rosa¹, João Vítor S. Machado¹, José Geraldo Ribeiro Júnior¹, Gabriella C. B. Costa¹

¹Centro Federal de Educação Tecnológica de Minas Gerais – Campus Leopoldina
Leopoldina – MG – Brasil

ruan_mg@hotmail.com, pedrodealmeidarosa@hotmail.com,
joaovitormachado@icloud.com, {jgrjunior,gabriella}@leopoldina.cefetmg.br

Abstract. *For many students, the complexity involved when using logical reasoning creates a barrier to knowledge construction, especially due to the difficulty in pre visualizing the final result. It is clear when you teach logic where is required to create algorithms. This study aims to present a tool to assist the student to build a reasoning logic and realize its execution, simply and clearly. The tool consists of an app and a hardware, both responsible for simulate and execute the sequence defined in the app. The developed tool has been evaluated by students who attend the initial subjects on programming algorithms in this institution.*

Resumo. *Para muitos estudantes, a complexidade envolvida na utilização do raciocínio lógico cria uma barreira para a construção do conhecimento, principalmente pela dificuldade em visualizar o resultado. Isso é evidente no ensino da lógica, necessária para a criação de algoritmos. Este trabalho tem como objetivo apresentar uma ferramenta para auxiliar o aluno na construção de uma lógica de raciocínio e na visualização da execução desta, de forma simples e clara. A ferramenta é composta por um aplicativo e um hardware responsável por simular e por executar a sequência definida no aplicativo. A ferramenta desenvolvida foi avaliada por alunos que cursam as disciplinas iniciais relacionadas à programação de algoritmos.*

1. Introdução

Entre especialistas, há um consenso na visão de que “Ensinar a programar é ensinar a pensar”, sendo forte a linha de estímulo do ensino da programação para crianças [Bopprê 2013]. Com base neste consenso, as empresas, escolas e o governo vêm incentivando o ensino da programação e o desenvolvimento da lógica computacional por meio de aulas, cursos, brinquedos e até feiras com premiações [Bopprê 2013][Pereira 2013][Olhar Digital 2014]. Tal incentivo justifica-se também devido ao fato de que a lógica de programação é uma das disciplinas com maior índice de reprovação em instituições de ensino técnico e superior [Campos 2010]. Além disto, a lógica de programação é uma disciplina de grande importância para alunos que ingressam em cursos de Matemática, Computação e Engenharias, haja vista que os conhecimentos obtidos na mesma influenciarão diretamente no desempenho em outras disciplinas correlatas existentes no curso.

A maioria dos alunos que chegam ao ensino técnico ou superior apresentam deficiências relacionadas ao raciocínio lógico e criatividade, muitas vezes por virem de um sistema educacional que privilegia e exige a memorização/reprodução de informações,

além de punir o erro, inviabilizando o pensamento crítico e criativo. Tal fato é percebido nas primeiras aulas de matérias relacionadas à programação de computadores. A dificuldade no aprendizado de conteúdos relacionados à programação de computadores é citada por Campos (2010) onde, em um estudo realizado com 60 alunos de 2 turmas de instituições de ensino superior, verificou-se que 65% destes têm dificuldades em construir algoritmos, mesmo que estes sejam de baixa complexidade.

Tal dificuldade no aprendizado vem sendo observada, especialmente nos últimos três anos, na instituição da qual os autores deste artigo fazem parte. Esta percepção se deu por meio de um estudo iniciado em 2012, nos cursos técnicos e em 2014 na graduação, envolvendo as disciplinas que trabalham os conceitos iniciais de programação de sistemas. Grande parte dos alunos não consegue assimilar, de forma imediata, a lógica e as estruturas de controle necessárias para a criação de algoritmos. É comum que estes alunos apresentem dúvidas sobre a estrutura do código, estruturas sequenciais, estruturas condicionais, estruturas de repetição, funções e outros conceitos relacionados ao desenvolvimento de algoritmos estruturados. Apesar do esforço dos educadores no ensino destes conceitos, o índice de reprovação nestas disciplinas é considerado alto. Na disciplina de Programação de Computadores I, no curso de graduação em Engenharia de Controle e Automação, por exemplo, em 2014, o índice de reprovação foi de 55% dos alunos matriculados.

Na disciplina de Programação de Computadores I, em um primeiro momento, a linguagem Portugol é usada. Por meio dos exercícios e de ferramentas utilizadas, o aluno visualiza o resultado apenas no próprio computador. Considera-se que a dificuldade para “imaginar” o resultado de um programa é determinante na hora de programar. Assim, se adicionarmos a possibilidade de visualizar, no mundo real, o resultado do que foi programado, estaremos motivando os alunos de programação.

Acredita-se que uma base sólida de conhecimentos, sobre as principais estruturas de controle necessárias para a criação de algoritmos traz, tanto para o aluno quanto para o professor, maior facilidade no processo de ensino e aprendizagem de novas linguagens e estruturas de programação. Assim, o presente trabalho tem como hipótese que, através de uma ferramenta específica para a criação de algoritmos e visualização da execução deste no mundo real, abstraindo, em um primeiro momento, detalhes relacionados a uma linguagem de programação específica, os alunos terão mais facilidade para aprender as principais estruturas de controle, necessárias para programação de sistemas, pois terão em mãos uma ferramenta interativa e de custo moderado.

O presente trabalho possui dois objetivos: (i) Apresentar um aplicativo para dispositivos móveis que permite a criação de uma trajetória fazendo uso dos conceitos de lógica de programação como: estruturas sequenciais, de controle, de repetição e uso de funções. Por meio do aplicativo, o aluno pode criar rotinas lógicas sequenciais que serão executadas por um hardware independente (carro), utilizando a lógica de forma a evitar obstáculos e percorrer caminhos; (ii) Apoiar, com a utilização da ferramenta desenvolvida, o processo de ensino aprendizagem de lógica de programação de sistemas, apresentando um instrumento mais dinâmico aos professores e alunos. Espera-se, assim, que os índices de reprovação nas matérias que envolvem lógica diminuam.

Especificamente, este trabalho destaca as seguintes contribuições: (1) Aplicativo com interface gráfica para dispositivos móveis de fácil utilização para a criação de trajetórias utilizando conceitos de lógica de programação e (2) Hardware que simula os movimentos de um carro, utilizando um microcontrolador arduino [Arduino 2015] e que permite a execução da sequência de comandos criada no aplicativo.

O restante deste artigo encontra-se organizado da seguinte maneira: a Seção 2 apresenta trabalhos que tratam sobre ferramentas ou abordagens relacionadas à proposta deste artigo. A Seção 3 apresenta a metodologia utilizada tanto para o desenvolvimento do aplicativo para dispositivos móveis quanto para a criação do hardware utilizado no projeto. A Seção 4 descreve a ferramenta proposta. Na Seção é apresentada uma avaliação da ferramenta proposta e, por fim, têm-se, na Seção 6 as discussões finais.

2. Trabalhos Relacionados

Nesta seção são apresentadas abordagens e ferramentas que se assemelham à proposta deste trabalho, explicitando seus pontos comuns e as principais diferenças.

Em Oliveira *et al.* (2012), é apresentada uma proposta de associação entre o *storytelling* (método que utiliza palavras ou recursos audiovisuais para transmitir uma história) e a robótica educativa, com o objetivo de auxiliar no ensino da programação. A proposta foi testada por 6 alunos. Diferente da proposta de Oliveira *et al.* (2012), a ferramenta proposta neste artigo faz uso de um hardware utilizando arduino (robô em forma de carro), que visa oferecer aos alunos uma possibilidade de visualização prática do que é programado no aplicativo.

Outra proposta é apresentada por Ferreira *et al.* (2010). São utilizadas técnicas de programação por demonstração, que possibilitam que os usuários, por meio de um ambiente baseado em interface gráfica, programem utilizando rotinas já definidas, sem a necessidade de implementar o código. No estudo, foi utilizada a ferramenta *StageCast Creator* [StageCast 2015], seguida por aulas que incluíam os principais conceitos de Portugol. Após isto, os estudantes foram submetidos a uma avaliação escrita onde foram tratadas questões sobre organização sequencial do raciocínio lógico. Assim como em Ferreira *et al.* (2010), a proposta deste artigo consiste em uma ferramenta a ser utilizada em disciplinas iniciais de programação de computadores, antes que seja introduzida qualquer linguagem de programação específica. Como diferencial, tem-se o hardware capaz de simular no mundo real o que foi programado, ao invés de apenas exibir na tela o que foi feito.

Além dos trabalhos anteriormente apresentados, a seguir são apresentadas algumas ferramentas similares à proposta deste trabalho.

O Scratch [Scratch, 2015] foi desenvolvido pelo grupo *Lifelong Kindergarten*, no *Media Lab* do MIT e está disponível para download gratuitamente. Com o Scratch, o usuário pode programar histórias interativas, jogos e animações e compartilhar suas criações com outros membros da comunidade online. Diferentemente do Scratch, a ferramenta proposta neste artigo foi desenvolvida com o objetivo de ser o mais intuitivo possível, sem grandes necessidades de treinamento sobre como utilizá-la, pois o Scratch possui recursos que não são tão intuitivos a alunos iniciantes em disciplinas relacionadas à programação. O Scratch não está disponível para dispositivos móveis.

O software RoboMind [RoboMind 2015] consiste em uma aplicação *desktop* que tem como objetivo facilitar o ensino da programação e da automação, e, segundo seus desenvolvedores, não se destina apenas ao público infantil devido aos diferentes níveis de dificuldade oferecidos. Muitos dos níveis do RoboMind não são gratuitos, fazendo-se necessário um investimento por parte do usuário para o uso do software. Diferentemente do que ocorre no RoboMind, na ferramenta proposta neste artigo a interação com o cenário se dá de forma física, movendo um carrinho através de um cenário com obstáculos, sem a necessidade de utilização de nenhuma linguagem de programação específica. O RoboMind

não está disponível para dispositivos móveis.

Alice 3.0 [Carnegie Mellon University 2015] aborda os principais conceitos de programação orientada a objetos e não necessita que seus usuários sigam uma sintaxe rígida para programação. Através de um editor com a funcionalidade de *drag-and-drop*, permite manipular personagens e ações para a criação de filmes animados e jogos em 3D. Diferentemente da ferramenta Alice, a ferramenta apresentada neste trabalho mostra o resultado criado no mundo real, através da movimentação de um carrinho.

A ferramenta Primo [Primo 2015] é a que mais se aproxima da ferramenta proposta neste trabalho. Por meio de um tabuleiro, são oferecidos comandos como: frente, esquerda, direita e função. A ferramenta é destinada a crianças de 3 a 7 anos. Uma das desvantagens desta ferramenta é não abranger todos os comandos essenciais para programação, como estruturas condicionais e estruturas de repetição, além do tamanho muito limitado para a programação ou criação da sequência de comandos. Quando comparada ao Primo, a ferramenta proposta neste artigo utiliza um dispositivo móvel para o desenvolvimento da parte lógica do aprendizado, ao invés do uso de uma plataforma física, como acontece no Primo. Com o desenvolvimento de um aplicativo para dispositivos móveis, a ferramenta se torna mais barata e acessível, com a possibilidade de compartilhamento e atualizações das funcionalidades.

3. Metodologia

A partir da observação das dificuldades de ensino e aprendizagem nas disciplinas que envolvem lógica de programação, dadas na instituição de ensino, foi trabalhada e desenvolvida a ideia de criar uma ferramenta para auxiliar neste processo. Para isto, foram elaboradas treze etapas, que foram seguidas de forma a possibilitar tanto o desenvolvimento quando a avaliação da ferramenta proposta, sendo elas: (1) Análise do problema; (2) Planejamento da solução; (3) Definição do escopo da ferramenta; (4) Definição da interface da aplicação móvel; (5) Definição da interface do hardware; (6) Desenvolvimento das funcionalidades da aplicação móvel; (7) Montagem do hardware da ferramenta; (8) Integração entre a aplicação móvel e o hardware; (9) Testes da ferramenta; (10) Elaboração do estudo de caso a ser aplicado na instituição; (11) Aplicação do estudo de caso; (12) Análise dos resultados do estudo de caso; (13) Documentação e publicação dos resultados.

4. A Ferramenta

A versão atual da ferramenta proposta neste artigo é composta por duas partes, um aplicativo e um hardware. Uma sequência de comandos que representam as direções a serem seguidas pelo carrinho para chegar a um determinado destino, desviando-se dos obstáculos existentes, é criada no aplicativo e posteriormente enviada do aplicativo (para dispositivos móveis) para o carrinho, conforme exibido na Figura 1(a). Após isto, o carrinho se move de acordo com os comandos previamente recebidos, de acordo com a Figura 1 (b). Maiores detalhes de cada uma destas duas partes são apresentados nas Seções 4.1 e 4.2, respectivamente.

4.1 Aplicativo para dispositivos móveis

Desenvolvido para o sistema operacional Android, o aplicativo da ferramenta desenvolvida

permite criar uma trajetória utilizando as principais estruturas condicionais e de repetição presentes em programação, além da possibilidade de criação de uma sub-rotina (ou função) pré-definida. A interface permite ainda a pré-visualização da trajetória, antes que a mesma seja enviada, via *bluetooth*, para o carrinho.

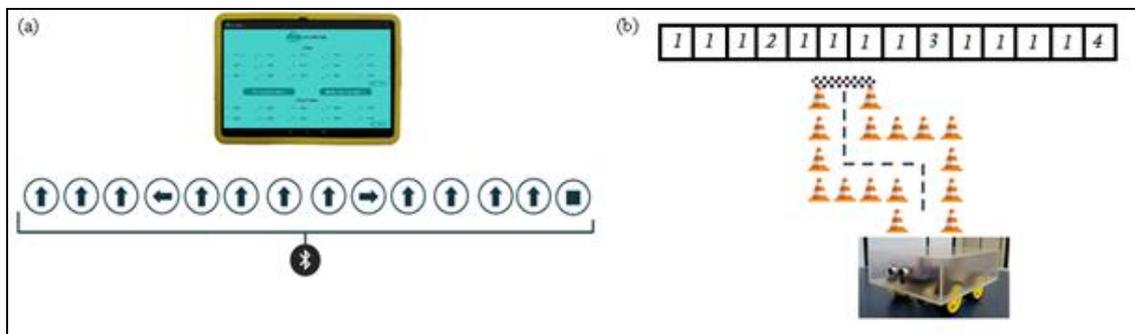


Figura 1: Esquema do Funcionamento do protótipo atual.

Na Figura 2, é possível visualizar a interface principal da versão atual do aplicativo proposto. Nesta interface o usuário é capaz de criar a trajetória a ser percorrida pelo carrinho, escolhendo (para cada movimento) uma das sete opções seguintes: (1) Para Frente; (2) Para Esquerda; (3) Para Direita; (4) Se Obstáculo - Vire Direita; (5) Se Obstáculo - Vire Esquerda; (6) Congelar; (7) Chamar Função.

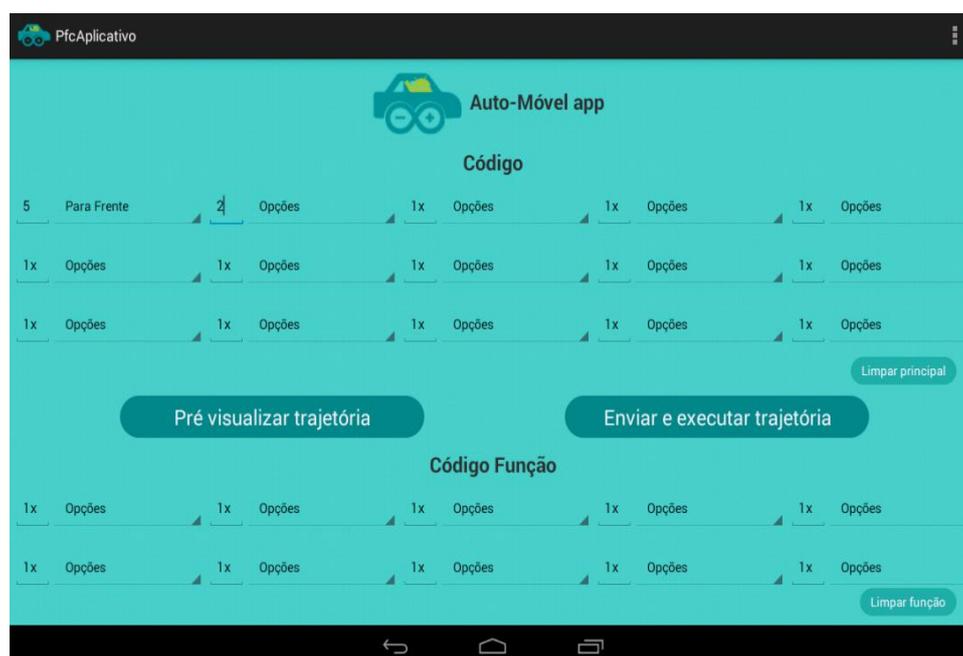


Figura 2: Interface do Aplicativo.

As estruturas de repetição são definidas no aplicativo por meio das caixas de texto (Figura 2) à esquerda da opção escolhida (Para Frente, Para Esquerda, ...). Um número inteiro deve ser digitado pelo usuário, a fim de definir a quantidade de vezes que o comando (à direita) será executado. Dessa forma, atividades com o objetivo de fazer códigos cada vez mais otimizados, podem ser executadas pelos usuários da ferramenta, permitindo resolver problemas mais complexos.

As estruturas condicionais funcionam por meio de duas opções especiais de



Figura 4: Hardware desenvolvido para a ferramenta.

Os componentes exibidos com o número 3 correspondem aos motores CC. Os motores CC utilizados têm como característica possuir duplo eixo e redução. Estes são responsáveis por movimentar o carrinho de acordo com os comandos recebidos pelo arduino. Girando ambos ao mesmo tempo o carrinho se move para frente ou para trás. No aplicativo para dispositivos móveis não há a opção de mover o carrinho para trás, porém essa operação pode ser facilmente realizada com pequenas alterações no código do arduino. Girando os motores que se encontram em uma mesma lateral do carrinho de forma conjunta, este se move para a esquerda ou direita.

O componente número 4 corresponde ao arduino MEGA 2560, uma placa de microcontrolador baseada no ATmega2560. Ele possui 54 pinos de entradas/saídas digitais, 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, uma entrada de alimentação, uma conexão ICSP e um botão de reset. O microcontrolador é o responsável por controlar o carrinho. Ele interpreta os comandos recebidos do aplicativo Android e em seguida envia os comandos aos motores, de acordo com as instruções recebidas.

O componente número 5 é o *shield Bluetooth*. É através dele que é feita a comunicação da aplicação Android com o arduino.

O componente número 6 corresponde ao Sensor ultrassônico. Esse sensor permite que o carrinho detecte obstáculos à frente. Seu funcionamento é semelhante à forma como os morcegos se localizam durante o voo. Para isso, uma onda é emitida e retorna quando colidir com um objeto. A distância desse objeto é calculada pela análise do tempo de retorno da onda.

A versão atual do carrinho desenvolvido para a ferramenta foi feita com acrílico e muitos dos fios foram excluídos por conta da fabricação de uma placa de circuito impresso (Figura 8(b)). Com ela diminui-se a possibilidade de mau contato.

Durante o desenvolvimento da ferramenta, vários desafios nos foram colocados em diferentes áreas. Foi necessário utilizar conceitos de programação de sistemas embarcados, de sistemas microprocessados, conceitos de eletrônica e de transmissão de dados. Apesar de existirem algumas melhorias necessárias na interface do aplicativo, na versão atual, já é possível programar trajetórias para o carrinho, incluindo estruturas de repetição, *delays*, estruturas condicionais e o uso de funções. É possível ainda, visualizar estas trajetórias tanto no aplicativo quanto por meio do carrinho.

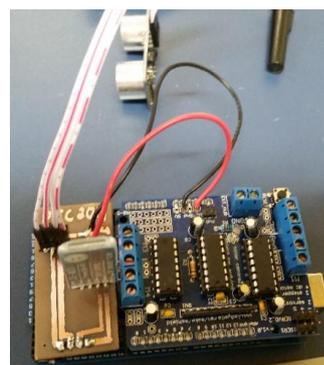
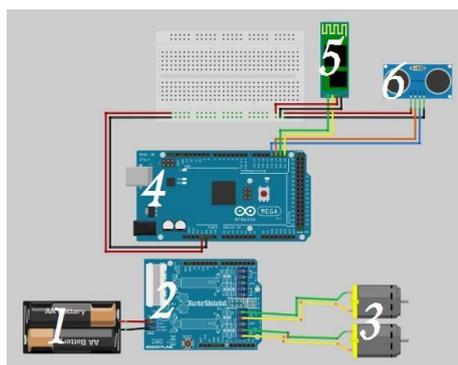


Figura 5(a): Circuito e seus componentes eletrônicos.

Figura 5(b): Placa impressa.

Figura 5: Circuito completo e seus componentes eletrônicos, incluindo a placa impressa.

4.3. Processo para Utilização da Ferramenta

Para a utilização da ferramenta proposta neste artigo, é indicado seguir o processo apresentado na Figura 6, que conta com seis passos: (1) Instruções sobre o funcionamento da ferramenta; (2) Elaboração do percurso a ser realizado pelo carrinho; (3) Criação dos comandos no aplicativo para percorrer o percurso; (4) Visualização da execução dos comandos pelo carrinho; (5) Avaliação do percurso criado pelo aluno; (6) Mostrar ao aluno os erros cometidos e o que precisa ser consertado - sendo estes passos divididos entre passos a serem realizados pelo professor e passos a serem realizados pelo aluno.

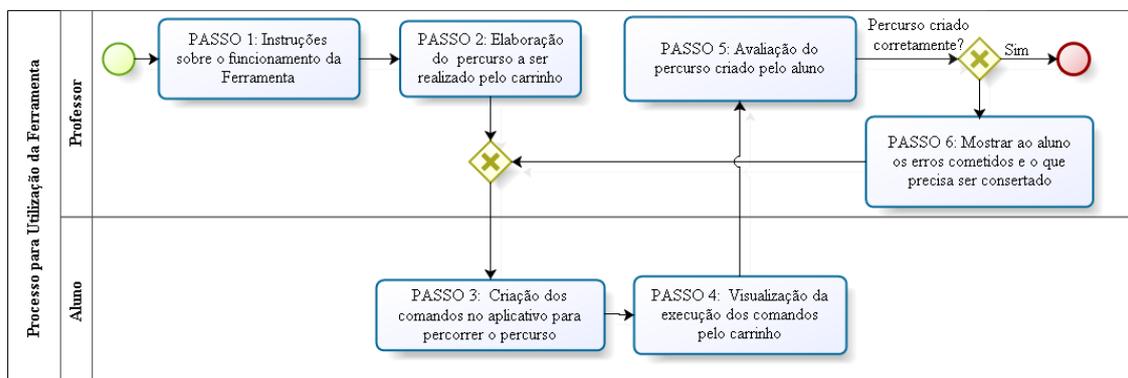


Figura 6: Como usar a ferramenta

Sugere-se a execução do processo exibido na Figura 6 por pelo menos 4 vezes distintas, para cada aluno. Na 1ª vez, deve-se solicitar ao aluno que utilize apenas os comandos que indicam uma direção para o carrinho (Para frente, Para a Direita e Para a Esquerda). Na 2ª vez, deve-se criar um percurso com obstáculos, incentivando ao aluno utilizar as estruturas condicionais ‘Se obstáculo, vire a direita’ e ‘Se obstáculo, vire a esquerda’. Na 3ª execução do processo, deve-se considerar a utilização do mínimo de comandos possíveis para a realização do percurso, utilizando, para isto, as estruturas de repetição de comandos. Por fim, na 4ª vez em que o processo for executado, sugere-se que seja incentivado o uso da função e sua respectiva chamada no programa principal.

Deve-se ressaltar que a instituição em que a ferramenta está sendo aplicada possui turmas de até 12 alunos. Portanto, para a utilização da ferramenta em turmas maiores, deve-se optar pela criação de pequenos grupos de trabalho.

5. Análise dos Resultados

Como cenário para estudo de caso, optou-se por avaliar a ferramenta proposta nas disciplinas de programação dos primeiros anos dos cursos técnicos. Metade destes alunos estava tendo o primeiro contato com a disciplina. Este estudo foi realizado em dezembro de 2014 e um questionário foi respondido após a utilização da ferramenta proposta.

Para utilizar a ferramenta, foram passadas as seguintes informações aos 20 alunos: (1) você deverá criar uma sequência de comandos que seja capaz de levar o carrinho do ponto A ao ponto B, utilizando o mínimo de comandos possíveis; (2) a “função” do aplicativo deverá ser usada sempre que existirem partes que se repetem no código. Esta deverá ser chamada na função principal; e (3) a sequência de execução dos comandos acontece da esquerda para a direita, sempre começando da linha superior.

O primeiro quesito avaliado foi a facilidade de uso no primeiro contato com a interface do aplicativo. Deve-se ressaltar que nenhum dos alunos considerou o aplicativo como sendo de difícil utilização, porém, sugestões de melhoria na interface foram feitas e estas se encontram listadas na Seção 6. Dos alunos avaliados, 50% deles considerou a ferramenta de fácil uso e 50% considerou a facilidade moderada.

No segundo quesito, os alunos tiveram opções de 1 a 5 para considerar a utilidade da ferramenta na disciplina de programação, sendo 5 a nota mais alta. Destes, 57% deu a nota máxima (5) e 43% deu nota 4.

A última avaliação foi sobre a capacidade da ferramenta em prender a atenção do usuário (aluno) onde, 85% considerou que SIM, a ferramenta prende a atenção dos alunos, e 15% consideraram que prende a atenção “MAIS OU MENOS”.

6. Considerações Finais

Este artigo apresenta uma ferramenta desenvolvida com o propósito de auxiliar na aprendizagem de lógica de programação, além de descrever uma sugestão de processo de utilização da mesma. Esta ferramenta é dividida em um aplicativo para dispositivos móveis, utilizado para a criação de uma sequência lógica de passos a serem executados e um hardware, responsável por simular e executar esta sequência.

Um estudo de caso foi realizado de forma a obter-se uma primeira avaliação sobre a utilidade da ferramenta. Esta se mostrou, em um primeiro momento, ter uma utilização classificada entre os níveis ‘fácil’ e ‘moderada’, além de ser útil para o ensino de lógica de programação e prender a atenção do usuário.

Apesar do resultado positivo obtido nesta primeira avaliação da ferramenta, para aprimorar a interface do aplicativo, algumas mudanças serão implementadas para que ela se torne mais dinâmica e autoexplicativa. Uma maneira de alcançar este ponto é por meio do uso da tecnologia *drag and drop* no aplicativo. Com ela pode-se arrastar e soltar os ícones da trajetória, ou seja, a maneira de criar a lógica através da trajetória se tornará mais visual. Isto fará com que o funcionamento da aplicação seja entendido com mais facilidade, permitindo que o usuário se concentre unicamente na criação da lógica.

Percebe-se também a necessidade de inclusão de conceitos como declaração de variáveis, *arrays* e estruturas e seus respectivos tipos na ferramenta proposta. Outra mudança possível é a criação de um menu onde o usuário possa escolher sua idade. Com base nessa escolha, uma interface diferente será exibida, ou seja, a interface para crianças

será mais simples e fácil de usar do que a interface para adolescentes e adultos, que apresentará mais opções. Outro conteúdo apresentado com base na escolha da idade seria dicas de trajetória com dificuldades variadas, adequadas à situação do usuário.

Como os objetivos principais do projeto onde a ferramenta apresentada neste artigo encontra-se inserida consiste em fazer com que os índices de reprovação nas disciplinas que envolvem programação diminuam, assim como, criar um conhecimento sólido a fim de formar profissionais mais capacitados, esta ferramenta será utilizada em maior escala a partir de 2015 em turmas que estão iniciando nos Cursos Técnicos e Graduação. De acordo com os resultados obtidos, a ferramenta está sujeita a atualizações e posteriormente a disponibilização e utilização dela dentro de nossa instituição ou em outras escolas.

Agradecimento

Este projeto foi parcialmente financiado pela FAPEMIG e pelo CEFET-MG.

Referências

- Arduino. (2015) “Arduino Website”. Disponível em: <<http://arduino.cc/>>. Acesso em: 17 jul. 2015.
- Bopprê, V. (2013) “Ensinar a programar é ensinar a pensar”. Disponível em: <<http://porvir.org/porpensar/ensinar-programar-e-ensinar-pensar/20130618>>. Acesso em: 17 jul. 2015.
- Carnegie Mellon University (2015) “Alice 3.0 - An Educational Software that teaches students computer programming in a 3D environment”. Disponível em: <<http://www.alice.org/>>. Acesso em: 23 ago. 2015.
- Ferreira, C., G. e Santos, R. (2010) “Um Estudo sobre a Aprendizagem de Lógica de Programação Utilizando Programação por Demonstração”. In: XVIII Workshop sobre Educação em Computação, XXX CSBC, Belo Horizonte, MG, p. 981-990.
- Olhar Digital. (2014) “Escola que ensina programação para crianças inicia atividades em SP”. Disponível em: <<http://olhardigital.uol.com.br/noticia/40972/40972>>. Acesso em: 17 jul. 2015.
- Oliveira, D., Ferreira, S., Celestino, H., Ferreira, S., Abrantes, P. (2012) “Uma Proposta de Ensino-Aprendizagem de Programação Utilizando Robótica Educativa e Storytelling”. In: II Congresso Internacional TIC de Educação, Lisboa, p. 2567-2576.
- Perreira, L. (2013) “Escolas defendem ensino de programação a crianças e adolescents”. Disponível em: <<http://olhardigital.uol.com.br/noticia/escolas-defendem-ensino-de-programacao-a-criancas-e-adolescentes/35075>>. Acesso em: 17 jul. 2015.
- Primo. (2015) “About”. Disponível em: <<http://www.primo.io/about>>. Acesso em: 17 jul. 2015.
- Robomind. (2015) “Robomind Desktop”. Disponível em: <<http://www.robomind.net/>>. Acesso em: 17 jul. 2015.
- Scratch. (2015) “Sobre o Scratch”. Disponível em: <<http://scratch.mit.edu/about/>>. Acesso em: 17 jul. 2015.
- StageCast.(2015) “StageCast Creator”. Disponível em <<http://www.stagecast.com>>. Acesso em: 17 jul. 2015.