

Laboratório remoto de robótica para o ensino de programação com suporte à análise, codificação e teste

Maísa Soares dos Santos Lopes¹, Rodrigo Silva Lima¹, João Vitor Ferraz¹,
Matheus Lima Viana¹, Roque Mendes Prado Trindade¹, Alzira Ferreira da Silva¹,
Antônio Cezar de Castro Lima²

¹Departamento de Ciências Exatas e Tecnológicas – Universidade Estadual do Sudoeste da Bahia (UESB) – Vitória da Conquista, BA – Brasil

²Departamento de Engenharia Elétrica – Universidade Federal da Bahia (UFBA) – Salvador, BA – Brasil

{maisa, roquetrindade}@uesb.edu.br, {rodrig0_lima, matheuscond15}@hotmail.com, {jvitor.ferraz14, arizlas}@gmail.com, acdcl@ufba.br

Abstract. *Remote laboratories are being applied in various educational settings, including in teaching computer programming. However, most of these laboratories do not support the phase of problem analysis. This work presents an educational environment for programming teaching that allows analyzing problems, planning, coding and testing solutions. It uses a virtual learning environment, a remote robotics lab, an online compiling tool, and an online text editor. In the environment, the teacher can define practical programming activities and the student can design a plan, build algorithm, implement it in a programming language and test the solution on a remote mobile robot.*

Resumo. *Os laboratórios remotos estão sendo aplicados em diversos cenários educacionais, entre eles, no ensino de programação de computadores. Entretanto, a maioria desses laboratórios não dá suporte a fase de análise de problemas. Este trabalho apresenta um ambiente educacional voltado ao ensino de programação que permite analisar problemas, planejar, codificar e testar soluções. Ele utiliza um ambiente virtual de aprendizagem, um laboratório remoto de robótica, uma ferramenta de compilação online e um editor de texto online. No ambiente, o professor pode definir atividades práticas de programação e o aluno pode projetar um plano, construir algoritmo, implementá-lo em uma linguagem de programação e testar a solução em um robô móvel remoto.*

1. Introdução

O avanço tecnológico permitiu o surgimento de vários laboratórios remotos que estão sendo aplicados em diversos cenários educacionais [Heradio et al., 2016], [Alkhaldi et al., 2016]. Alguns deles possibilitam o ensino de programação de computadores através da integração com ferramentas de programação. Entretanto, gerar código é apenas uma etapa do processo de aprendizagem de programação. Antes disso, é necessário entender o problema para formular uma solução, produzir um algoritmo e, então, escrever o algoritmo em uma linguagem de programação.

Os estudantes novatos tendem a ignorar as fases de análise e projeto, desenvolvendo programas diretamente no computador usando tentativa e erro para encontrar uma solução e não entendem de fato o que estão fazendo [Tan et al., 2014]. Estas fases são importantes pois ajudam a fixar conceitos e desenvolver habilidades de programação uma vez que os alunos são estimulados a pensar sobre o problema e propor soluções adequadas que devem ser implementadas, testadas, validadas e, se necessário, corrigidas e aprimoradas.

Este trabalho apresenta o AAPC (Ambiente de Análise, Programação e Controle) voltado ao ensino de programação para iniciantes. O ambiente tem como objetivo permitir a aplicação das etapas de análise, codificação e teste nas atividades práticas de programação. O AAPC utiliza um ambiente web de programação e controle de um robô móvel de baixo custo integrado a um editor de texto simples. Dentro do mesmo ambiente o estudante pode desenvolver seus algoritmos, implementá-los e testá-los no laboratório remoto.

O artigo está organizado da seguinte forma: a seção 2 apresenta conceitos relacionados ao ensino de programação e laboratórios remotos; a seção 3 descreve alguns laboratórios remotos de robótica móvel voltados ao ensino de programação; a seção 4 descreve o AAPC e, finalmente, a seção 5 apresenta as considerações finais sobre o trabalho.

2. Fundamentação teórica

2.1. Ensino de programação

O objetivo do ensino de programação é capacitar os alunos a desenvolverem soluções computacionais para resolver problemas do mundo real. Portanto, programação não se restringe ao estudo de linguagem de programação, mas envolve um conjunto de princípios, técnicas e formalismos que visam o desenvolvimento de programas de qualidade [Santos & Costa, 2006]. Segundo Koorsse et al. (2015), para desenvolver programas de computador são necessárias três etapas:

1. Entender o problema para formular uma solução – é necessário estudar a declaração do problema, o conjunto de requisitos e decidir sobre a melhor estratégia de programação a ser usada.
2. Produzir um algoritmo para resolver o problema – é necessário conhecer os princípios e conceitos de programação para selecioná-los e aplicá-los corretamente.
3. Traduzir o algoritmo para uma linguagem de programação – isto exige o conhecimento da sintaxe da linguagem de programação. O código do programa é testado e alterado até que o programa atenda ao conjunto original de requisitos e, assim, resolva o problema.

Os alunos devem, então, aprender o processo de resolução de problemas, desenvolver o raciocínio lógico e aprender linguagem de programação. Este é um processo considerado difícil, especialmente para alunos iniciantes. Gomes et.al (2008) identificam um conjunto de fatores que complicam o aprendizado de programação: os métodos de ensino não são de fato centrado no aluno, as turmas grandes não permitem

um ensino personalizado com feedback e supervisão adequados às necessidades de cada estudante; os conteúdos de programação são dinâmicos e abstratos, além disso, é comum começar ensinar os detalhes sintáticos de uma linguagem de programação antes que os alunos percebam qual a finalidade e utilidade de aprender programar; e os alunos estão habituados com o modelo de estudo baseadas em leituras, memorização de fórmulas e uma certa mecanização de procedimentos, mas programação exige intenso trabalho prático extraclasse, uma verdadeira compreensão dos assuntos e reflexão.

Souza et al. (2016) enfatizam que os alunos tem dificuldade para aprender os conceitos de programação, para aplicar estes conceitos durante a construção de programas e falta motivação para realizar as atividades de programação. A maior parte do tempo das disciplinas é gasta ensinando a sintaxe da linguagem de programação e faltam sessões práticas *hand on* [Husain et al., 2013]. Como forma de amenizar tais problemas e dificuldades, vários trabalhos apontam o uso de robôs no ensino de programação.

Robótica na educação é uma tendência em expansão [Cruz-Martín et al., 2012]. A capacidade dos robôs de explorar e interagir com o mundo real através de sensores e atuadores possibilita sua aplicação em uma série de atividades educativas que ajudam e promovem a aprendizagem [Benitti, 2012].

Apesar das vantagens educacionais, a maioria dos laboratórios de robótica só está disponível para os alunos em períodos restritos. O uso de laboratório remoto possibilita compartilhamento de recurso e garante flexibilidade de tempo e espaço para os usuários.

2.2 Laboratório remoto

A expressão laboratório remoto é utilizada para definir experimento que é conduzido e controlado remotamente através da Internet. Os laboratórios remotos utilizam componentes ou instrumentação reais em um local diferente de onde eles estão sendo manipulados. O usuário acessa e controla o computador do laboratório e pode acionar equipamentos, fazer observações, testar condições e coletar dados do experimento. A instalação de câmeras de vídeo no ambiente do laboratório permite que o usuário acompanhe em tempo real a execução do experimento [Chen et al., 2010], [Ma & Nickerson, 2006].

Os laboratórios remotos modernos são ferramentas de software e hardware que possuem um conjunto típico de componentes [Gomes & Bogosyan, 2009]: o objeto controlado (experimento, instrumento ou maquete experimental); os equipamentos e dispositivos de instrumentação que permitem a aquisição e controle de dados; o computador servidor do laboratório que assegura o controle e monitoramento do experimento; o servidor que faz a ligação, através da Internet, entre os usuários remotos e o servidor do laboratório; o servidor de câmera web; e a aplicação cliente que permite ao usuário acessar o experimento através de um navegador web.

Estudos mostram que este tipo de laboratório é uma ferramenta eficaz na compreensão conceitual de conteúdo [Gomes & Bogosyan, 2009], [Machotka et.al, 2009], [Lang, 2012], [Brinson, 2015].

3. Trabalhos relacionados

Apesar de existirem vários laboratórios remotos de robótica atualmente, não foi encontrado na literatura nenhum com suporte as atividades de análise e implementação de programas. Alguns destes laboratórios permitem que os usuários façam *upload* de código que são executados pelos robôs remotos. Alguns outros fornecem ferramenta de programação onde o estudante pode criar e executar o seu próprio código no ambiente do laboratório.

Aroca et al. (2012) propõem um sistema web onde o usuário pode desenvolver seu próprio programa para controlar o robô. O sistema utiliza um smartphone como computador principal para controlar o robô e como servidor web. Entretanto, não há um controle que permita acompanhar as atividades e desempenho de cada aluno.

Iturrate et al. (2013) mostram a integração de um laboratório remoto robótico com *Serious Games*. Os estudantes podem controlar o robô usando o teclado ou através da linguagem de programação gráfica *Google Blockly*. O código é executado no computador que controla e envia comandos para o robô. O programa não é executado no Arduino do robô, mas no computador.

Park & Lenskiy (2014) apresentam um laboratório remoto de robótica para ensino de programação C. O projeto de baixo custo simula sensores através de uma única câmera complementada com o algoritmo de visão computacional. Entretanto, o artigo não mostra o ambiente onde os alunos podem desenvolver os programas.

Chen & Zhou (2015) propõem um ambiente de programação robótico baseado na Web denominado eRobotic. O ambiente permite a programação no estilo arrastar e soltar. Os objetivos do sistema são motivar os alunos e melhorar a compreensão dos conceitos básicos de computação e programação.

4. Ambiente de Análise Programação e Controle (AAPC)

O AAPC é parte do projeto LARA – Laboratório Remoto em AVA (Ambiente Virtual de Aprendizagem), desenvolvido na UESB, que tem como objetivo projetar e implementar uma arquitetura pedagógica que integra recursos tecnológicos e metodologia de ensino para melhorar o processo de aprendizagem de disciplinas do curso de Ciência da Computação.

A estratégia do LARA para atrair e motivar os estudantes de computação para realizar as atividades práticas de programação foi a utilização de laboratório remoto de robótica móvel através do ambiente de programação e controle – APC (Figura 1) [Lopes et al., 2016]. Ele permite que o usuário crie seus próprios códigos para manipular e controlar o robô em um ambiente web sem a necessidade de baixar ou instalar qualquer software adicional.

A versão inicial do projeto permitia a criação e teste de códigos. Em um ambiente educacional, devem ser estimuladas as boas práticas de programação incluindo o entendimento do problema, análise e projeto da solução. O AAPC foi projetado para fornecer suporte a todas estas etapas, além de possibilitar a aprendizagem em contexto, através do uso de laboratório remoto de robótica. Os componentes fundamentais do ambiente são:

- Módulo de estudo – responsável por hospedar material didático que ajuda os alunos na aquisição de conceitos teóricos e na realização de atividades práticas, além de permitir o gerenciamento de mídias educacionais, o gerenciamento dos usuários e a comunicação entre eles.

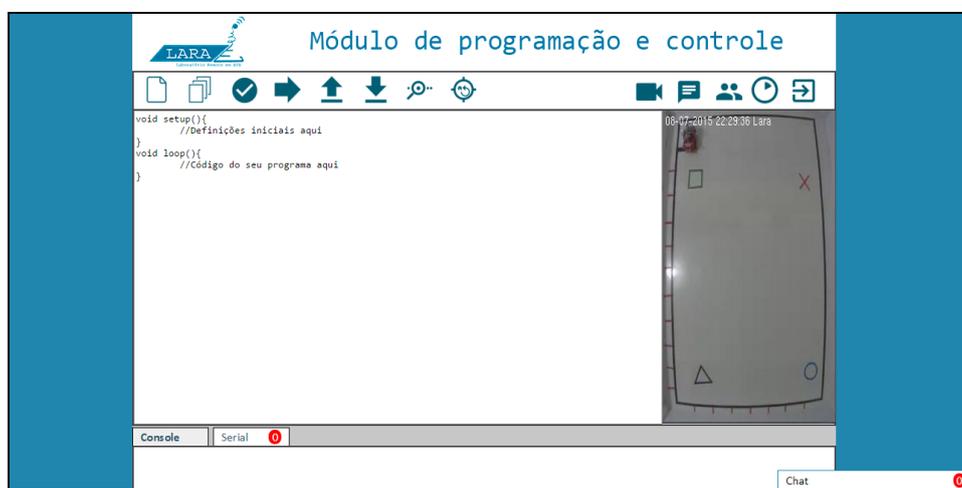


Figura 1 – Ambiente de Programação e Controle do LARA

- Módulo de análise – tem como objetivo estimular as boas práticas de programação incluindo o entendimento do problema e análise da solução. A proposta é que o professor defina um modelo de plano que deve ser seguido e preenchido pelo aluno antes de iniciar a codificação.
- Módulo de codificação – permite a implementação de algoritmos na linguagem de programação Arduino C/C++.
- Módulo de experimentação (ou modulo de laboratório remoto) – permite a manipulação e controle do robô através de códigos desenvolvidos pelo próprio aluno.

A Figura 2 apresenta a visão geral do AAPC. O ambiente é acessado através do AVA *Moodle* que compõe o módulo de estudo. O *Moodle* possui funções de gerenciamento de usuários, de cursos, de objetos de aprendizagem, sistema instrucional de ensino-aprendizagem, entre outras. Para realizar as atividades práticas, o usuário deve realizar a análise do problema, codificação e teste de soluções acessando a área de realização de tarefas que é formado pelos módulos de análise, de codificação e de experimentação. O módulo de análise é composto pelo *Etherpad Lite* (<http://etherpad.org/>) um editor de texto online de código aberto que permite criar e editar planos para resolução de problemas. Os módulos de codificação e experimentação foram agrupados na área de desenvolvimento. Nesta área, é possível implementar algoritmos através do IDE online do APC e testar e validar a solução através do robô remoto.

O sistema hospeda material didático que ajuda os alunos no processo de aprendizagem de programação. Ele permite ao aluno acessar os materiais, recursos e atividades disponibilizados pelo professor, além de realizar atividades utilizando os módulos de análise, codificação e experimentação. Ao definir uma atividade prática, o professor pode disponibilizar a estrutura (modelo ou *template*) do plano para a solução

de um problema. Por exemplo, o professor pode solicitar que o aluno informe o que ele entendeu do problema, quais os requisitos, as entradas e as saídas, quais as variáveis e seus tipos, um algoritmo para solucionar o problema, etc. O aluno entra na área de análise, abre o plano, pensa sobre o problema, edita o plano com as informações solicitadas e salva.

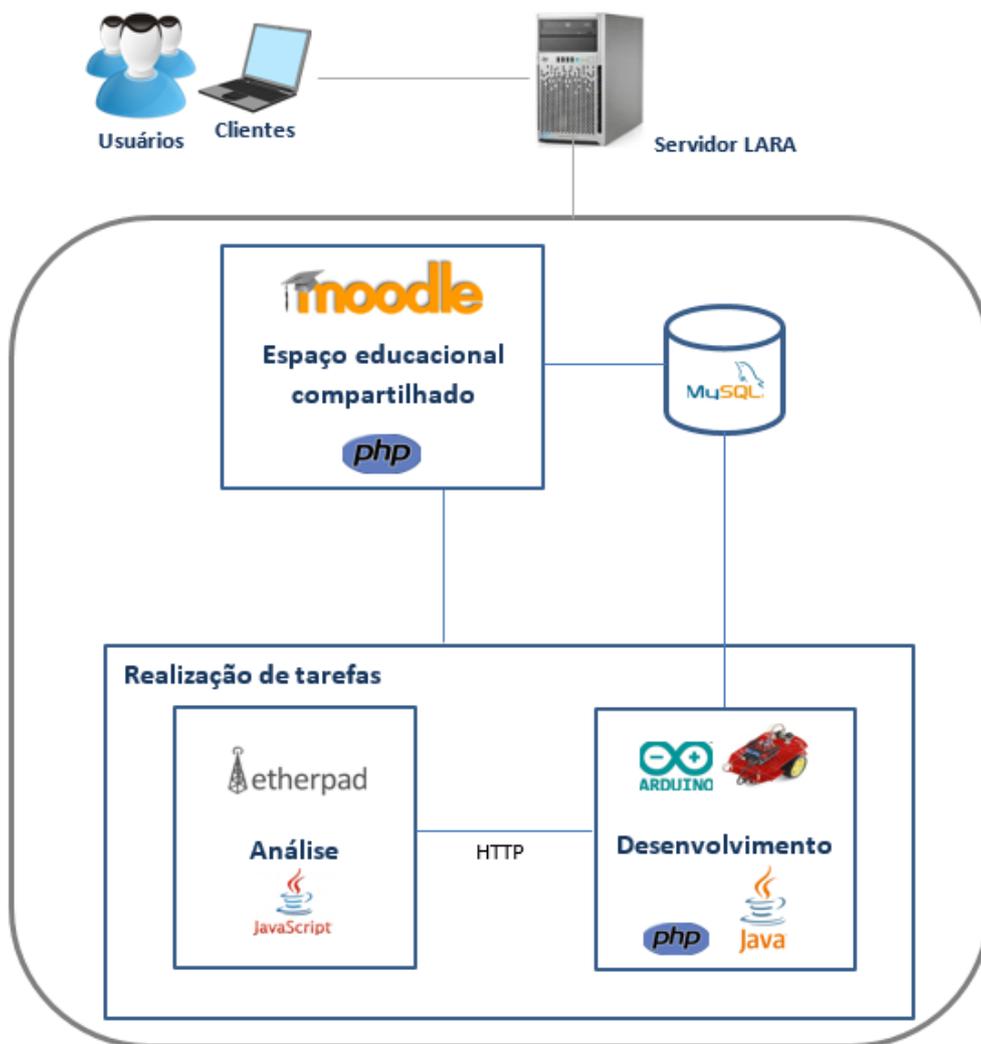


Figura 2 – Visão Geral do AAPC

Na área de desenvolvimento, o aluno executa o plano traçado. Ele cria, salva, abre, edita e compila programas escritos na linguagem de programação. Estes programas devem manipular e controlar o robô. Isto ajuda o aluno no teste e depuração de soluções de problemas. A interação com o robô tem como objetivo permitir a contextualização da programação, pois os professores podem definir contextos de aplicações para as atividades e os alunos podem verificar onde e como aplicar os conceitos abstratos estudados.

A ideia do AAPC é que o alunos leia a atividade a ser realizada, entre no ambiente de resolução de tarefa, na área de análise escreva o problema, pense e registre os requisitos e as restrições referentes ao problema, escreva diversas ideias para resolve-

lo e analise qual a melhor e, então, escreva um algoritmo para a solução escolhida. Na área de desenvolvimento ele traduz o algoritmo para a linguagem de programação, executa e testa. Caso encontre algum erro semântico, ele deve voltar a área de análise, registrar o erro, fazer as correções necessárias na solução, depois implementá-la e testá-la novamente até resolver o problema corretamente.

4.1 Interface do ambiente

A interface do AAPC utiliza os mesmos princípios aplicados na GUI do APC para tornar o ambiente agradável e contínuo, sem ruptura para o usuário. O ambiente é acessado através do site lara.uesb.br, nele os professores podem criar cursos e definir atividades práticas que serão realizadas no laboratório remoto. Para executar as atividades, o aluno escolhe, no menu Laboratório, a opção Acessar Laboratório (Figura 3).

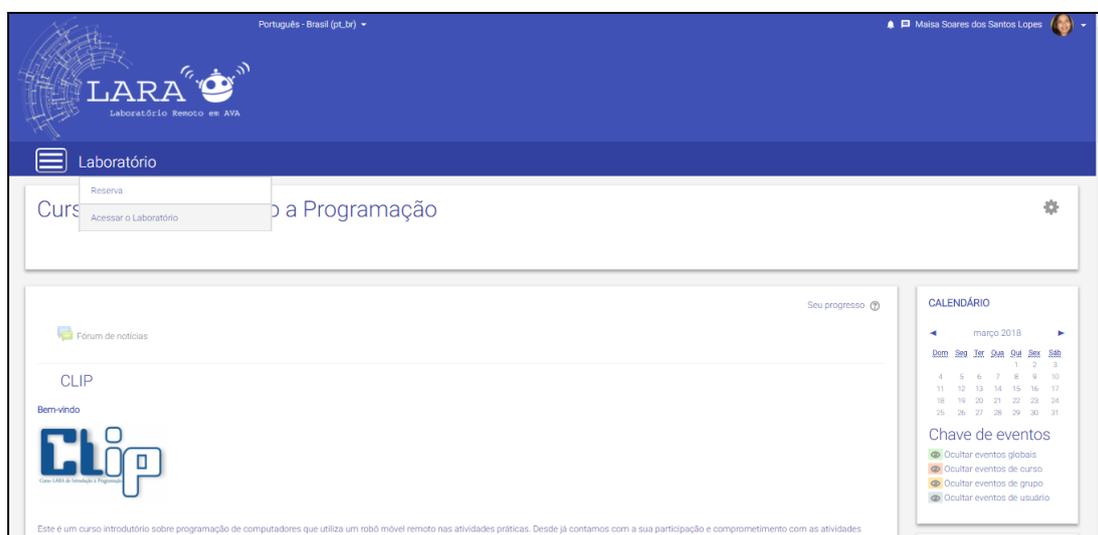


Figura 3 – Interface do LARA

Ao escolher “Acessar Laboratório”, o usuário é direcionado para o ambiente de realização de tarefa mostrado na Figura 4. Nele é possível acessar a área de análise e a área de desenvolvimento. Ambas possuem: uma aba de identificação (1), menu da aba selecionada (2) e área de trabalho (3).

Na aba Análise é possível criar, salvar e abrir um documento, além de formatá-lo. Na área de desenvolvimento (Figura 5), é possível criar novo código, compilar código, enviar código para robô, salvar código, abrir código, abrir porta serial, retornar o robô a posição inicial, abrir/fechar câmera, cronometrar o tempo e sair do ambiente.

5. Considerações finais e trabalhos futuros

Neste trabalho, foi apresentado o ambiente de análise, programação e controle de um robô remoto para o ensino de programação para iniciantes. Para a caracterização do ambiente educacional, foi utilizado o *Moodle* que permite gerenciamento de materiais didáticos e aplicação de estratégias e objetivos de aprendizagem. Para dar suporte as fases de análise e projeto no ensino de programação, foi incorporado um editor de texto onde os estudantes podem definir planos para solucionar as atividades propostas. O

laboratório remoto possibilita a realização de atividades práticas contextualizadas de forma ubíqua, superando as limitações de tempo e espaço dos laboratórios presenciais. A IDE online garante a geração de código em uma linguagem de programação.

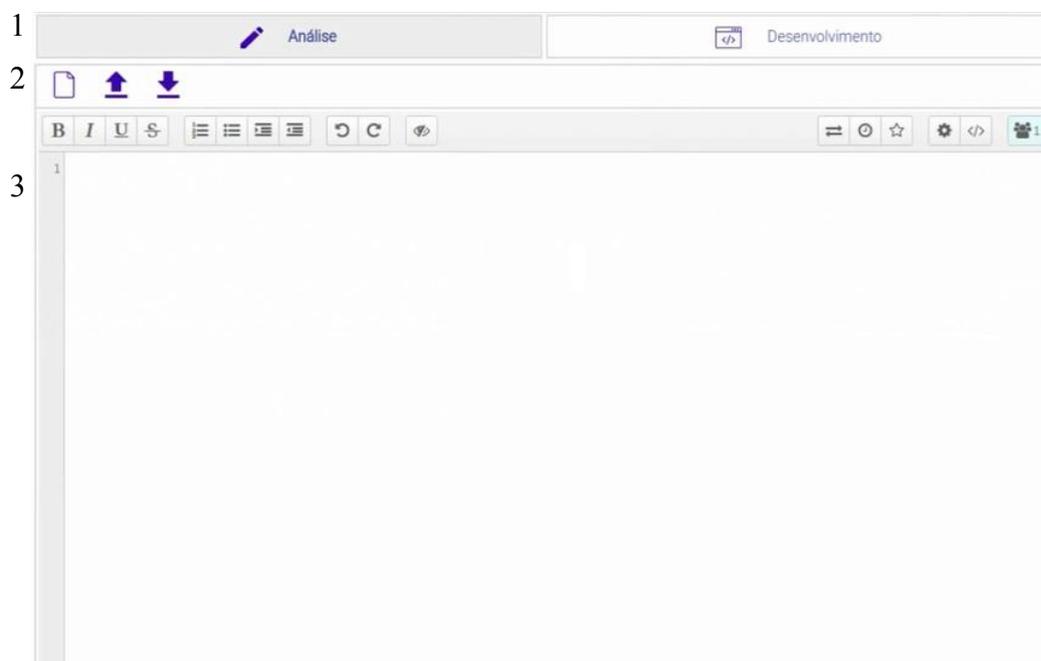


Figura 4 – Ambiente de realização de tarefa



Figura 5 – Área de desenvolvimento

O ambiente aqui apresentado é, portanto, uma alternativa para unir vantagens, baixar custos e proporcionar flexibilidade para criação de contextos educacionais de programação agregando componentes para resolução de problemas (analisar o problema a ser resolvido, formular soluções, produzir algoritmo), codificação de algoritmos,

deapuração e teste de soluções. Entretanto, o sistema ainda não foi testado em um ambiente real e, portanto, não foi avaliada a satisfação dos alunos e suas interações com o AAPC.

Como trabalho futuro, pretende-se utilizar o ambiente em um curso de programação com alunos de graduação e também aprimorar os relatórios do AAPC para que os professores possam acompanhar o progresso de cada estudante, mostrando, por exemplo, o plano elaborado, número de tentativas para codificar o algoritmo, se a solução foi efetiva, tempo investido na realização de cada atividade, etc.

Referências

- Alkhaldi, T., Pranata, I., & Athauda, R. I. (2016). A review of contemporary virtual and remote laboratory implementations: observations and findings. *Journal of Computers in Education*, 3(3), 329–351.
- Aroca, R. V., Gardiman, R. Q., & Goncalves, L. M. G. (2012). Web-Based Robot Programming Environment and Control Architecture. *2012 Brazilian Robotics Symposium and Latin American Robotics Symposium*, 27–32.
- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978–988.
- Brinson, J. R. (2015). Learning outcome achievement in non-traditional (virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research. *Computers & Education*, 87, 218–237.
- Chen, X., Song, G., & Zhang, Y. (2010). Virtual and Remote Laboratory Development: A Review. *Earth and Space 2010: Engineering, Science, Construction, and Operations in Challenging Environments*, 368–368.
- Chen, Y., & Zhou, Z. (2015). Robot as a Service in Computing Curriculum. *2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems*, 156–161.
- Cruz-Martín, A., Fernández-Madrigal, J. a., Galindo, C., González-Jiménez, J., Stockmans-Daou, C., & Blanco-Claraco, J. L. (2012). A LEGO Mindstorms NXT approach for teaching at Data Acquisition, Control Systems Engineering and Real-Time Systems undergraduate courses. *Computers & Education*, 59(3), 974–988.
- Gomes, L., & Bogosyan, S. (2009). Current Trends in Remote Laboratories. *IEEE Transactions on Industrial Electronics*, 56(12), 4744–4756.
- Gomes, A., Henriques, J., & Mendes, A. J. (2008). Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias*, 1, 93–103.
- Heradio, R., de la Torre, L., Galan, D., Cabrerizo, F. J., Herrera-Viedma, E., & Dormido, S. (2016). Virtual and Remote Labs in Education: a Bibliometric Analysis. *Computers & Education*, 98, 14–38.
- Husain, M., Tarannum, N., & Patil, N. (2013). Teaching programming course elective: A new teaching and learning experience. *2013 IEEE International Conference in MOOC, Innovation and Technology in Education (MITE)*, 275–279.

- Iturrate, I., Martín, G., García-zubia, J., Angulo, I., Dziabenko, O., & Orduña, P. (2013). A Mobile Robot Platform for Open Learning based on Serious Games and Remote Laboratories. In *International Conference of the Portuguese Society for Engineering Education (CISPEE)* (pp. 1–7). Porto.
- Koorsse, M., Cilliers, C., & Calitz, A. (2015). Programming assistance tools to support the learning of IT programming in South African secondary schools. *Computers & Education*, 82, 162–178.
- Lang, J. (2012). Comparative Study of Hands-on and Remote Physics Labs for First Year University Level Physics Students. *Transformative Dialogues: Teaching & Learning Journal*, 6(1), 1–25.
- Lopes, M., Gomes, I., Trindade, R., Silva, A., & Lima, A. C. (2016). Web environment for programming and control of mobile robot in a remote laboratory. *IEEE Transactions on Learning Technologies*, 1–1.
- Ma, J., & Nickerson, J. V. (2006). Hands-On, Simulated, and Remote Laboratories: A Comparative Literature Review. *ACM Computing Surveys*, 38(3).
- Machotka, J., Nedic, Z., Nafalski, A., & Gol, O. (2009). A remote laboratory for collaborative experiments. *American Society for Engineering Education*.
- Park, J. S., & Lenskiy, A. (2014). Mobile Robot Platform for Improving Experience of Learning Programming Languages. *Journal of Automation and Control Engineering*, 2(3), 265–269.
- Santos, R. P. dos, & Costa, H. A. X. (2006). Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática. *Infocoomp, Journal of Computer Science*, 5(1).
- Souza, D. M. De, Batista, M. H. S., & Barbosa, E. F. (2016). Problemas e dificuldades no ensino de programação: Um mapeamento sistemático. *Revista Brasileira de Informática Na Educação*, 24, 39–52.
- Tan, J., Guo, X., Zheng, W., & Zhong, M. (2014). Case-based teaching using the Laboratory Animal System for learning C/C++ programming. *Computers & Education*, 77, 39–49.