

Desenvolvimento de módulo didático para ensino de programação

Gustavo Alcântara Elias

CEFET-MG

Av. Amazonas 5253

Belo Horizonte, MG, Brasil

+55-31-3319-7066

gustavo@deii.cefetmg.br

Ana Clara S. Carvalho

CEFET-MG

Av. Amazonas 5253

Belo Horizonte, MG, Brasil

+55-31-3319-7066

anacarvasant@gmail.com

Caroline Silva Araújo

CEFET-MG

Av. Amazonas 5253

Belo Horizonte, MG, Brasil

+55-31-3319-7066

carolinesa.araujo@gmail.com

ABSTRACT

For many students, learn programming skills is a challenge that involves the development of logical-mathematical and abstract reasoning. Sometimes students do not see meaning in the contents they are learning, which can lead to low motivation leading them to fail in the course and high dropout ratings [4,8,15,24,27]. Among the solutions proposed in the literature, one of them is to use programmable and interactive hardware [26]. The objective of this work is to develop a didactic module to be used with Arduino aiming to increase students' learning and motivation in programming. The module will consist of a printed circuit board (PCB) containing components such as sensors, switches, keyboard, LEDs, LCD display and a buzzer, with which students can interact through programs. The development of the module consists of six steps: Electronic circuit design, protoboard circuit assembly and testing, PCB layout design, PCB assembly, final layout design and module assembly and final testing. All the steps were performed and the module worked correctly. It is intended to use the module in the discipline of introduction to programming and to verify its importance in student learning.

RESUMO

Para muitos alunos, aprender a programar é um desafio que envolve o desenvolvimento do raciocínio lógico-matemático e capacidade de abstração. Por vezes os alunos não veem sentido nos conteúdos que estão aprendendo, o que pode culminar em baixa motivação levando a reprovação na disciplina e evasão do curso [4,8,15,24,27]. Entre as soluções propostas na literatura, uma delas consistem em utilizar um hardware programável e interativo por parte do usuário [26]. O objetivo deste trabalho é desenvolver um módulo didático para ser utilizado com o Arduino a fim de aumentar a aprendizagem e motivação dos alunos na aprendizagem de programação. O módulo consistirá de uma placa de circuito impresso (PCB) contendo componentes como sensores, chaves, teclado, LEDs, display LCD e buzzer, com os quais os alunos poderão interagir através de programas. O desenvolvimento do módulo consiste de seis etapas: Projeto do circuito eletrônico, montagem e teste do circuito em protoboard,

elaboração do layout da PCB, montagem em testes da PCB, elaboração do layout final e montagem do módulo e testes finais. Todas as etapas foram realizadas e o módulo funcionou corretamente. Pretende-se utilizar o módulo em disciplina de introdução à programação e verificar como este influencia na aprendizagem dos alunos.

Categories and Subject Descriptors

• Social and professional topics ~ Adult education. • Applied computing ~ Interactive learning environments

General Terms

Algorithms, Design, Experimentation, Languages.

Keywords

Education; Programming; Arduino.

1. INTRODUÇÃO

Os computadores, microcontroladores e sistemas embutidos desempenham um papel fundamental na sociedade atual. São utilizados em áreas como geração e distribuição de energia elétrica, plantas de processamento industrial, saúde, segurança, instituições financeiras, governo, trânsito, aviação, dentre inúmeras outras. Os smartphones, com capacidade cada vez maior de processamento, ocupam um lugar de destaque principalmente entre os jovens e os profissionais. São utilizados tanto para o trabalho, comunicação e entretenimento quase que de maneira onipresente. Profissionais de cursos técnicos de informática, redes de computadores, desenvolvimento de sistemas, programação de jogos digitais e cursos superiores de bacharelado ou licenciatura em ciência da computação, engenharia da computação, engenharia de software, sistemas de informação e computação, desenvolvem softwares que são utilizados nos sistemas citados acima [19,20].

A disciplina de programação, comum nas matrizes curriculares destes cursos, é voltada para a solução de problemas, podendo ser considerada como uma ciência que envolve um conjunto de princípios, técnicas e formalismos. Normalmente, ocorre no início da trajetória curricular com a finalidade de fornecer aos alunos subsídios básicos para o desenvolvimento do raciocínio lógico e da lógica de programação [17]. Seu uso adequado e o entendimento de conceitos, princípios, teorias e tecnologias podem conduzir ao desenvolvimento de software bem estruturado, confiável e de qualidade [26].

O conteúdo programático geral da disciplina de programação aborda temas como conceitos fundamentais, representação de

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 '10, Month 1–2, 2010, City, State, Country.

Copyright 2010 ACM 1-58113-000-0/00/0010 ...\$15.00.

dados, operações, instruções primitivas, desvio condicional, laços de repetição, tipos compostos, matrizes, vetores. Ainda, dependendo da fase em que a disciplina é lecionada, o ensino pode estar atrelado à utilização de alguma linguagem de programação para a construção de programas. Entretanto, não existe um padrão definido, cabendo a cada instituição selecionar e organizar a estrutura da disciplina de seu curso conforme a necessidade de ensino ou de mercado [11,14].

Aprender a programar é algo desafiador para a maioria dos alunos. Muitos precisam desenvolver o raciocínio lógico-matemático, a capacidade de abstração, as habilidades de resolução de problemas e a competência para interpretar os enunciados dos exercícios [8,10,14,23,24,27]. Segundo Chaves de Castro [8], os estudantes tendem a dotar o computador de qualidades humanas, assumindo que este poderia entender comandos incompletos ou errôneos, tornando a necessidade de descrever comandos com rigorosa precisão algo muito artificial e até mesmo antipático. Ainda, as aulas tradicionais que enfatizam a resolução repetitiva de problemas na forma textual, são monótonas e cansativas. A falta de exemplos interessantes faz com que o aluno não veja relevância em certos conteúdos para sua formação, culminando na crença que a disciplina é um obstáculo difícil de ser superado. Tais fatores levam à baixa motivação, aversão ao conteúdo ensinado, apatia, medo, baixa autoestima e frustração, ocasionando altos índices de evasão e reprovação [4, 6,8,15].

Há relatos na literatura de altas taxas de reprovação ou evasão na disciplina de programação. Tal fato faz com que o número de egressos seja significativamente menor que o número de ingressantes tanto nos cursos de nível médio quanto de nível superior [11,21]. De acordo com Campos [7], as disciplinas de lógica, algoritmos e programação iniciam com uma média de 50 alunos, e em poucos meses constata-se que a taxa de reprovação (ou desistência) chega a 60%. Prietch e Pazeto [22] encontraram um percentual de 48,44% de reprovação na disciplina de Programação de 2001 a 2008 no curso de Licenciatura Plena em Informática da UFMT/CUR. A evasão causa, tanto para o setor público como para o setor privado, grandes prejuízos, tais como a ociosidade dos professores, funcionários e infraestrutura [29].

Uma das soluções propostas com o intuito de se reduzir as causas de evasão e retenção e melhorar os resultados na aprendizagem de programação consiste em transformar os processos abstratos em concretos, utilizando exemplos que possuam relação com o mundo real. Isso pode ser feito utilizando ferramentas para auxiliar o aluno no processo de aprendizagem, exemplificando conceitos e trazendo explicações dinâmicas da teoria [28]. Softwares de apoio ao aprendizado, jogos educativos e simulações são alguns exemplos de facilitadores da aprendizagem [26]. Uma outra estratégia consiste em utilizar um hardware programável no qual existam componentes passíveis de interação por parte do usuário. O Arduino é uma plataforma muito difundida e que possibilita essa interação.

Criado na Itália em 2005, o Arduino é uma plataforma aberta de prototipagem eletrônica baseada em hardware e software de fácil utilização. É composto por microcontrolador da família ATMEL em uma placa única contendo uma interface USB, pinos para entradas e saídas digitais, entradas analógicas, canais PWM, comunicação serial, I2C e SPI. Possui um ambiente integrado de desenvolvimento (IDE) que pode ser baixado livremente e utiliza uma linguagem de programação que se assemelha ao C e C++,

suportando estruturas de programação como procedimentos, objetos e controle de fluxo estruturado. É largamente utilizado em nível global e possui aplicações na área de programação, robótica, física, eletrônica, música e artes dentre outros [1,5,6,17].

O objetivo desse trabalho é desenvolver um módulo didático para ser utilizado com o Arduino no intuito de aumentar a aprendizagem e motivação dos alunos na aprendizagem de programação.

2. TRABALHOS RELACIONADOS

Em um projeto intitulado “Computadores podem ser programados para enviar mensagens em código Morse?”, dez estudantes de sete a doze anos participaram com o objetivo de utilizar um microcontrolador programado para enviar mensagens de código Morse através de um LED [14]. Somente dois dos dez estudantes tinham um conhecimento mínimo de programação em C, mesmo assim insuficiente para o trabalho. Escolheu-se o Arduino para essa tarefa, tendo sido realizados três encontros nas manhãs de sábado. Os estudantes ficaram motivados com a tarefa, se dedicando fora dos encontros, inclusive procurando os instrutores. A tarefa foi concluída com êxito por todos os grupos. Os instrutores avaliaram que no curto espaço de tempo do trabalho os estudantes aprenderam conceitos como desenvolvimento de algoritmos, tipos de variáveis, operações com matrizes, loops, decisão (se-então), operadores condicionais, definição e chamada de funções, dentre outros.

Bordignon e Iglesias [3] desenvolveram um projeto para que professores e estudantes do ensino médio pudessem ser capazes de criar objetos interativos programáveis utilizando a plataforma Arduino. Participaram 40 alunos e 18 professores durante 8 meses desenvolvendo projetos propostos pelos autores ou criando seus próprios projetos. Foi necessário aos participantes aprender conteúdos das áreas de eletrônica, programação e da plataforma Arduino, uma vez que, em sua maioria, não os possuíam. Os participantes viram a programação como uma habilidade necessária para desenvolver os projetos e não como um fim em si mesmo. Para aprendê-la, foi utilizado um material didático produzido para o curso. A possibilidade de programar e ver os resultados dessa programação no mundo físico como uma primeira experiência de programação foi atrativa tanto para os estudantes quanto professores participantes do curso. Segundo os autores, embora não tenha sido desenvolvido um conhecimento profundo de programação, foram desenvolvidos os conceitos básicos e despertado o interesse e motivação dos participantes para aprender mais.

A fim de entender os resultados da aplicação do Arduino no ensino de programação, Macedo e Prietch [16] ministraram um curso de extensão para alunos de graduação. Foram utilizados kits contendo uma placa Arduino Uno e componentes eletrônicos. Eram introduzidos conteúdos referentes ao Arduino, conexão de componentes à placa, exercícios práticos e desenvolvimento de um projeto decorrente dos conteúdos aprendidos. Notou-se que os estudantes aprendiam os conteúdos com relativa facilidade, sendo capazes de montar os projetos individualmente ao final. Também foi observado que os estudantes ficaram motivados principalmente por visualizar no hardware montado o que foi programado. Foi observado que os participantes estavam dispostos a ampliar os estudos sobre o Arduino após o término do curso. Segundo os autores, o projeto auxiliou os alunos na

aprendizagem de lógica de programação e prática de ensino de algoritmos.

Para introduzir conceitos programação em sistemas embarcados, no primeiro semestre do curso de Bacharelado em Eletrônica na universidade de Sofia, foi utilizado o Arduino em aulas práticas de laboratório [30]. Conceitos como loops, tomada de decisão, matrizes e controle de entradas analógicas e digitais são aprendidos. A maneira como os conceitos foram tratados atraiu a atenção da maioria dos estudantes, que superavam as tarefas com relativa facilidade. Este sucesso os encorajava e convencia a aprender mais para desenvolver suas habilidades. O autor afirma que em um curso introdutório é melhor enfatizar a prática do que teorias acadêmicas pontuais. Verificou-se que o Arduino ofereceu um ambiente amigável e útil para uma educação introdutória na área de programação no qual os estudantes obtiveram conhecimento e experiência prática que serão úteis ao longo de sua jornada acadêmica.

Estudantes de engenharia elétrica na University College of Southeast Norway cursavam no primeiro e segundo semestre do curso as disciplinas de programação em C++ para PC e assembly e programação em C para o microcontrolador PIC18, respectivamente. Ambas as disciplinas utilizavam simulação uma vez que havia dificuldade prática em utilizar circuitos reais, pois estes ocupavam grande parte do tempo dos alunos, dificultando-os ao invés de ajuda-los a desenvolver habilidades de programação. Em uma reformulação curricular, o PIC18 foi substituído pelo Arduino e o conteúdo passou a ser ministrado em um semestre, ao invés de dois. É requerido que os estudantes desenvolvam conhecimento de programação em C/C++ e em estrutura de microcontroladores, bem como sua utilização em aplicações profissionais. Os estudantes podem adquirir seu próprio kit contendo uma placa Arduino e um conjunto de sensores e atuadores, trabalhando quando desejarem. Tornou-se fácil motivar os alunos, uma vez que eles veem o resultado de suas ações no mundo real, ao invés de medir tensões em terminais de circuitos integrados ou em simuladores. O nível de conhecimento dos estudantes aumentou significativamente em relação à maneira anterior, tendo aumentado significativamente sua compreensão e capacidade de codificação em disciplinas ministradas posteriormente no curso. O número de estudantes com nota D na disciplina diminuiu de 50% para cerca de 20% enquanto o número de estudantes com nota A aumentou de 20 para quase 50% [13].

Um modelo para ensinar programação em Python através do Arduino é proposto por Ariza [2]. O autor propõe a integração entre Python e Arduino através da *Pyfirmata*, uma interface Python para o protocolo *Firmata* que, por sua vez, é o protocolo para comunicação entre microcontroladores e o software do computador. Foi ministrado um curso no qual os estudantes eram capazes de ler valores de portas digitais e analógicas, escrever valores em portas digitais, gerenciar periféricos através de comunicação SPI (*Serial Peripheral Interface*), UART (*Universal Asynchronous Receiver Transmitter*) ou I2C (*Inter-Integrated Circuit*). Também era possível criar interfaces gráficas de usuário, exportar dados e plotar sinais, dentre outras funções. Em resumo, a estratégia proposta permite combinar conceitos relacionados à linguagem de programação bem como de engenharia eletrônica e de tecnologia.

Para auxiliar os estudantes de graduação a superarem as dificuldades de aprendizagem de programação em um curso introdutório, Rubio e colaboradores [25] desenvolveram módulos

didáticos para uso com o Arduino. Estes módulos contêm LEDs, autofalantes e servo motores para possibilitar o ensino de diferentes conceitos de programação como matrizes, estruturas condicionais e loops. Para avaliar a aprendizagem utilizando-se os módulos, os autores dividiram alunos de 18 anos sem conhecimento prévio de programação em dois grupos com 39 e 38 alunos. O primeiro foi denominado grupo controle no qual seria ensinado programação da maneira tradicional. O segundo foi denominado grupo experimental e utilizaria os módulos desenvolvidos. O mesmo professor ministrou a disciplina em ambos os grupos, no mesmo semestre visando ensinar o mesmo conteúdo. Verificou-se que a taxa de reprovação no grupo experimental (7,9%) era significativamente menor do que no grupo controle (25,6%). O aumento da percepção da facilidade de programar também foi maior no grupo experimental (34%) ante o grupo controle (20%). O índice de satisfação de utilização da plataforma Arduino chegou a 90% dos alunos do grupo experimental, tendo os estudantes demandado mais aulas utilizando módulos didáticos. Os autores afirmam que os módulos são projetados para aprimorar a metodologia de ensino tradicional e não substituí-la. Os professores explicam um conceito computacional usando a metodologia tradicional e depois o reforçam usando os módulos com o Arduino.

3. DESENVOLVIMENTO

O módulo didático proposto tem como característica principal permitir a visualização no mundo físico da execução de programas com o intuito de motivar a aprendizagem dos alunos. Consistirá de uma placa eletrônica que será conectada ao Arduino. Nesta placa estarão presentes dispositivos para leitura e escrita de dados, descritos abaixo. Pretende-se que ele seja utilizado para ensinar conceitos fundamentais de programação C/C++ como tipos de variáveis (*int, char, byte, double, string*, etc), controle de fluxo (*if-else, for, while, do-while, switch case*, etc), operadores (aritméticos, relacionais, lógicos, bit a bit) e entrada e saída. Buscou-se torna-lo de baixo custo para que o aluno que tiver interesse possa adquiri-lo ou mesmo construí-lo para utilizar quando possível.

Os componentes presentes no módulo foram determinados baseando-se em kits didáticos para o ensino de sistemas microcontrolados [9,12,18] uma vez que estes kits utilizam compiladores em linguagem C e são utilizados para aprendizagem de conteúdos como os citados no parágrafo anterior. Há elementos de entrada digital e analógica e de saída digital. Os elementos de entrada digital são compostos por quatro *push buttons*, quatro chaves duas posições (liga-desliga) e um teclado de membrana 4x4 e os elementos de entrada analógica são um sensor de temperatura tipo NTC e um sensor de luminosidade tipo LDR. Já os elementos de saída digital são compostos de nove LEDs sendo três verdes, três amarelos e três vermelhos, um *buzzer* e um display LCD 16x2. A figura 1 mostra como os elementos irão interagir com o Arduino.

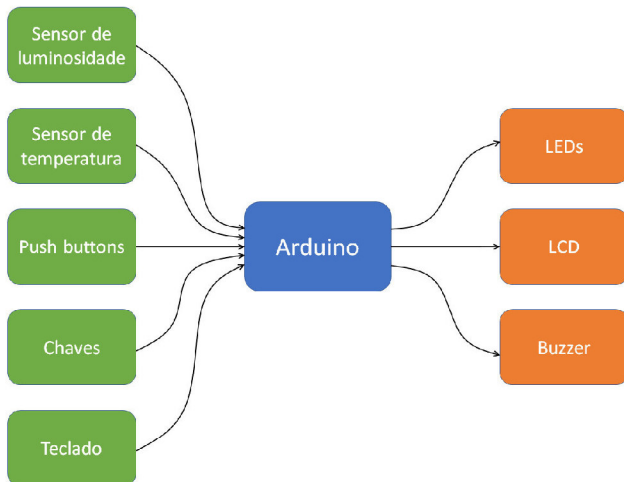


Figura 1. Componentes do módulo e interação como Arduino

Baseando-se nos componentes citados, temos a necessidade de uma placa que comporte, pelo menos, 25 entradas/saídas (E/S) digitais, duas entradas analógicas e 6 entradas digitais com PWM para o display. Inclui-se a capacidade de fornecimento de corrente elétrica para os componentes conectados a cada porta. Existem cerca de 21 placas diferentes do Arduino disponíveis [1] e a que atende as especificações técnicas em termos de disponibilidade de entradas e saída e corrente é a Arduino MEGA 2560. Esta placa possui 54 pinos de E/S digital sendo que 15 possuem saída PWM e 16 entradas analógicas. Caso seja necessário, é possível acrescentar outros componentes no módulo uma vez que o Arduino MEGA ainda comporta essa expansão.

Após a definição dos componentes presentes no módulo, as etapas necessárias para desenvolvê-lo são mostradas na Figura 2. A primeira etapa consiste em projetar um circuito eletrônico para cada um dos oito itens mostrados na Figura 1 (sensor de luminosidade, sensor de temperatura, *push buttons*, chaves, teclado, LEDs, LCD e *buzzer*). Em seguida, o circuito projetado será montado individualmente em um protoboard e conectado ao Arduino para testes. Após os circuitos terem sido testados, eles serão montados todos juntos para o teste final de funcionamento. Possíveis problemas devem ser sanados para que possa ser elaborado um layout da placa de circuito impresso na qual o módulo será montado para conexão no Arduino Mega. A placa eletrônica será montada e testada a fim de verificar seu funcionamento, corrigir eventuais problemas e elaborar o layout definitivo. Por fim, o módulo será montado e testado para poder ser aplicado em sala de aula.

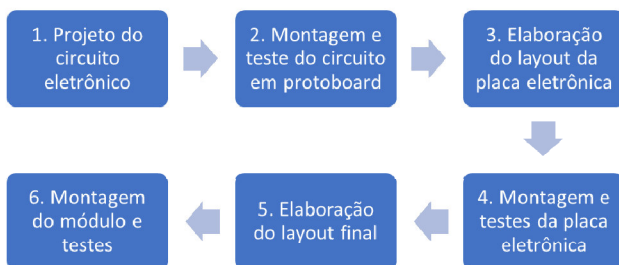


Figura 2. Etapas de desenvolvimento do módulo

4. RESULTADOS

Os circuitos foram projetados, montados e testados no protoboard, conforme as etapas 1 e 2 mostradas na figura 2. Foi elaborado um layout de uma placa de circuito impresso que foi confeccionada e o circuito montado conforme mostrado na figura 3. Foi elaborado um programa para testar todos os seus componentes. Verificou-se que era necessário colocar um componente para ajustar a luz de fundo do LCD e que a posição do conector do teclado de membrana estava dificultando sua utilização, tendo que ser reposicionado. As respectivas modificações no layout circuito foram feitas. Foi elaborado um layout da placa em face dupla, diminuindo seu tamanho. O custo estimado do módulo confeccionado e montado está em torno de R\$ 120,00 até o momento.

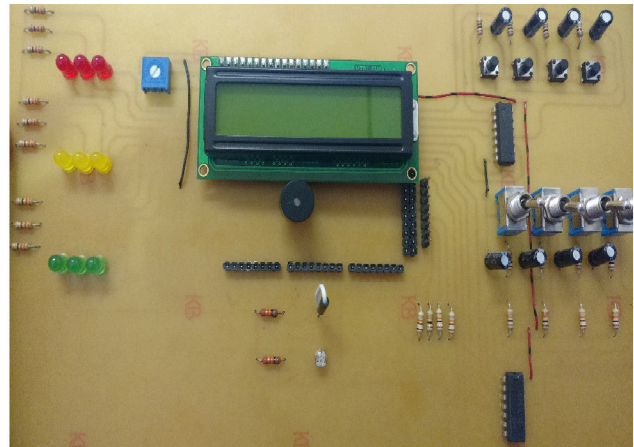


Figura 3. Primeiro protótipo do módulo

5. DISCUSSÃO E TRABALHOS FUTUROS

Uma empresa especializada confeccionará 10 placas dupla face para realizar a montagem dos módulos que serão utilizados em aulas. Será desenvolvido material didático com o objetivo de ensinar conceitos de programação utilizando o módulo desenvolvido. Pretende-se aplicar o trabalho em aulas práticas da disciplina de Informática Aplicada do curso técnico em Equipamentos Biomédicos do Cefet-MG. Será elaborado e aplicado um questionário ao final do curso objetivando conhecer como e se este módulo influenciou na aprendizagem, bem como a percepção dos alunos sobre as práticas.

Conforme a literatura descrita anteriormente, a utilização de dispositivos físicos para o ensino de programação tem aumentado a compreensão dos alunos. É possível para eles ver o efeito de suas ações no mundo real, o que gera significado no que estão fazendo, aumentando sua motivação para aprender um tema por vezes percebido como difícil. Assim, acreditamos que este trabalho irá contribuir para a aprendizagem dos conteúdos de programação, diminuindo a reprovação e evasão nos cursos.

6. AGRADECIMENTOS

A Fapemig e ao CEFET-MG pelo apoio financeiro.

Ao prof. Leonardo Vasconcelos Alves pelas considerações a respeito do trabalho.

7. REFÊRENCIAS

- [1] Arduino. Disponível em <www.arduino.cc>. Acessado em: Setembro 2017.
- [2] Ariza, J.A. (2016). A proposal for teaching programming languages through open hardware tools. *IEEE International Conference on Engineering Education (ICEED)*. 202-207.
- [3] Bordignon, F.R.A., Iglesias, A.A. (2016). Más allá de las pantallas: experiencias en diseño y programación de objetos interactivos digitales. *Virtualidad, Educación y Ciencia*, 12(7), 49-58.
- [4] Borges, M. A. F. (2000). Avaliação de uma Metodologia Alternativa para a Aprendizagem de Programação. *VIII Workshop de Educação em Computação – WEI*. Curitiba, PR, Brasil.
- [5] Brock, J.D., Bruce, R.F., Reiser, S.L. (2009) Using Arduino for Introductory Programming Courses: A Tutorial. *Journal of Computing Sciences in Colleges*. 25(2), 129-130.
- [6] Bruce, R., Brock, D., Reiser, S. (2013). Teaching Programming Using Embedded Systems. *IEEE Southeastcon*. 1-6.
- [7] Campos, R. L. B. L. Lógica de Programação: Há como melhorar o aprendizado fugindo dos padrões estabelecidos nos livros didáticos e adotados pela maioria dos docentes?. *XVII Congresso Iberoamericano de Educación Superior em Computacion - CLEI*, Pelotas, RS, Brasil.
- [8] Chaves de Castro, T., Castro Junior, A., Menezes, C., Boeres, M. e Rauber, M. (2002). Utilizando Programação Funcional em Disciplinas Introdutórias de Computação. *XI Workshop de Educação em Computação – WEI*. Florianópolis, SCP, Brasil.
- [9] Datapool Eletrônica. Disponível em <eletronica.datapool.com.br>. Acessado em: Setembro 2017.
- [10] Delgado, C., Xexeo, J. A. M., Souza, I. F., Campos, M., Rapkiewicz, C. E. (2004). Uma Abordagem Pedagógica para a Iniciação ao Estudo de Algoritmos. *XII Workshop de Educação em Computação - WEI*. Salvador, BA, Brasil.
- [11] Deters, J. I.; Da Silva, J. M. C.; de Miranda, E. M.; Fernandes, A. M. R. (2008). O Desafio de Trabalhar com Alunos Repetentes na Disciplina de Algoritmos e Programação. *Simpósio Brasileiro de Informática na Educação*.
- [12] Exsto Tecnologia. Disponível em <www.exsto.com.br>. Acessado em: Setembro 2017.
- [13] Graven, O.H., Bjørk, J. (2016). The use of an Arduino pocket lab to increase motivation in Electrical Engineering students for Programming. *IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. 239-243.
- [14] Hill, L., Ciccarelli, S. (2013). Using a low-cost open source hardware development platform in teaching young students programming skills. *4th annual ACM SIGITE conference on Information technology education*. 63-68.
- [15] Koliver, C., Dorneles, R. V., Casa, M. E. (2004). Das (muitas) dúvidas e (poucas) certezas do ensino de algoritmos. *XIII Workshop de Educação em Computação - WEI*. Salvador, BA, Brasil.
- [16] Macedo, R.S., Prietch, S.S. (2013). Utilizando a Placa Arduino como Ferramenta de Apoio ao Ensino de Programação. *Revista de Exatas e Tecnológicas*, 4(2), 56-62.
- [17] Melo, R.M. et al. (2014). Utilização do Arduino como Estratégia Pedagógica na Aprendizagem da Lógica de Programação. *9th Iberian Conference on Information Systems and Technologies -CISTI*. 1-5.
- [18] Microgênios. Disponível em <www.microgenios.com.br>. Acessado em: Setembro 2017
- [19] MINISTÉRIO DA EDUCAÇÃO. Parecer CNE/CES nº 136/2012. Diretrizes Curriculares Nacionais para os cursos de graduação em Computação. Disponível em <http://portal.mec.gov.br/index.php?option=com_docman&ask=doc_download&gid=11205&Itemid=>. Acessado em: Agosto 2017.
- [20] MINISTÉRIO DA EDUCAÇÃO E CULTURA (2016). Catálogo Nacional dos Cursos Técnicos 3ª Ed. Disponível em <sitesistec.mec.gov.br/component/banners/click/18>. Acessado em: Agosto 2017.
- [21] Pereira Júnior, J.C.R., Rapkiewicz, C.H. (2006). O Processo de Ensino-Aprendizagem de Fundamentos de Programação: Uma Visão Crítica da Pesquisa no Brasil. *XIII Workshop sobre Educação em Computação = WEI*, Campo Grande, MS, Brasil.
- [22] Prietch, S.S., Pazeto, T.A. (2010). Estudo sobre a Evasão em um Curso de Licenciatura em Informática e Considerações para melhorias. *VIII Workshop de Educação e Informática – Weibase*, Maceió, AL, Brasil.
- [23] Raabe, A. L. A. e Silva, J. M. C. (2005). Um Ambiente para Atendimento das Dificuldades de Aprendizagem de Algoritmos. *XIII Workshop de Educação em Computação - WEI*. São Leopoldo, RS, Brasil.
- [24] Rodrigues, M. C. (2002). Como Ensinar Programação?. *Informática – Boletim Informativo*. Ano 1 nº 01, ULBRA.
- [25] Rubio, M.A., Romero-Zaliz, R., Mañoso, C., Madrid, A.P. (2014). Enhancing an introductory programming course with physical computing modules. *IEEE Frontiers in Education Conference (FIE) Proceedings*. 1019-1026.
- [26] Santos, R.P., 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. *INFOCOMP – Journal of Computer Science*. 5(1), 41-50.
- [27] Schultz, M. R. O. (2003). Metodologias para Ensino de Lógica de Programação de Computadores. Monografia de Especialização (Ciência da Computação). Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, Brasil. 69p.
- [28] Silva, I.F.A., Silva, I.M.M., Santos, M.S. (2009). Análise de problemas e soluções aplicadas ao ensino de disciplinas introdutórias de programação. *IX Jornada de Ensino, Pesquisa e Extensão - JEPEX*. Disponível em <www.eventosufupe.com.br/jepex2009/cd/resumos/r1479-1.pdf>.
- [29] SLHESARENKO, M.; GONÇALO, C. R.; BEIRA, J.C.; CEMBRANEL, P. (2014) A Evasão na Educação Superior para o Curso de Bacharelado em Sistemas de Informação. *Revista Gestão Universitária na América Latina (GUAL)*, Vol.7, Número 1. Disponível em:

<<https://periodicos.ufsc.br/index.php/gual/article/view/1983-4535.2014v7n1p128>>.

- [30] Yakimov, P.I. (2016). Open-source Platforms Application in Introductory Embedded Systems Teaching. *XXV*

International Scientific Conference Electronics (ET) (Vol1, No1, pp.1-4).