

Plataforma para auxílio ao ensino de programação e robótica pedagógica



Vitória Heliane P. S. Sobrinha ^[1], Gabriela Roberta A. do Nascimento ^[2], Ruan Delgado Gomes ^[3], Otacilio de Araújo Ramos Neto ^[4]

[1] vitoriaheliane@gmail.com, [2] gabrielaalverga@gmail.com, [3] ruan.gomes@ifpb.edu.br, [4] otacilio.ramos@ifpb.edu.br - ^{1,2} Curso Técnico em Informática – IFPB – Campus Guarabira; ^{3,4} Laboratório de Computação Embarcada e Distribuída – IFPB – Campus Guarabira.

RESUMO

Este trabalho descreve uma Application Programming Interface (API), que pode ser utilizada para comunicação e controle de um robô pedagógico. A API foi desenvolvida nas linguagens C++ e Python, que são as primeiras linguagens ensinadas no curso técnico em informática do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB, Campus Guarabira. Os métodos implementados permitem ao robô andar para frente, andar para trás, virar à direita, virar à esquerda e medir a distância até um obstáculo à sua frente. A API encapsula toda a complexidade para comunicação e transmissão dos comandos. Junto a ela, está sendo desenvolvida uma Plataforma de Programação Visual (PPV) em Python. Essa plataforma pode ser utilizada para tornar mais fácil e atrativa a aprendizagem de programação para os alunos iniciantes e alunos surdos. A motivação deste trabalho é utilizar a robótica como uma ferramenta de ensino, ajudando no entendimento de conceitos complexos na área de computação e de outras disciplinas, como física e matemática. Devido à associação de imagens com os comandos da linguagem de programação na PPV, espera-se que os alunos surdos tenham maior facilidade no aprendizado de programação, por meio do uso desta ferramenta.

Palavras-chave: : Ensino de programação. Robótica educacional. Informática na educação. Inclusão social.

ABSTRACT

This paper describes an Application Programming Interface (API), which can be used to communicate and control an educational robot. The API was developed in C++ and Python, which are the first programming languages taught in the technician course in informatics at the IFPB Guarabira. The methods allow the robot walk forward, walk backward, turn right, turn left, and read the distance to an obstacle ahead. The API encapsulates all the complexity of the protocol to communicate with the robot. A visual programming platform (PPV) based on the Python language is also being developed. This platform can be used to facilitate and make more attractive the learning of programming for beginning students, and deaf students. The motivation of this work is the use of robotics as a teaching tool, helping in the understanding of complex concepts about computer programming, and other disciplines, such as physics and mathematics. Due to the association of pictures to the commands of the programming language in the PPV, it is expected that the learning process of deaf students will be improved through the use of this tool.

Keywords: *Programming teaching. Educational robotics. Informatics in education. Social inclusion.*

1 Introdução

As disciplinas introdutórias de programação formam a base para várias outras disciplinas dos cursos da área de informática focados em desenvolvimento de software (MENDONÇA, 2010). Além disso, outros cursos, como os cursos de engenharia, possuem disciplinas de programação, que permitem aos alunos utilizarem o computador de forma mais efetiva no desenvolvimento de suas atividades profissionais e de novas soluções e produtos. Embora seja introdutória, a disciplina “Algoritmos e Lógica de Programação” apresenta alto índice de desistência e reprovação nas turmas do curso técnico em informática do IFPB – Guarabira. Por exemplo, no ano de 2014, 75% dos alunos foram avaliados pelo conselho de classe devido à reprovação nessa disciplina.

A utilização de ferramentas que promovem a prática de lógica de programação, como *Scratch* (SCRATCH, 2016) e *Code.org* (CODE.ORG, 2016), é uma estratégia para aumentar a motivação e melhorar o entendimento dos conceitos por parte dos alunos. Além disso, a robótica pode ser utilizada no ensino de conceitos mais complexos da área de computação, como também na aplicação de conhecimentos obtidos em outras disciplinas, como física e matemática.

Este artigo descreve o desenvolvimento de uma *Application Programming Interface* (API) para permitir a programação de um robô pedagógico por alunos iniciantes. A API foi desenvolvida nas linguagens C++ e Python, que atualmente são as linguagens em uso nas disciplinas introdutórias do curso técnico em informática do IFPB – Guarabira.

Por meio da API, alunos iniciantes de programação poderão colocar em prática os conceitos aprendidos nas disciplinas, utilizando um conjunto de métodos de alto nível que permitem a comunicação com o robô. Toda a complexidade para a comunicação e troca de informações com o robô é encapsulada pela API.

Com base na API, está sendo desenvolvida uma plataforma de programação visual para facilitar o aprendizado de programação para alunos iniciantes e também para alunos surdos. Atualmente, o curso de informática do IFPB – Guarabira conta com dois alunos surdos. Portanto, faz-se necessário o desenvolvimento de ferramentas e metodologias que

permitam melhorar o aprendizado desses sujeitos e lidar com as suas limitações.

2 Trabalhos relacionados

Existem ferramentas de apoio ao ensino de programação e robótica pedagógica, como o *Scratch*, o *Code.org* e a plataforma Mindstorms (LEGO, 2016). O *Scratch* possibilita a criação de histórias interativas, por meio de blocos de comandos lógicos, com palavras-chave indicando os respectivos comandos a serem executados por uma personagem. O *Code.org* visa estimular o ensino de programação, utilizando conceitos básicos e elementos lúdicos para facilitar o entendimento. Assim como o *Scratch*, o *Code.org* utiliza pequenos textos que fazem referência a cada comando a ser executado. O *Mindstorms* propõe uma iniciação à robótica, a partir de uma plataforma para a construção de robôs e a criação de software para ser executado nos robôs desenvolvidos. Também há a ferramenta *Scratch for Arduino* (S4A, 2016), baseada no *Scratch*, que permite o desenvolvimento de softwares para Arduino por meio de blocos de comandos, com o objetivo de facilitar e estimular o interesse pela programação e robótica.

O que diferencia a plataforma em desenvolvimento nesta pesquisa das plataformas *Code.org*, *Scratch* e *Mindstorms*, por exemplo, é a sua maior relação com as linguagens de programação utilizadas nas disciplinas de programação (C++ e Python), o que é fundamental para os cursos técnicos e superiores que possuem tais disciplinas. A ferramenta não sobrepõe os objetivos de ferramentas como o *Scratch* e o *Code.org*, que podem ser utilizadas na parte inicial das disciplinas, como um primeiro contato com a lógica de programação. Em um segundo momento, pode-se utilizar a plataforma proposta neste trabalho para permitir aos alunos um contato inicial com uma linguagem de programação.

Alguns artigos descrevem soluções visando a integração de linguagens de programação com ambientes visuais (MOTA *et al.*, 2008), e outros descrevem ferramentas para auxílio ao ensino de alunos surdos (SANTOS *et al.*, 2014). Mota *et al.* propuseram uma ferramenta de auxílio ao ensino de programação baseada na linguagem Java. Na plataforma desenvolvida, o código escrito é interpretado e uma animação é gerada, mostrando os estados das variáveis durante a execução do programa e também uma descrição textual explicando a execução de cada comando. Embora essa ferramenta seja útil para

facilitar o entendimento da execução de códigos em Java, ela não apresenta componentes que facilitem a construção de programas, como a utilização de elementos gráficos para o desenvolvimento do código, e também não apresenta funcionalidades para o auxílio ao ensino de programação para surdos.

Em Santos *et al.* (2014) é descrita uma plataforma baseada em uma pseudo-linguagem, cujas palavras reservadas são da língua portuguesa, que posteriormente é traduzida para código Java. No entanto, o uso de comandos em português não resolve por completo o problema, uma vez que muitos surdos também possuem dificuldades com língua portuguesa. Além disso, a passagem da linguagem proposta para uma linguagem de programação não é muito natural, devido às significativas diferenças de sintaxe. Para mitigar o problema de entendimento dos comandos, a plataforma apresenta uma janela em que um intérprete realiza explicações em Libras para cada comando, ajudando os alunos a praticarem sozinhos utilizando a ferramenta. A integração dessa funcionalidade na plataforma que está sendo desenvolvida nesta pesquisa é um dos objetivos de trabalhos futuros.

Em Soares *et al.* (2014) é descrita uma ferramenta para auxiliar o ensino de programação para alunos deficientes físicos e auditivos. A ferramenta tem como foco o ensino de programação orientada a objetos, que é um tópico mais avançado e fora do escopo da presente pesquisa. A plataforma desenvolvida no trabalho provê a associação de objetos criados na linguagem com objetos visuais, de modo que modificações nos atributos do objeto no código apareçam de forma visual, assim como a execução dos métodos. A ferramenta descrita no artigo permite a criação de guias de estudo com tradução dos conteúdos para Libras.

Alguns trabalhos já exploraram o uso de robótica como ferramenta de apoio ao ensino. Em Vahldick *et al.* (2009), foi descrito um trabalho que utiliza a plataforma Lego *Mindstorms* no apoio ao ensino de programação. Em Benitti *et al.* (2009), a plataforma Lego *Mindstorms* também foi considerada em conjunto com a linguagem de programação educativa Logo, para criar um ambiente de programação de robôs de fácil entendimento. Em Alves *et al.* (2014) foi descrita uma plataforma de programação para robótica pedagógica, semelhante à ferramenta *Scratch*. Apesar de apresentar uma interface amigável para iniciantes em programação, os comandos disponíveis

são de baixo nível (ex: leia do pino analógico), o que pode dificultar o uso por iniciantes. A API descrita no presente artigo oferece comandos de nível mais alto (ex: andar para frente, virar à direita etc.).

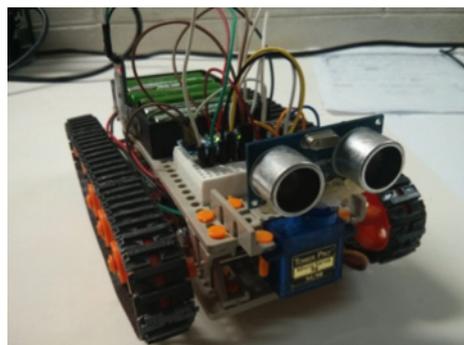
Nossa proposta propõe a utilização de um ambiente visual para facilitar o aprendizado de alunos iniciantes e surdos, com o uso de um robô pedagógico, ao mesmo tempo em que permite um contato inicial com uma linguagem de programação.

3 API para controle do robô pedagógico

O robô pedagógico utilizado neste trabalho (Figura 1) possui dois sistemas embarcados base-ados na plataforma Arduino, um para controle dos motores e outro para realizar o controle geral do robô, incluindo a leitura de dados dos sensores e a interface com o transceptor *Bluetooth*. A comunicação ocorre por meio de um protocolo que permite configurar o robô, acionar os atuadores ou ler dados dos seus sensores.

Esta seção descreve o projeto da API para controle do robô pedagógico utilizado nesta pesquisa. Para desenvolver a API, foi necessário criar um protocolo de comunicação que permitisse a interação entre o software que executa no robô e o *software* de aplicação que utiliza a API. A Seção 3.1 descreve o protocolo de comunicação implementado no robô, e a Seção 3.2 descreve os métodos implementados na API.

Figura 1 – Robô pedagógico utilizado no projeto



Fonte: Dados da pesquisa.

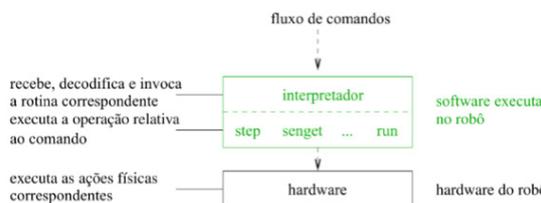
3.1 Projeto do protocolo para controle do robô

O robô utilizado no desenvolvimento da Plataforma de Programação Visual (PPV) foi criado para ser teleoperado, ou seja, o *software* controlador do robô não executa no *hardware* dele, mas em outro

computador qualquer. Robôs teleoperados precisam de técnicas diferentes para serem programados, pois neste tipo de robô não é possível ao *software* de controle acessar diretamente o *hardware* da máquina. Quando existe o acesso direto, a interação com o *hardware* pode ser feita rapidamente, viabilizando assim a criação de controladores reativos. Na plataforma proposta nesta pes-quisa, o sistema de controle não executa no *hardware* do robô. Nesta abordagem, as ações do *software* precisam ser transmitidas aos sensores e atuadores por algum meio de comunicação. A forma como essas ações são executadas também não pode ser feita de maneira idêntica a dos controladores que executam no *hardware* do próprio robô, uma vez que, no caso dos robôs teleoperados, existem outros complicadores, como a latência da rede de comunicação e as quedas de conexão.

Para solucionar o problema do acesso ao *hardware* por um controlador que executa remotamente, foi escrita uma camada de *software* que executa no robô e tem função semelhante a um sistema operacional. Esta camada interage diretamente com o *hardware* e disponibiliza um interpretador simples para um conjunto de comandos. Esses comandos são como pequenos programas que o usuário (ou um *software*) invoca para que o robô realize tarefas simples, como deslocar-se em linha reta ou girar em torno do seu eixo. Os comandos são a base do protocolo de comunicação, pois são eles que trafegam entre o controlador e o robô. Uma ilustração da organização do *software* que execu-ta no robô é apresentado na Figura 2.

Figura 2 – Ilustração da organização do software do robô



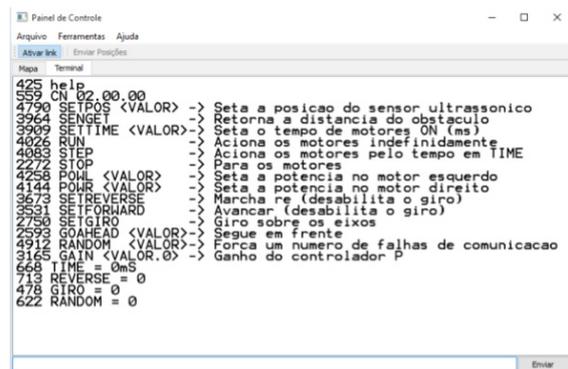
Fonte: Elaborada pelo autor.

Para que os comandos possam trafegar entre o controlador e o robô, foi adicionado um módulo de comunicação *Bluetooth*, que é responsável por disponibilizar a pilha de protocolos *Bluetooth* para a transmissão dos comandos para o robô. O módulo disponibiliza os protocolos até a camada de rede e funciona como um dispositivo de comunicação serial.

Também foram adicionadas rotinas para detecção de erros na transmissão e recepção das mensagens. Estas rotinas atuam sobre todos os dados que trafegam entre o robô e o controlador. A técnica utilizada para a detecção de erros consiste no cálculo do *checksum* do pacote no transmissor e a verificação desse *checksum* no receptor. Se os valores calculados no transmissor e receptor forem iguais, conclui-se que não ocorreu erro. O mecanismo de comunicação utiliza uma pilha com vários protocolos. O que está no topo são os comandos que o interpretador processa; logo abaixo desse está o protocolo responsável pela detecção de erros e, abaixo desse, está a pilha de protocolos *Bluetooth* disponibilizada pelo módulo, que per-mite a transmissão dos pacotes por meio de um canal de comunicação sem fio.

Os comandos do protocolo do Guarabô realizam tarefas simples. Alguns comandos possuem parâmetros numéricos, como tempo ou percentual de potência, enquanto outros comandos não recebem qualquer parâmetro. A lista completa é apresentada na Figura 3. Nela, é exibido um *software* de comunicação serial em uma sessão de comunicação com o robô. Cada linha corresponde a uma mensagem. Na primeira coluna, é apresentada a soma do valor ASCII de todos os caracteres da mensagem, que corresponde ao *checksum*. Em seguida, é apresentado cada um dos comandos: os que possuem <VALOR> logo depois da palavra reservada recebem um parâmetro. Por último, é apresentada uma descrição breve da funcionalidade após a seta.

Figura 3 – Comandos do protocolo de comunicação



Fonte: Elaborado pelo autor.

3.2 Descrição da API

A API foi criada para auxiliar o ensino-aprendizagem de disciplinas introdutórias dos cursos de informática. Ela foi desenvolvida em Python e C++, que são as primeiras linguagens de programação

ensinadas no curso técnico em informática do IFPB –Guarabira. Por meio dela, os alunos iniciantes podem desenvolver *softwares* para controle de um robô pedagógico.

Foram implementados métodos de interação com o robô que permitem fazê-lo andar, girar ao redor do próprio eixo e também medir a distância de um objeto por meio de um sensor ultrassônico. Os métodos implementados são descritos na Tabela 1. Os métodos para controle do robô recebem como parâmetro a distância, tempo ou ângulo, dependendo do tipo do co-mando. O método para leitura do sensor ultrassônico não recebe parâmetros e retorna a distância para o obstáculo à frente em centímetros.

Toda complexidade para a implementação do protocolo de comunicação com o robô (descrito na Seção 3.1) é encapsulada pelos métodos de alto nível, o que facilita o desenvolvimento de softwares pelos alunos iniciantes. A Figura 4 mostra exemplos de código em Python e C++ utilizando a API. Os códigos mostrados fazem o robô andar para frente a uma distância de 5 cm e depois medir a distância para o obstáculo à frente

Quadro 1 – Descrição dos métodos implementados na API

API		
Descrição do método	Parâmetros	Retorno
Andar para frente	distância ou tempo	-
Andar para trás	distância ou tempo	-
Girar à esquerda	ângulo ou tempo	-
Girar à direita	ângulo ou tempo	-
Obter distância	-	distância em cm

Fonte: Elaborado pelo autor.

Utilizando a API, aulas práticas podem ser realizadas, nas quais os alunos terão que utilizar os conceitos estudados em sala de aula, não só nas disciplinas de programação, mas também em outras, como matemática e física. O uso da API e do robô pedagógico pode despertar o interesse dos alunos pela robótica, programação e áreas correlatas.

Figura 4 – Exemplos de código em Python e C++

```

1  from guarabo import Guarabo
2
3  gua = Guarabo()
4  gua.andarFrente_distancia(5)
5  print gua.getDistancia()

1  #include <iostream>
2  #include "Guarabo.h"
3  using namespace std;
4
5  int main(){
6      Guarabo *t;
7      t = new Guarabo();
8      t->andarFrente_distancia(5);
9      cout<<t->getDistancia();
10     delete t;
11     return 0;
12 }
```

Fonte: Elaborado pelo autor.

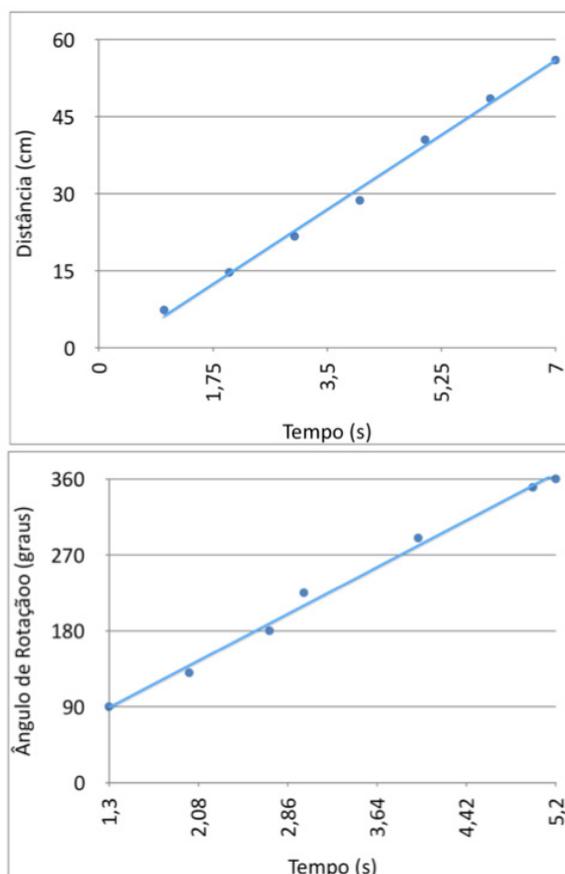
O controle de movimentação do robô é feito por meio do acionamento dos dois motores, um posicionado na esteira do lado esquerdo e outro posicionado na esteira do lado direito. Dessa forma, para implementar os métodos que recebem como parâmetro a distância e os que recebem como parâmetro o ângulo de rotação, foi necessário definir regressões para mapear os valores passados como parâmetro para valores de tempo de ativação dos motores. As regressões foram obtidas a partir de medições realizadas utilizando o robô.

A Figura 5 mostra os gráficos que relacionam as variáveis envolvidas nas regressões obtidas a fim de implementar os métodos para andar para frente (de acordo com a distância recebida) e girar à direita (de acordo com o ângulo de rotação), respectivamente. Por meio dos dados experimentais (os pontos mostrados nos gráficos), foi possível traçar as retas de regressão, que são mostradas na figura. As equações (1), (2), (3) e (4) mostram as regressões que relacionam o tempo de ativação dos motores (T), com os valores passados como parâmetro para os métodos citados acima:

$$\begin{aligned}
 T &= 0,116d - 0,573. \\
 T &= 0,138d - 0,285. \\
 T &= 0,0141a + 0,112. \\
 T &= 0,0055a + 0,133.
 \end{aligned}$$

Em que d é a distância a ser percorrida e α é o ângulo de rotação.

Figura 5 – Gráfico da regressão que relaciona tempo e distância (em cima). Gráfico da regressão que relaciona tempo e ângulo de rotação (em baixo)



Fonte: Elaborado pelo autor.

4 Plataforma de programação visual

A Plataforma de Programação Visual (PPV) é uma ferramenta para auxiliar no ensino-aprendizagem de alunos surdos e iniciantes do curso Técnico em Informática e áreas correlatas, tornando a prática das disciplinas de programação mais fácil e interativa, por meio da possibilidade de construir *softwares* para controle de um robô pedagógico, aplicando os conceitos básicos das disciplinas introdutórias de programação.

A PPV foi integralmente desenvolvida e baseada em Python, pois sua sintaxe é mais simples e melhor compreendida pelos alunos. Os principais comandos e métodos fornecidos pela linguagem estão presentes em uma lista e cada um é associado a uma figura representativa, o que facilita o entendimento da funcionalidade pelos alunos iniciantes e também a

memorização por parte dos alunos surdos, uma vez que eles se comunicam de forma visual.

Por meio do mecanismo de interação *Drag and Drop* (termo em Inglês que significa Arrasta e Solta), os usuários arrastam os comandos desejados, a partir da lista de comandos, e os soltam no campo de desenvolvimento do algoritmo. Essas características podem ser vistas no exemplo de código desenvolvido usando a plataforma, mostrado na Figura 6.

Os comandos são listados no bloco laranja à esquerda, separados por seções (ex: entrada e saída, decisão) e os códigos são desenvolvidos na área branca à direita, por meio de digitação ou arrastando os comandos da lista.

Os botões de gerar código e executar ficam na parte inferior à direita. O primeiro permite salvar um arquivo com código Python equivalente ao algoritmo desenvolvido na plataforma, enquanto o segundo permite abrir um terminal para execução do algoritmo na própria plataforma.

Também é possível enviar comandos de controle para o Guarabô, por meio dos métodos da API do robô pedagógico (descrito na seção 3.2), que é integrada na ferramenta para estimular o interesse pela programação, assim como pela robótica.

Figura 6 – Exemplo de código desenvolvido utilizando a PPV



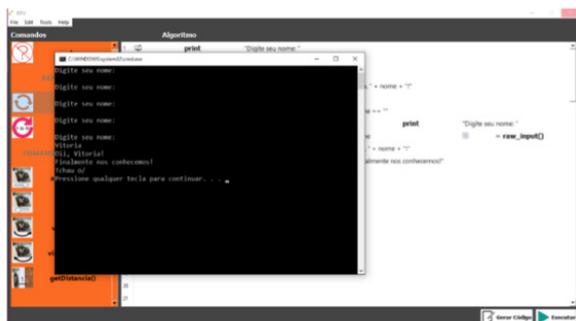
Fonte: Elaborado pelo autor.

Os códigos desenvolvidos podem ser executados na própria plataforma e, caso contenham algum erro de sintaxe, é exibida uma mensagem de fácil entendimento indicando o erro e mostrando um exemplo para ajudar o aluno a descobrir e corrigir o erro cometido. Se o código não apresentar erros de sintaxe, o algoritmo pode ser executado por meio da plataforma (Figura 7). A Figura 8 ilustra um exemplo de mensagem de erro gerada pela plataforma.

Também existe a opção de salvar um arquivo com o código em Python equivalente ao algoritmo

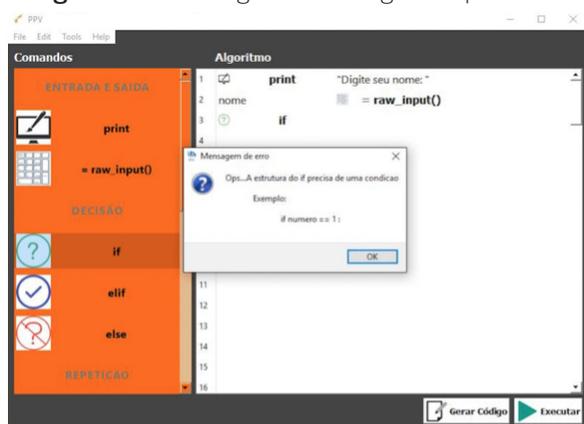
desenvolvido na plataforma. Dessa forma, o aluno pode ter a possibilidade de visualizar e analisar o código, o que pode ajudá-lo na transição entre o uso da plataforma e a programação, diretamente, utilizando a linguagem de programação em forma textual (Figura 9).

Figura 7 – Código executado na plataforma por meio de um terminal aberto



Fonte: Elaborado pelo autor.

Figura 8 – Mensagem de erro gerada pela PPV



Fonte: Elaborado pelo autor.

Figura 9 – Código em Python gerado a partir do algoritmo mostrado na Figura 6

```

1 print "Digite seu nome: "
2 nome = raw_input()
3 if nome != "":
4     print "Ola, " + nome + "!"
5 else:
6     while nome == "":
7         print "Digite seu nome: "
8         nome = raw_input()
9         print "Oii, " + nome + "!"
10        print "Finalmente nos conhecemos!"
11        print "Tchau! o/"

```

Fonte: Elaborado pelo autor.

5 Exemplo de aula prática

Nesta seção, será descrito um exemplo de aula prática – apresentando conceitos de algoritmos e

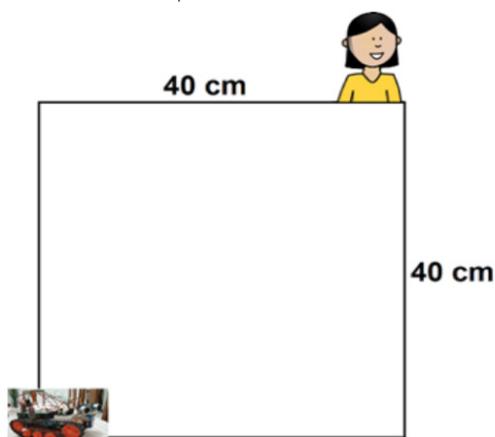
matemática – que pode ser realizada utilizando a PPV e o robô pedagógico.

5.1 Descrição do problema

O Guarabô está em cima de uma mesa quadrada e a nossa amiga Maria precisa dele para terminar o trabalho, mas ela está com preguiça de levantar para pegá-lo, então decide fazer ele vir até ela. Você poderia ajudá-la mostrando qual seria a forma mais rápida de fazer o Guarabô chegar até nossa amiga?

O mapa da Figura 10 ilustra a posição do robô e de Maria e as distâncias entre alguns pontos. Considere que o Guarabô está com a frente virada para a direita, perpendicular às linhas verticais mostradas na figura.

Figura 10 – Mapa para ajudar no entendimento do problema



Fonte: Elaborado pelo autor.

5.2 Solução do problema

Calculando as possíveis rotas para fazer o Guarabô chegar até Maria, pode-se perceber que a rota mais rápida é trafegando em cima da diagonal do quadrado, que mede aproximadamente 56 cm. Como o robô está posicionado com a frente virada para a direita, perpendicular às linhas verticais, é necessário realizar um giro de 45° à esquerda, para depois fazer com que ele ande por 56 cm até chegar em Maria. Na Figura 11, é mostrado o código em Python que resolve o problema, e na Figura 12, é mostrado o código desenvolvido na PPV.

Esse exemplo simples mostra como a plataforma pode ser utilizada para estimular os alunos nos estudos sobre programação e outras disciplinas, como matemática, além de estimulá-los a pensarem na resolução de problemas.

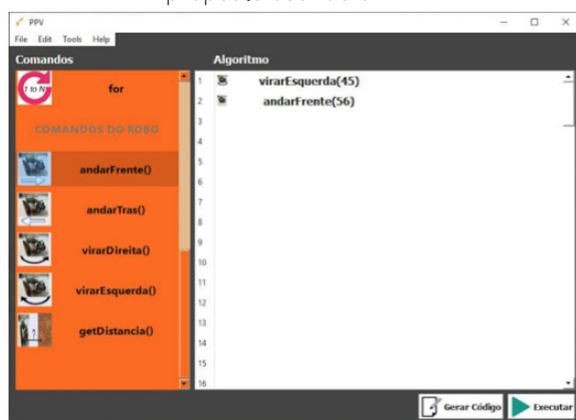
Gradativamente, problemas mais complexos, que envolvam conceitos mais avançados de programação, matemática e outras disciplinas, podem ser introduzidos.

Figura 11 – Código em Python que resolve o problema proposto usando a API

```
1 from guarabo import Guarabo
2
3 gua = Guarabo()
4 gua.girar_esquerda(45)
5 gua.andarFrente_distancia(56)
```

Fonte: Elaborado pelo autor.

Figura 12 – Algoritmo que resolve o problema proposto usando a PPV



Fonte: Elaborado pelo autor.

6 Conclusões e trabalhos futuros

Este artigo descreveu uma API, implementada em Python e C++, para controle de um robô pedagógico. Espera-se que o uso dessa ferramenta aumente o nível de interesse dos alunos iniciantes e sirva de apoio para as disciplinas introdutórias de programação.

Também foi mostrada uma plataforma de programação que permite a construção de programas em Python por meio da composição de elementos visuais. Essa característica é muito útil para o ensino de algoritmos para alunos surdos, pois eles geralmente possuem dificuldade de leitura em língua portuguesa e inglesa. Além disso, a plataforma de programação visual é integrada à API para controle do robô, de modo que os alunos poderão construir algoritmos para desafios de robótica pedagógica utilizando a plataforma.

A próxima etapa do projeto é a integração de todos os comandos básicos de Python e todos os métodos da API para controle do robô com a plata-

forma visual, uma vez que na versão atual apenas alguns comandos estão funcionando completamente (incluindo a identificação de erros de sintaxe). Após isso, roteiros de aula prática, como o mostrado na Seção 5, serão montados para aplicação na turma de Algoritmos do curso técnico integrado em informática do IFPB – Guarabira. Após a aplicação das aulas práticas, será feito um levantamento para identificar possíveis melhorias a serem realizadas e para investigar a influência do uso da ferramenta no nível de aprendizagem e motivação dos alunos.

O código fonte da API e da PPV está disponível publicamente e pode ser acessado por meio do seguinte endereço: <https://github.com/vitoriaHeliane/Plataforma-de-Programa-o-Visual>.

AGRADECIMENTOS

Os autores agradecem ao Laboratório de Computação Embarcada e Distribuída (LACED), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e ao Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), Campus Guarabira.

REFERÊNCIAS

- ALVES, R. M.; SAMPAIO, F. F.; ELIAS, M. F. Duinoblocks: Desenho e implementação de um ambiente de programação visual para robótica educacional. **Revista Brasileira de Informática na Educação**, v. 22, n. 3, p. 126-140, 2014.
- BENITTI, F. B. V.; VAHLICK, A.; URBAN, D. L.; KRUEGER, M. L.; HALMA, A. Experimentação com robótica educativa no ensino médio: ambiente, atividades e resultados. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 29., 2009, Bento Gonçalves – RS. **Anais...** Bento Gonçalves – RS: 2009. p. 1811-1820.
- CODE.ORG. Disponível em: <<https://code.org>>. Acesso em: 15 fev. 2016.
- LEGO mindstorms. Disponível em: <<http://www.lego.com/en-us/Mindstorms>>. Acesso em: 15 fev. 2016.
- MENDONÇA, A. P. **Programação Orientada ao Problema**: uma metodologia para entendimento de problemas e especificação no contexto de ensino de programação para iniciantes. 2010. 187 f. Tese (Doutorado em Ciência da Computação)—Departamento de Sistemas e Computação, Universidade Federal de Campina Grande, Campina Grande, 2010.

MOTA, M. P.; PEREIRA, L. W. K.; FAVERO, E. L. Javatool: uma ferramenta para ensino de programação. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 29., 2008, Belém - PA. **Anais...** Belém – PA: 2008. p. 127-136.

S4A. Disponível em: < <http://s4a.cat>>. Acesso em: 15 fev. 2016.

SANTOS, R. E. S.; MAGALHÃES, C. V. C.; CORREIA-NETO, J. S.; QUEIROS, L. M.; VILAR, G. Trabalhando lógica de programação com portadores de deficiência auditiva: a experiência com a linguagem proglib e a ide hands. **Revista Brasileira de Computação Aplicada**, v. 6, n. 1, p. 32-44, 2014.

SCRATCH. Disponível em: <<https://scratch.mit.edu>>. Acesso em: 15 fev. 2016.

SOARES, M. I. S.; FURTADO JÚNIOR, C. G.; SILVA, L. C.; OLIVEIRA, F. C. M. B.; OLIVEIRA; LIMA, N. A. S.; SOARES, E. F. VISUAL JO2: Um Objeto de Aprendizagem para o Ensino de Programação Java a Deficientes Físicos e Auditivos através do Estímulo Visual – Um Estudo de Caso. **Revista Novas Tecnologias na Educação**, v. 12, n. 2, p. 1-10, 2014.

VAHLDICK, A.; BENITT, F. B. V.; URBAN, D. L.; KRUEGER, M. L.; HALMA, A. O uso do lego mindstorms no apoio ao ensino de programação de computadores. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 29., 2009, Bento Gonçalves – RS. **Anais...** Bento Gonçalves – RS: 2009. p. 523-526.