

## ENSINO DE PROGRAMAÇÃO ATRAVÉS DOS SISTEMAS EMBARCADOS

Ernesto de Souza Massa Neto<sup>1</sup>  
Giovani José Machado da Rocha<sup>2</sup>  
Leandro dos Santos Gonzalez<sup>3</sup>

UNIVERSIDADE DO ESTADO DA BAHIA (UNEB)

### RESUMO

As altas taxas de reprovação nas disciplinas de programação são preocupantes nos cursos superiores de graduação em informática. Aliado a esse problema, deficiências vindas do ensino médio nas disciplinas de ciências exatas só fazem com que esse quadro se torne ainda mais preocupante, gerando problemas nas habilidades necessárias para o desenvolvimento satisfatório dos alunos. Desse modo, esse estudo tem como objetivo desenvolver habilidade cunhada como “computational thinking”, ou o “pensamento computacional” com o auxílio dos sistemas embarcados. Para que essa pesquisa fosse realizada, efetuou-se um levantamento de dados para verificação dos índices de aprovação nas principais disciplinas em que tem a programação em sua estrutura, expondo um panorama geral do curso de sistemas de informação na Universidade do Estado da Bahia (UNEB – Campus II). A metodologia empregada foi baseada atividades práticas com a placa arduino. Os resultados mostraram como pode ser difícil a compreensão dos alunos de conceitos necessários para o desenvolvimento do profissional de informática, assim como a transmissão desse conteúdo por parte dos professores. Mostrou também que, o arduino possui grande potencial de utilização em sala de aula para o ensino de programação.

**Palavras-chave:** Sistemas Embarcados. Pensamento Computacional. Educação.

### ABSTRACT

The high reprobation rates in the programming disciplines are of concern in the undergraduate courses in computer science. In addition to this problem, shortcomings in high school in the exact sciences only make this picture even more worrying, generating problems in the skills necessary for the satisfactory development of students. Thus, this study aims to develop skills coined as "computational thinking" or "computational thinking" with the help of embedded systems. In order to carry out this research, a data survey was carried out to verify the approval indexes in the main disciplines in which it has the programming in its structure, presenting a general overview of the course of information systems at the State University of Bahia (UNEB - Campus II). The methodology employed was based on practical activities with the arduino board. The results showed how difficult it is to understand the students of concepts necessary for the development of the

<sup>1</sup> Doutor em Ciência da Computação – Alagoinhas/BA - massa.ernesto@gmail.com

<sup>2</sup> Graduado em Sistema de Informação – Alagoinhas/BA – giovanirocha2011@gmail.com

<sup>3</sup> Mestre em Modelagem Computacional – Salvador/BA – email.lgonzalez@gmail.com

computer professional, as well as the transmission of this content by the teachers. It also showed that arduino has great potential for classroom use for programming teaching.

**Keywords:** Embedded systems, Computational Thinking, Education

## 1 INTRODUÇÃO

O ensino do pensamento computacional principalmente nas disciplinas do eixo de programação nos cursos superiores tem sido tarefa bastante difícil. Fazer com que os alunos desenvolvam a capacidade de pensar de forma estruturada e encadeada tem sido um grande desafio, e aliado a essa realidade problemas como deficiências vindas do ensino médio nas disciplinas de exatas, baixa capacidade de abstração faz com que aumentem os índices de reprovação nas disciplinas de programação. Observando essa realidade, existe a necessidade de que seja elaborado uma nova abordagem de ensino para que sejam diminuídos esses problemas com os alunos dos cursos superiores de graduação.

O desenvolvimento da “computacional thinking” nos alunos vai além de criar códigos. A habilidade de gerar conceitos, estimular o raciocínio lógico, formar, disponibilizar novos modelos e torna-lo disponível são algumas contribuições que o pensamento computacional com o auxílio dos sistemas embarcados, ajudará a desenvolver nos alunos iniciantes a capacidade de resolução de problemas e ideias abstratas, aumentando a capacidade de percepção dos alunos.

A utilização dos sistemas embarcados tem se difundido de maneira significativa nos últimos anos. Muito utilizado em sistemas aviônicos, celulares, eletrodomésticos, equipamentos de redes de computadores (roteadores, switches e hubs). Na área educacional tem se destacado a utilização do arduino. O arduino se popularizou na robótica educacional por ser uma placa de baixo custo e baixa complexidade na criação de projetos por profissionais da área de tecnologia, hobistas e entusiastas, fazendo com que seja desenvolvida a capacidade de lidar tanto com hardware quanto com software em seus projetos, criando assim habilidades antes não desenvolvidas com métodos tradicionais de ensino.

Neste contexto, de que forma os sistemas embarcados podem melhorar a construção da habilidade cunhada como “computacional thinking”, ou o “pensamento computacional”? Como objetivo geral desse artigo é desenvolver habilidade cunhada como “computacional thinking”, ou o “pensamento computacional” utilizando

sistemas embarcados. Como objetivos específicos de implementar o projeto piloto de ensino, utilizar os sistemas embarcados nas aulas especiais, elaborar material didático específico, implementar aulas constantes no projeto educacional e elaborar relatórios sobre essas aulas especiais.

Sendo assim, o presente estudo pretende também auxiliar os métodos tradicionais de ensino de programação, ou seja, incluir nas atividades normais de sala de aula, atividades práticas que visem a idealizar o pensamento organizado para a solução dos eventuais problemas que surjam nas atividades profissionais dos alunos.

Busca-se com os sistemas embarcados ajudar o desenvolvimento da habilidade do pensamento computacional e com isso reduzir os índices de reprovação nas disciplinas de programação dos alunos do curso de sistemas de informação da Universidade do Estado da Bahia (UNEB) – Campus II.

Como resultado principal, é esperado que os alunos do curso de sistemas de informação adquiram maior expertise nas disciplinas do eixo de programação, obtendo a habilidade rotulada como "Computational Thinking".

O artigo é composto de uma introdução discutindo suas principais dificuldades e problemas que motivaram a pesquisa, onde é explanado o objetivo geral e objetivos específicos. Na seção 2 a 5 estará a fundamentação teórica, sendo que na seção 2 é abordado o pensamento computacional ou "Computational Thinking", na seção 3 o ensino da programação no ensino superior, na seção 4 será mostrado o que é um sistema embarcado, na seção 5 foi discutido o que é um arduíno e alguns modelos de placas finalizando a fundamentação teórica.

Na seção 6 é mostrado como é a metodologia aplicada na pesquisa, universo, população amostral, coleta de dados para que seja avaliado em que patamar se encontram os alunos, principais dificuldades e expectativas quanto ao aprendizado dessa técnica de desenvolvimento do pensamento computacional. Na seção 6.1 descreve como foram realizadas as atividades práticas e os objetivos dessas práticas na aprendizagem das estruturas de programação.

Na seção 7 está localizada a análise dos resultados onde é discutido os resultados encontrados na aplicação das aulas práticas e a aplicação dos questionários e na seção 8 está a conclusão que estará evidenciado os pontos mais relevantes, as contribuições do estudo e possíveis trabalhos futuros. Logo após estarão as referências e o apêndice.

## 2 O PENSAMENTO COMPUTACIONAL OU “COMPUTACIONAL THINKING”

O desenvolvimento profissional em qualquer área do conhecimento requer habilidades que estão ligadas com o raciocínio lógico. Mais especificamente nas áreas de engenharia e computação de forma geral, os profissionais dessas áreas precisam certos níveis de abstração, pois não é somente aprender estruturas lógicas e programar, é preciso que haja interdisciplinaridade de conhecimentos para que possam surgir soluções para os mais variados problemas. Segundo Wings (2008, p. 3717)

A essência do pensamento computacional é abstração. Na computação, abstraímos noções além das dimensões físicas do tempo e do espaço. Nossas abstrações são extremamente gerais porque são simbólicas, onde as abstrações numéricas são apenas um caso especial.

Sendo assim, é importante observar que para o desenvolvimento do pensamento computacional é necessário que os profissionais não só das áreas de tecnologia mais de todas as áreas estimulem em tarefas diárias a resolução de problemas fazendo com que se desenvolva o processo criativo criando assim novas habilidades. Segundo (Wings, 2006, p. 33-35), o pensamento computacional tem as seguintes características:

- a) conceituando, não programando: se refere a dar mais importância do desenvolvimento de conceitos, fazendo análises necessárias evitando estar somente focado no desenvolvimento de códigos;
- b) habilidade fundamental, não mecânica: buscar desenvolver o raciocínio lógico, não focando somente na mecanização do trabalho;
- c) uma maneira que os humanos, não os computadores, pensam: é dar ênfase em como os seres humanos desenvolvem seu pensamento. As máquinas apenas são abstrações humanas, humanos não devem pensar como máquinas, mas sim máquinas representar pensamentos humanos;
- d) complementa e combina o pensamento matemático e de engenharia: é importante trabalhar a interdisciplinaridade das ciências. Nesse caso específico teorias matemáticas já desenvolvidas servem de apoio para criação de novos conceitos;
- e) ideias, não artefatos: as soluções para os problemas são importante, porém as novas ideias e modelos são o mais importante.

- f) para todos, em todo lugar: o desenvolvimento tem que estar disponível para a utilização de todos, não restrito para apenas um grupo de pessoas.

Pode-se afirmar, portanto, que com essas características, o mais importante para o pensamento computacional não está restrito apenas nas ciências exatas. É estimular o processo criativo, incentivar a resolução de problemas, a concepção de novos conceitos, a geração de valor para todas as áreas do conhecimento é de fundamental importância para qualquer área do conhecimento.

### **3 O ENSINO DA PROGRAMAÇÃO NO ENSINO SUPERIOR**

O desafio de desenvolver o pensamento computacional esbarra no problema das instituições de ensino em reduzir os altos índices de reprovação nas disciplinas que requerem programação. Mas para que ocorra uma mudança efetiva nesse quadro preocupante, se faz necessário entender quais os motivos que levam a maioria dos alunos sentir tanta dificuldade na compreensão dos principais conceitos computacionais. Segundo Coutinho, Lima e Santos (2017, p.2), afirmam que:

Na literatura sobre ensino introdutório de programação existe uma longa lista de motivos que impactam no aprendizado dos conceitos ligados não só à programação de computadores, mas de dispositivos computacionais de maneira geral, tais como: falta de conhecimentos básicos do ensino médio, falta de motivação, falta de maturidade do aluno, professores despreparados e exercícios teóricos ou inadequados, entre outros.

Pode-se observar que o conhecimento acumulado durante os anos escolares nos ensinos fundamental e médio nas disciplinas da área das ciências exatas são importantes quando esses alunos forem para o ensino superior com uma base sólida compreendendo os conceitos computacionais sem grandes dificuldades. Ainda Coutinho, Lima e Santos (2017, p.2), afirmam que:

Aprender a escrever programas de computador, seja utilizando componentes que se encaixam formando uma sentença lógica ou codificando de maneira textual, não é uma tarefa fácil para quem está iniciando e, principalmente, para quem não possui uma base consistente em matemática e lógica. A programação, entretanto, integra cada vez mais currículos de cursos de graduação que, tradicionalmente, não a teriam como disciplina do curso, como é o caso das Engenharias e cursos de Computação.

Então se faz necessário uma reflexão de como é abordado o conteúdo de programação nos cursos superiores de educação. Observa-se também que o ensino fundamental e médio não está preparando os alunos para as carreiras de engenharia e computação. Ensinar a programar computadores da forma tradicional mostra-se que é um modelo superado e novas alternativas de ensino devem ser testadas para que se atraiam mais jovens para essas carreiras.

Para que as aulas se tornem mais atrativas, o uso de sistemas embarcados nas aulas de programação pode ser de grande valia, pois pequenas práticas em sala de aula podem dinamizar o processo de aprendizagem dos alunos incentivando assim a criatividade dos alunos trazendo problemas reais para que sejam desenvolvidos em sala de aula.

#### **4 SISTEMAS EMBARCADOS**

Sistemas embarcados são dispositivos que integram hardware e software para uma solução que pode ser de baixo custo ou uma solução mais robusta e consequentemente de alto custo. Segundo Almeida, Moraes e Seraphin (2016, p.5), afirmam que:

Sistemas embarcados são sistemas eletrônicos microprocessados que, após serem programados, possuem uma função específica que geralmente não pode ser alterada. Uma impressora, por exemplo, mesmo possuindo um processador que poderia ser utilizado para qualquer tipo de atividade, tem funcionalidade restrita apenas a impressão de páginas, um computador de propósito geral, no entanto, pode ser utilizado no instante como um ambiente de entretenimento, em outro como estação de trabalho, ou até mesmo o telefone.

Portanto, a utilização de um sistema embarcado pode variar muito da necessidade do consumidor. Geralmente possuem baixo custo, já vindo integrado com software próprio e objetivos bem definidos. Segundo Almeida, Moraes e Seraphin (2016, p.5), afirmam que:

Outra característica da maioria dos sistemas embarcados é a restrição de recursos, tanto computacionais: memória e processamento, quanto físicos: números de terminais e interfaces de exibição/entrada de dados. Grande parte desta restrição se deve as questões de custo, consumo de energia e robustez. Isso leva ao desenvolvimento de alguns sistemas que não possuam nenhum botão ou luz indicativa.

O exemplo mais comum hoje é o smartphone. Existem diversos modelos, várias marcas concorrentes e os mais variados preços, porém pode-se afirmar que o hardware possui algumas características em comum: esses aparelhos são miniaturizados, possuem uma capacidade tanto de processamento como de armazenamento limitados e tem um software específico feito para funcionar perfeitamente nos aparelhos, fazendo com que se encaixe na categoria de embarcados.

A restrição de recursos tanto físicos quanto computacionais dos sistemas embarcados é a característica marcantes desses tipos de dispositivos. Essas restrições não são vistas como pontos negativos, mas sim como forma de explorar a capacidade máxima dos recursos, limitação de agentes externos a recursos, pois muitos embarcados possuem uma única funcionalidade, e não é desejável o acesso de terceiros a esses sistemas.

Na educação superior é uma prática muito difundida principalmente nos cursos de engenharia são atividades práticas com placas microprocessadas. Um componente muito utilizado nessas praticas é o arduíno. Com essa placa o aluno desenvolve conceitos fundamentais de programação e também aprende com a prática fundamentos de eletrônica, elétrica e mecatrônica.

## 5 ARDUÍNO

Arduíno é sem dúvida quando se fala em sistemas embarcados, a placa microprocessada muito difundida em ambientes acadêmicos e também por todos os aqueles que desejam desenvolver projetos que envolvam eletrônica e programação. O arduíno é o que chamamos de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software. (MCROBERTS, 2011)

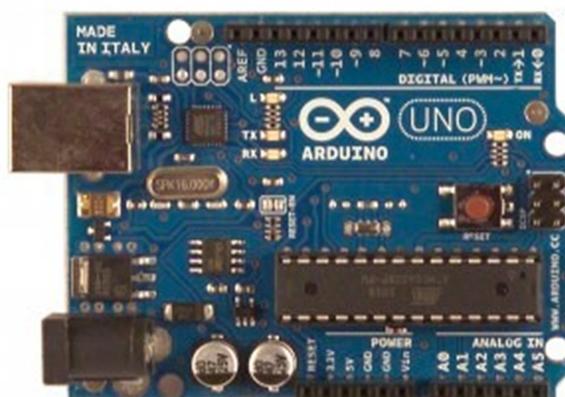
A popularização do arduíno se deu pela facilidade de integração de hardware e software em projetos. Além disso, o custo de aquisição dessa placa é baixo comparado com soluções de mesmo valor tecnológico disponíveis no mercado para aquisição. Segundo Evans, Noble e Hockenbaum (2013, p.26) afirmam que:

A primeira placa foi baseada no ATmega8 rodando a uma velocidade de clock de 16 MHz com memória flash de 8 KB; mais tarde, placas tais como a Arduino NG plus e a Diecimila (nome italiano para 10.000) usava o ATmega168 com memória flash de 16 KB. As versões mais recentes do

Arduino, Duemilanove e Uno, usam o ATmega328 com memória flash de 32 KB e podem comutar automaticamente entre USB e corrente contínua (DC). Para projetos que exigem mais Entrada/Saída e memória, há o Arduino Mega1280, com memória de 128 KB, ou o mais recente Arduino Mega2560, com memória de 256 KB.

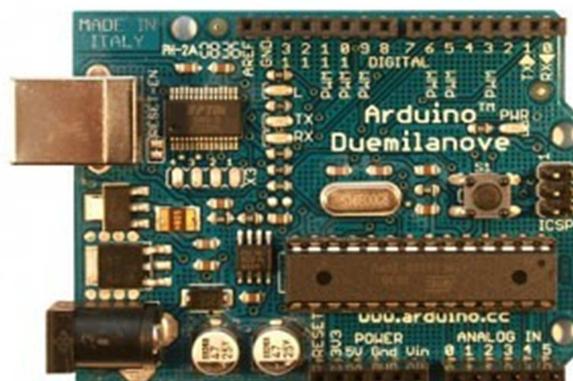
São alguns exemplos de placas: Arduíno Uno, Arduíno Duemilanove, Arduíno Ethernet, Arduíno Mega e outros, são alguns modelos disponíveis para aquisição.

**Figura 1 - Arduíno Uno**



Fonte: Embarcados (2018).

**Figura 2 – Arduíno Duemilanove**



Fonte: Embarcados (2018)

Referente ao software de integração do arduíno com o computador, existe uma Integrated Development Environment (IDE) disponível tanto para os sistemas operacionais Windows, MAC e Linux, aumentando ainda mais a sua versatilidade.

## 6 METODOLOGIA

Na metodologia está sendo abordado como realizou-se a pesquisa e como foi desenvolvida as aulas práticas, portanto, a classificação da pesquisa pode ser de dois tipos segundo VERGARA (1990): quanto aos fins e quanto aos meios. O projeto de ensino de programação através de sistemas embarcados é classificado quanto aos fins como: Metodológica, Aplicada, Explicativa e Descritiva. E em relação aos meios a pesquisa é uma: pesquisa de campo, documental, estudo de caso e bibliográfica.

O primeiro ponto a ser abordado são os fins metodológicos. Inicialmente realizou-se a aplicação do primeiro questionário para ter uma visão geral do nível dos alunos em programação. Logo após se desenvolveu uma atividade prática em sala de aula que buscou-se trabalhar com os alunos participantes as principais estruturas de programação com o auxílio do arduino. Logo após a aula aplicou-se outro questionário para a verificação do método de ensino baseado em atividades práticas. Também houve a introdução de alguns conceitos de eletrônica para que os alunos conseguissem realizar essas atividades.

O segundo ponto é quanto sua aplicação. É aplicada porque fez-se pelas dificuldades dos alunos no aprendizado da programação, visando desenvolver um modo em que se torne mais fácil a absorção dos conceitos e coloca-los em prática com o auxílio dos sistemas embarcados.

O terceiro ponto é relativo à sua explicação. A pesquisa foi explicativa porque quer fazer do ensino de programação algo palpável, não somente com aulas expositivas que muitas vezes entediam o aluno, mas sim ultrapassando as barreiras das provas escritas como único mecanismo de avaliação. Quis mostrar que é possível o aprendizado de maneira intuitiva e fácil compreensão.

O quarto ponto refere-se a sua descrição. Foi descritiva porque discorreu sobre as características de uma população bem conhecida, os alunos do curso de sistema de informação do campus II – Alagoinhas, buscando melhorar a capacidade de desenvolvimento do pensamento computacional.

Quanto aos meios de investigação, o primeiro ponto a ser abordado é pesquisa de campo. Realizou-se a execução de práticas durante as disciplinas de programação com atividades preparadas respeitando o nível de profundidade dos alunos em cada semestre.

O segundo ponto é documental. Fez-se um levantamento de dados com o auxílio do colegiado de sistemas de informação, coletando dados como índices de aprovação e reprovação desde a fundação do curso dando um panorama geral do aproveitamento dos estudos envolvendo programação.

O terceiro ponto é de estudo de caso. Foi um estudo de caso porque se limita a somente a um curso pertencente ao departamento de ciências exatas e da terra.

O quarto ponto é referente a bibliografia. Essa pesquisa utilizou livros, periódicos, revistas e a internet com o intuito de embasar cientificamente a pesquisa e de coletar o máximo de informações pertinentes para o estudo.

O universo estudado tratou-se pelos alunos do curso de sistemas de informação da Universidade do Estado da Bahia do Campus II - Alagoinhas. A amostra foram os alunos os alunos matriculados nas disciplinas de programação em 2018.1. Ainda relativo a amostra da pesquisa, podem ser probabilísticas e não-probabilísticas. Para o presente estudo, utilizou-se a pesquisa não-probabilística, pois fez-se uma escolha deliberada pelos alunos que cursaram as disciplinas de programação. Características como acessibilidade e tipicidade são bastante relevantes para a pesquisa.

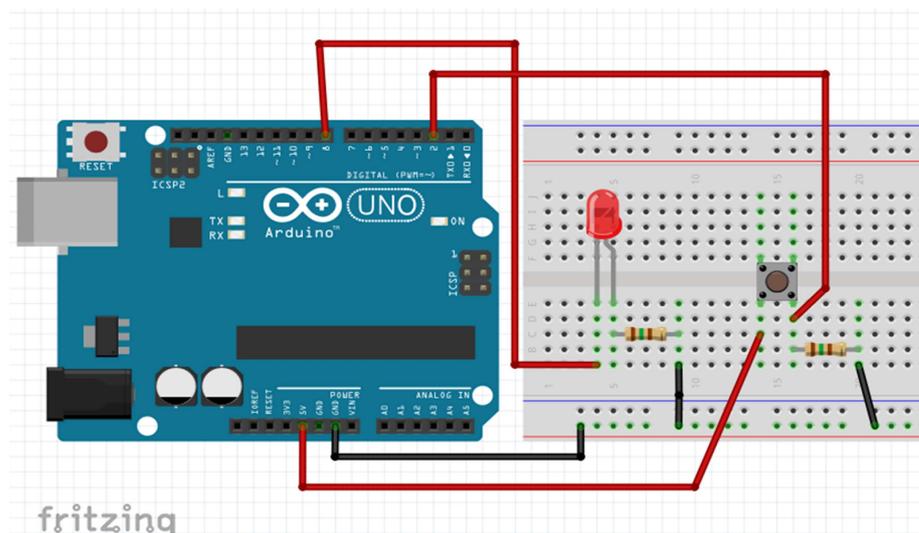
Essas características foram selecionadas em decorrência do universo. A facilidade de acesso ao público alvo e a sua representatividade foram determinantes para o projeto de pesquisa. Por isso tangeu os alunos matriculados nas disciplinas de programação do curso sistemas de informação do campus II – Alagoinhas.

Outro fator de grande relevância é de que forma como ocorreu a coleta de dados. Para essa pesquisa, realizou-se a obtenção dos dados por meio de formulários antes da aplicação das práticas e depois da aplicação para a verificação da efetividade e se o método era relevante para o aprendizado dos alunos.

## 6.1 APLICAÇÃO DA ATIVIDADE PRÁTICA

Para avaliar o modelo desenvolveu-se uma atividade prática que onde aconteceu da seguinte forma: No início da aula realizou-se uma pequena introdução sobre eletrônica e seus componentes, dentre eles podemos citar, o diodo emissor de Luz (LED), resistores, fios de ligação, protoboard (para a montagem do circuito



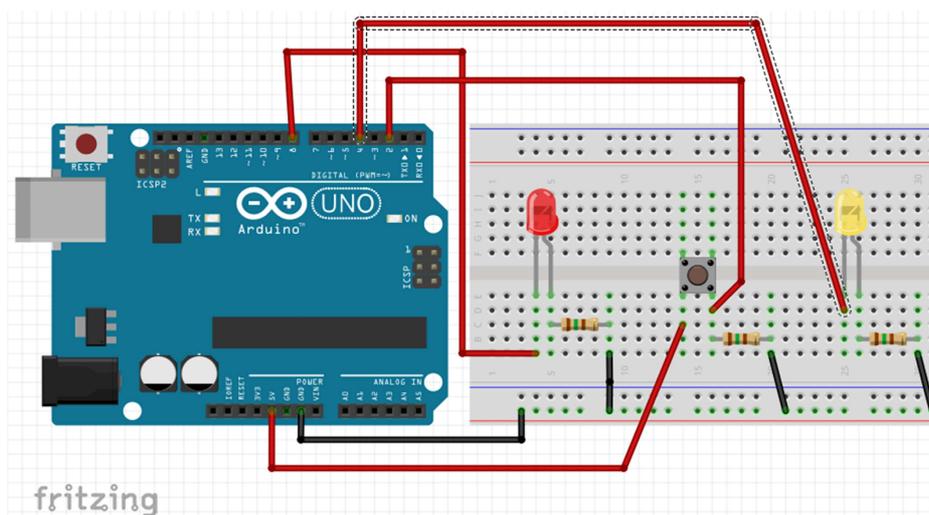


Fonte: Autoria própria (2018)

Nessa segunda prática, foram abordadas as seguintes estruturas para o funcionamento do circuito: criação de variáveis e programação sequencial e estrutura de decisão. O objetivo dessa prática mostrou como funciona uma estrutura de decisão na prática. No exemplo acima toda vez que o aluno acionou o botão, o led vermelho acendeu. A estrutura de decisão caracteriza-se por executar um trecho de código caso uma condição seja satisfeita. No caso da prática, a condição de execução era se o botão fosse acionado então acendia o led vermelho.

Na terceira prática inseriu-se ao circuito já existente um led de cor amarelo. O objetivo dessa prática era toda vez que pressionasse o botão acendia o led vermelho e logo após o led amarelo piscasse 5 vezes. Segue abaixo como foi feita essa prática

**Figura 5 – Prática 3 – Acende Led e pisca Led amarelo**



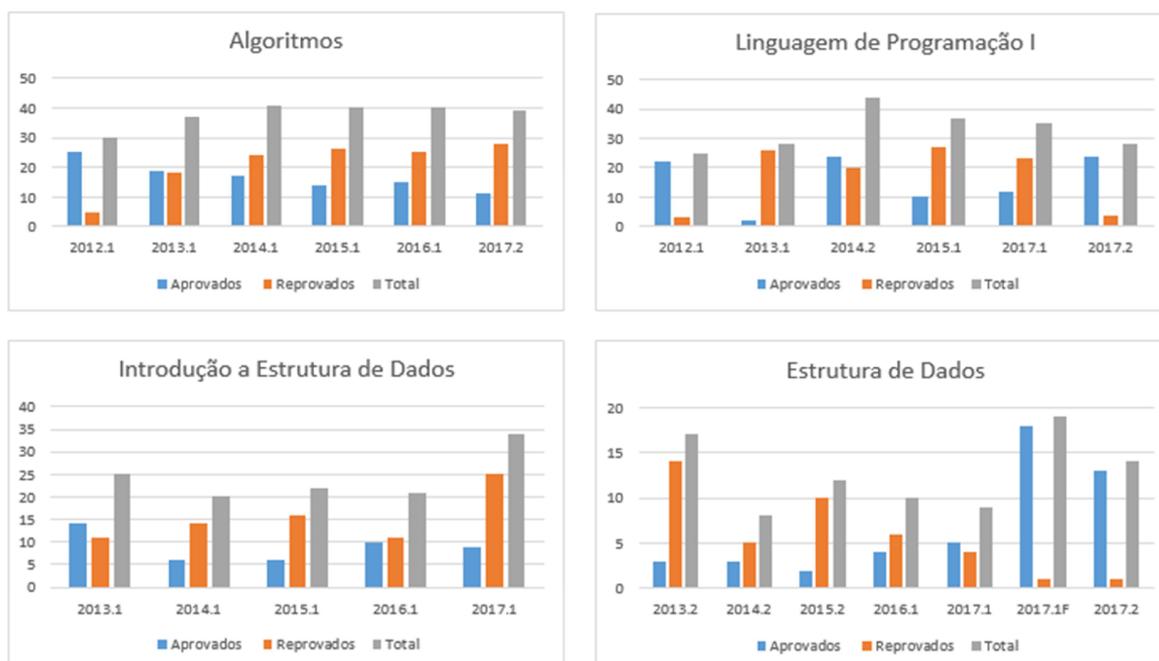
Fonte: Autoria própria (2018)

Nessa terceira prática, foram abordadas as seguintes estruturas para o funcionamento do circuito: criação de variáveis e programação sequencial, estrutura de decisão e estrutura de repetição. Já nessa prática, ficou caracterizada pela introdução da estrutura de repetição no circuito. Toda vez que o aluno acionava o botão acendia o led vermelho e logo após dentro da estrutura de decisão foi introduzido uma estrutura de repetição para ligar e desligar o led amarelo 5 vezes. A estrutura de repetição caracteriza-se por uma repetir um mesmo trecho de código sem a necessidade de reescreve-lo por muitas vezes seguidas.

## 7 ANÁLISE DE RESULTADOS

A proposta deste artigo é investigar se o uso dos sistemas embarcados poderá auxiliar a desenvolver a habilidade cunhada como “computacional thinking”, ou o “pensamento computacional”. Para que isso seja possível, foi feito um levantamento das principais disciplinas de programação do curso de sistemas de informação da Universidade do Estado da Bahia (UNEB – Campus II) para dar um panorama geral dos índices de aprovação e reprovação dos anos de 2012.1 a 2017.2 das disciplinas de Algoritmos, Linguagem de Programação I, Introdução a Estrutura de Dados e Estrutura de Dados. Segue abaixo o gráfico feito com esses dados.

**Gráfico 1 - Coleta de dados das disciplinas**



Fonte: Autoria Própria (2018)

Após análise dos gráficos pode-se afirmar que: A disciplina de algoritmos ao longo dos anos apresentados, vem crescendo substancialmente os índices de reprovação. A taxa de reprovação no semestre de 2012.1 foi de 16.6%, no semestre de 2013.1 foi de 48,64% quase metade da turma foi reprovada, no semestre de 2014.1 esse índice subiu para 58,53%, ou seja, já nesse semestre as reprovações foram maiores que as aprovações. No semestre de 2015.1 a taxa subiu novamente alcançando 65% da turma, no semestre de 2016.1 uma pequena redução para 62,5, mas mesmo assim a maioria da turma foi reprovada e no semestre de 2017.2 alcançou 28.2% de aprovação na turma.

A disciplina de linguagem de programação I sofreu variações importantes no decorrer dos anos. No semestre 2012.1 obteve um percentual de aprovação de 88%, mas já no semestre seguinte 2013.1 sofreu uma queda bastante significativa passando a aprovação para apenas 7,14%. No semestre de 2014.2 houve um equilíbrio nas taxas de aprovação e reprovação da disciplina. A aprovação ficou em torno de 54,54% e a reprovação com 45,45%. Nos anos seguintes 2015.1 e 2017.1 respectivamente, a aprovação foi para 27,02% e 34,28% causando também altas taxas de reprovação nessa disciplina. A disciplina só voltou a apresentar bons índices de aprovação no semestre 2017.2 onde a taxa voltou a 85,71% de aprovação na turma desse semestre.

A disciplina Introdução a Estrutura de Dados também sofreu variações no decorrer dos semestres. No semestre 2013.1 obteve uma aprovação maior que a reprovação ficando o índice de 56% e 44% respectivamente. Já no semestre 2014.1 obteve uma aprovação de apenas 30% dos estudantes matriculados na disciplina. No semestre de 2015.1 mais uma vez o índice de aprovação foi baixíssimo, tendo apenas 27,27% de aprovação. No semestre de 2016.1 foi onde houve um maior equilíbrio entre aprovações e reprovações na disciplina. 47,61 foram aprovados e 52,38% dos alunos foram reprovados, mesmo assim as reprovações foram maiores que as aprovações. No semestre de 2017.1 novamente o índice de aprovações foi baixo. Cerca de 26,47% dos alunos conseguiram êxito e concluíram com sucesso a disciplina.

E por último foi analisada a disciplina estrutura de dados. Além de obter os piores índices de aprovação do curso, essa disciplina ainda possui uma quantidade

bastante baixa de alunos matriculados, exceto no curso de férias oferecido pela universidade. No semestre de 2013.2 somente 17,64% dos alunos foram aprovados, tendo uma taxa de reprovação de 82,35%. No semestre de 2014.2 a aprovação subiu para 37,5%, mas mesmo assim um índice muito baixo de aproveitamento de alunos na matéria. No semestre de 2015.2 houve a pior taxa de aprovação da disciplina, 16,6% de alunos aprovados, uma expressiva taxa de reprovação de 83,33% na matéria. No semestre de 2016.1 a taxa de aprovação voltou a subir para 40% da turma. A partir dos semestres seguintes, que são 2017.1(Curso de Férias), 2017.1 e 2017.2 as taxas de aprovação foram respectivamente: 94.73%, 55.55% e 92,85%.

Observa-se após a análise desses dados que podem ocorrer diversos fatores para que ocorram tanta discrepância entre um curto espaço de tempo nas disciplinas avaliadas. Dentre esses fatores pode-se citar: As metodologias empregadas pelos professores podem contribuir de forma bem significativa para que se varie tanto as aprovações das reprovações de um semestre. Níveis de dificuldade na forma de cobrança nas aulas e nas avaliações podem refletir nesses índices causando mudanças tão significativas de um semestre para outro.

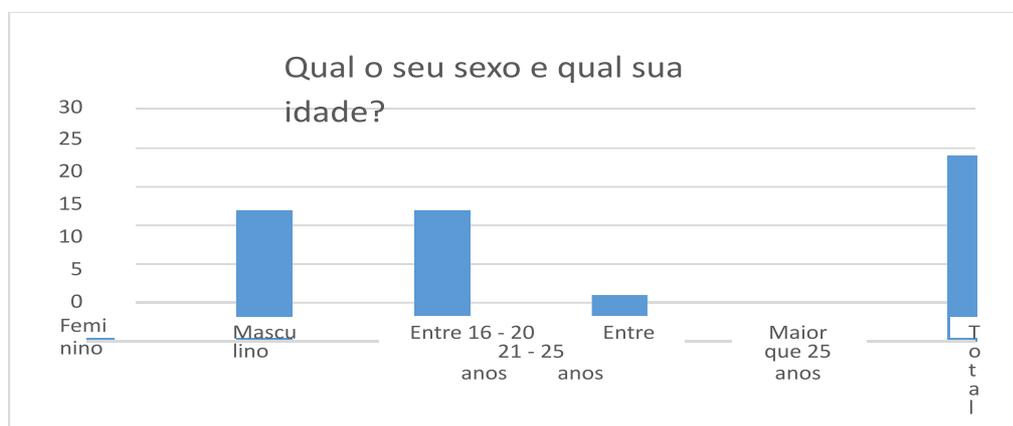
A falta de preparo dos alunos ingressantes com deficiências vinda dos anos escolares nas matérias de exatas, a falta de interesse perante as dificuldades encontradas, a forma de relacionamento do professor com as turmas pode também influenciar no desempenho geral dos alunos.

Após essa análise foi elaborada uma metodologia baseada no ensino de programação. Essa forma é utilizando uma placa de microprocessada chamada Arduíno Uno, onde foi feito algumas práticas em sala de aula com alunos das disciplinas de Linguagem de Programação I, Introdução a Estrutura de Dados e Estrutura de Dados. Esse experimento não foi feito na turma de algoritmos porque essa disciplina não foi oferecida no semestre em curso 2018.1.

Foi aplicado um questionário antes da aplicação prática que teve como objetivo ter uma ideia geral do estudante e algumas questões que visavam ter uma noção geral de sua proximidade com a programação. Logo após foi feita uma aula prática com o auxílio da placa arduíno uno. Depois da aula foi aplicado outro questionário para saber se os alunos gostariam se esse modelo fosse implantado como auxílio das aulas de programação.

A amostragem total para essa pesquisa foi de 24 alunos. A princípio foi questionado aos participantes da pesquisa sobre seu sexo e sua faixa etária, conforme o gráfico abaixo:

**Gráfico 2 - Levantamento do sexo e idade dos participantes**

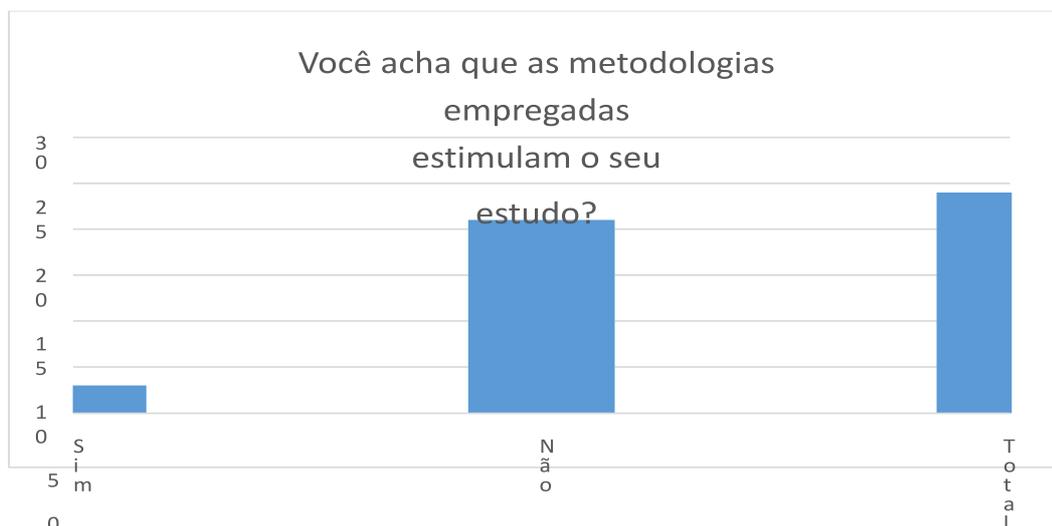


Fonte: Autoria Própria (2018)

Mostra-se com isso uma predominância masculina no curso e formado essencialmente por jovens.

Foi abordado acerca das metodologias aplicadas em sala de aula pelos professores como influenciariam e estimulavam o estudo. Observe o gráfico abaixo

**Gráfico 3 – Avaliação dos alunos se as metodologias empregadas estimulam o seu estudo**

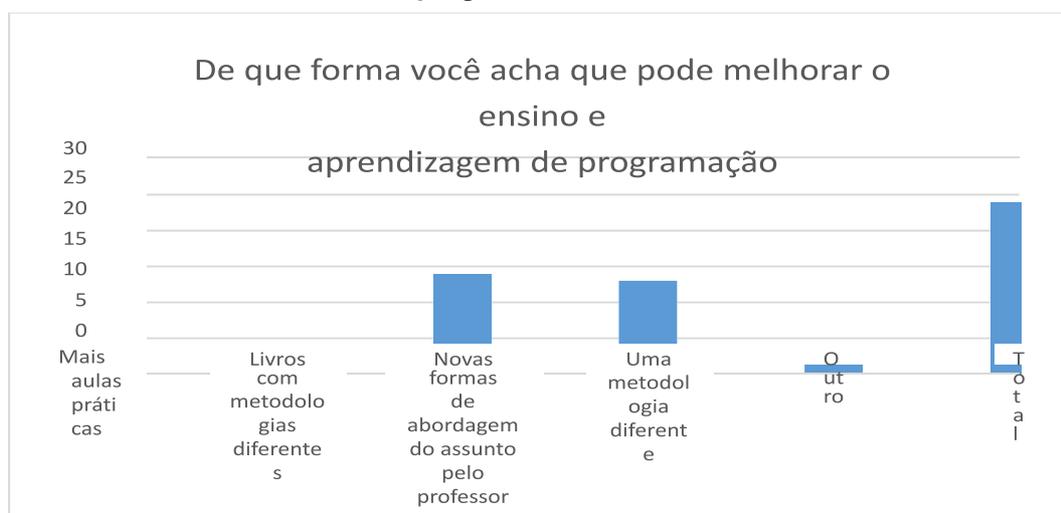


Fonte: Autoria Própria (2018)

Conforme Júnior e Rapkiewicz (apud Rodrigues 2002, Schultz 2003, Chaves de Castro et al 2003, Delgado et al 2004), “O nível de dificuldade no processo faz com que haja baixa motivação dos alunos, apatia, baixa autoestima culminando com muita evasão e reprovação”. Com isso observa-se que são muitas as dificuldades encontradas no ensino das disciplinas de programação.

Foi questionado aos participantes como contribuir para uma melhor no ensino e aprendizagem da programação. Para esse quesito foi permitido ao aluno marcar mais de uma alternativa caso ele achasse necessário. Observe o gráfico abaixo

**Gráfico 4 - De que forma você acha que pode melhorar o ensino e aprendizagem de programação**

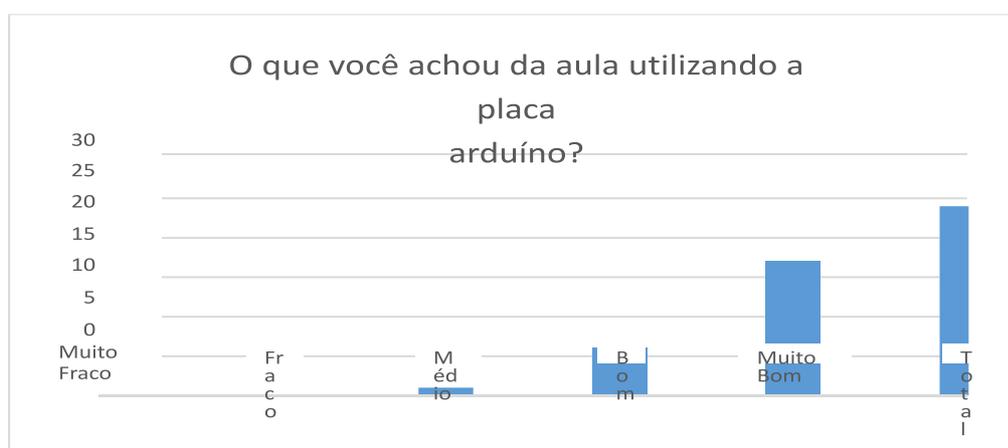


Fonte: Autoria Própria (2018)

Após essas respostas dessa questão é possível verificar que há uma insatisfação com o atual modelo de ensino – aprendizagem utilizado no curso. É necessário desenvolver novas técnicas para estimular o aprendizado dos alunos e fazer com que conceitos computacionais sejam difundidos de maneira mais clara para que seja motivado o aprendizado dos alunos.

Foi perguntado aos alunos o que eles tinham achado do ensino com a utilização da placa arduino.

**Gráfico 5 – O que você achou da aula utilizando a placa arduino?**

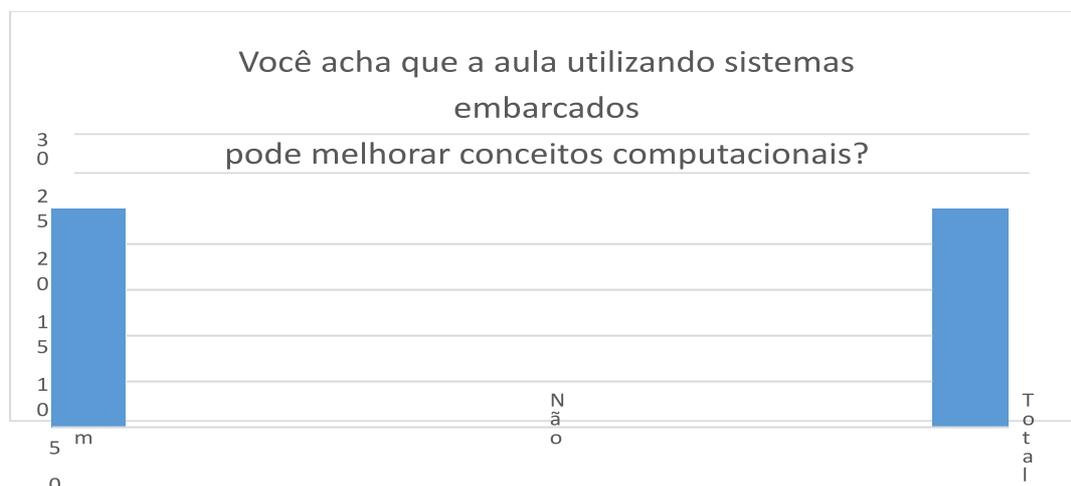


Fonte: Autoria Própria (2018)

Pode-se afirmar que a receptividade dos alunos foi muito boa com a utilização da placa arduino uno. Ainda conforme Moreira et al (2017, p.4) afirma que, “o arduino ajuda aos alunos aprenderem matérias consideradas complexas de maneira interessante com aplicações reais, além de despertar o interesse em aprender o próprio arduino. ”

Foi questionado se os sistemas embarcados podem ajudar no desenvolvimento de conceitos computacionais:

**Gráfico 6 - Você acha que a aula utilizando sistemas embarcados pode melhorar conceitos computacionais?**

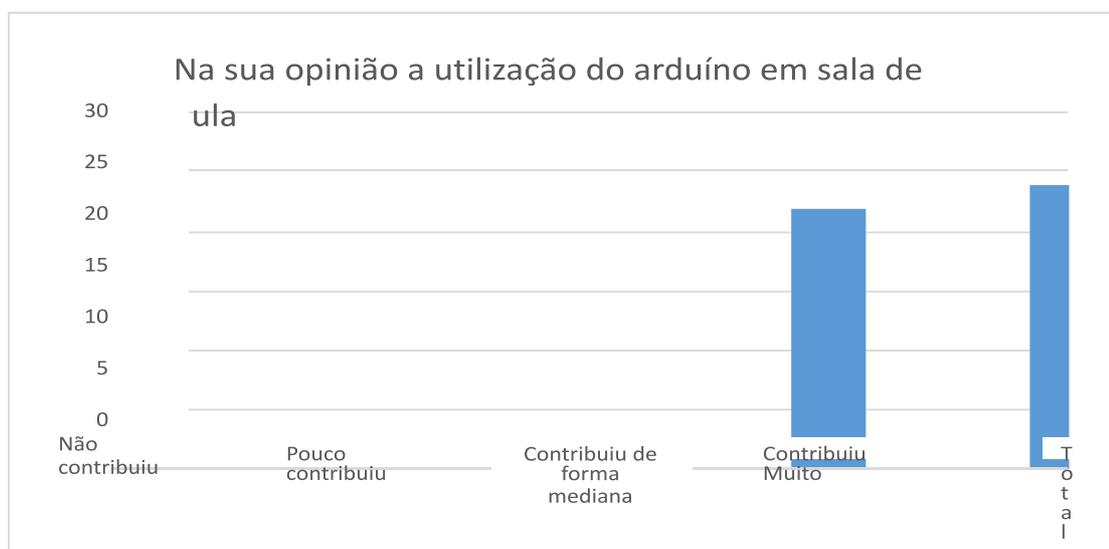


Fonte: Autoria Própria (2018)

Essa alternativa mostra como é nítida a aceitação dos alunos pesquisados para a utilização do arduino. 100% dos alunos acreditam que os conceitos trabalhados em sala de aula como, programação sequencial, estruturas de decisão e repetição podem ser melhorados com a utilização do arduino.

Outra questão relevante foi sobre a introdução do arduino em sala de aula. O gráfico abaixo mostra o resultado:

**Gráfico 7 - Na sua opinião a utilização do arduino em sala de aula**



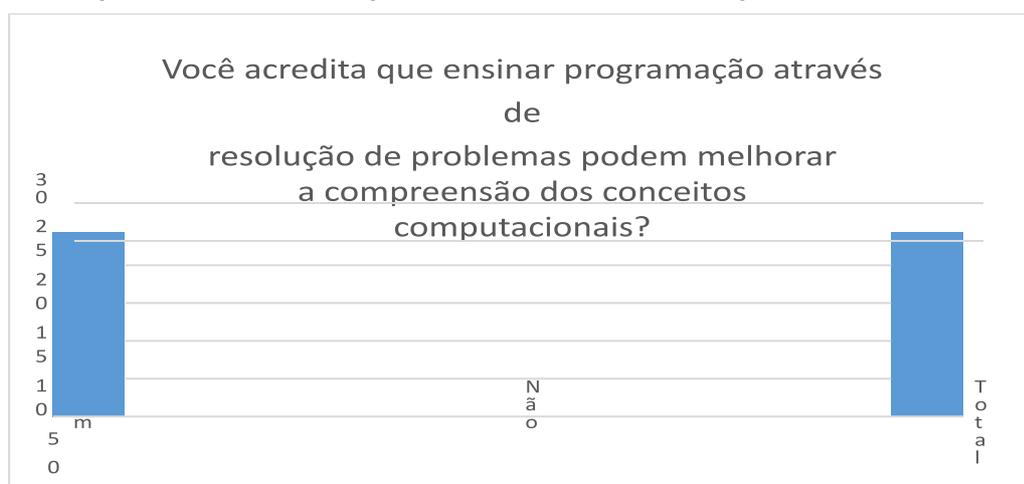
Fonte: Autoria Própria (2018)

O arduino utilizado em sala de aula conforme os dados acima, pode ser uma ferramenta muito importante no ensino de programação. 22 alunos responderam que a sua utilização em sala de aula contribuiu muito para a sua aprendizagem.

Portanto, a sua utilização pode ser um facilitador e estimulante no ensino em sala de aula.

E finalizando última questão foi perguntado se o ensino da programação através de resolução de problemas poderia melhorar a compreensão dos conceitos computacionais:

**Gráfico 8 - Você acredita que ensinar programação através de resolução de problemas pode melhorar a compreensão dos conceitos computacionais?**



Fonte: Autoria Própria (2018)

Ainda segundo Junior e Rapkiewicz (apud Lemos, Barros e Lopes, 2003), “existe uma forte indicação de que os conhecimentos anteriores sobre a resolução de problemas dentro do contexto anterior à experiência do estudante com a informática, torna-se um pré-requisito importante no aprendizado de técnicas de programação.”

Essa segunda avaliação mostra uma que é possível avançar no processo ensino aprendizagem, bastando que assim que os professores busquem novas metodologias para assim estimular o aprendizado dos alunos.

## 8 CONSIDERAÇÕES FINAIS

Observa-se que são muitos fatores que influenciam no desenvolvimento de uma nova metodologia de ensino. Desenvolver uma nova mentalidade e fazer com que seja aceita é uma tarefa bastante difícil e complexa. Com objetivo principal do trabalho foi desenvolver habilidade cunhada como “computacional thinking”, ou o

“pensamento computacional” utilizando sistemas embarcados, foi satisfatório no que se diz respeito a aceitação do arduíno em sala de aula como recurso para auxiliar a aprendizagem, fato que pode ser notado nas análises de resultados nas questões que abordaram a utilização da placa em sala de aula e nos questionamentos relativos a o auxílio na resolução de problemas.

Porém, a hipótese lançada na introdução era que os alunos do curso de sistemas de informação adquiram maior expertise nas disciplinas do eixo de programação, obtendo a habilidade rotulada como "Computational Thinking", não se pode afirmar com exatidão que os alunos que realizaram essas práticas desenvolveram as principais características do pensamento computacional por que foi realizada apenas uma aula em cada turma, O tempo foi insuficiente para o desenvolvimento de tantas competências que se é exigida na "Computational Thinking".

A utilização da placa arduíno foi de fundamental importância porque para sua utilização foi necessário que o aluno se envolvesse com a prática aplicada na sala de aula, buscado também novos conhecimentos para que se tenha êxito em suas tarefas. Quando o aluno monta um pequeno circuito e depois desenvolve um código que faz esse circuito funcionar, desperta um desejo logo de imediato de entender o funcionamento, fazer outras alterações e ver essas modificações funcionando na prática, fascinando desde os iniciantes, até os programadores experientes. As aulas práticas só incentivaram os alunos a compreenderem o quanto é importante saber as estruturas básicas de programação.

A utilização dos índices de aprovação e reprovação no curso, mostram como a metodologia aplicada por cada professor interfere diretamente nesses resultados encontrados. Algumas disciplinas investigadas mudaram drasticamente os seus índices de aprovação e reprovação de um semestre para o outro, fato que pode ser atribuído a metodologia empregada pelo professor em sala de aula ou mesmo dificuldades dos alunos vindo com deficiências do ensino médio.

Através dos questionários aplicados foi possível mensurar a importância de uma metodologia bem aplicada no ensino da programação. Relativo a placa arduíno uno tem muito a ser explorada e com certeza pode ser bastante útil para o desenvolvimento de novos profissionais de informática.

Ainda no que se refere ao resultado principal da pesquisa, as atividades desenvolvidas em sala de aula trabalharam com conceitos básicos de programação,

dentre eles podemos destacar: criação de variáveis, programação sequencial ou seja execução do programa linha a linha, estruturas condicionais, estruturas de repetição e funções nativas da IDE.

Sendo assim, pode-se afirmar que esse modelo de aula poderá ser aplicado com mais frequência e elaborado de acordo com a necessidade do assunto desejado, contribuindo de forma eficiente nas disciplinas do eixo de programação. A aceitação do modelo de ensino-aprendizagem pode-se aferir no segundo questionário quando todos os alunos responderam “sim” quanto ao questionamento se as práticas poderiam ajudar no desenvolvimento dos conceitos computacionais.

Como proposta para trabalhos futuros se faz necessário mais atividades em sala de aula, com uma frequência regular para que se prove a eficiência com exatidão desse método de ensino, para que seja desenvolvida todas as características necessárias para alcançar resultados expressivos e aumentar a complexidade das atividades, inserindo outras estruturas como vetores, matrizes, funções que são tão importantes para os estudantes.

## REFERÊNCIAS

ALMEIDA, Rodrigo Maximiano Antunes; MORAES, Carlos Henrique Valério de; SERAPHIM, Thatyana de Farias Piola. **Programação de Sistemas Embarcados: Desenvolvendo software para microcontroladores em linguagem C**. Elsevier, 2016.

COUTINHO, Emanuel Ferreira; LIMA, Ernesto Trajano de; Santos, Clemilson Costa. **Um panorama sobre o desempenho de uma disciplina inicial de programação em um curso de graduação**. Revista Tecnologias da educação, Ano 9, vol. 19, Julho de 2017.

EVANS, Martin; NOBLE, Joshua; Hochenbaum, Jordan. **Arduíno em Ação**. Novatec, 2013.

FRITZING. fritzing. 2018. Disponível em <[www.fritzing.org](http://www.fritzing.org)>. Acesso em 18 de maio 2018.

GOMES, Cristiane Grava et. al. **A robótica como facilitadora do processo ensino aprendizagem de matemática no ensino fundamental**. São Paulo, cultura Acadêmica, 2010.

JUNIOR, J. C. R. P.; RAPKIEWICZ, C. E. **O processo de ensino-aprendizagem de fundamentos de programação: Uma visão crítica da pesquisa no Brasil**. Rio de Janeiro, In: Anais do XII Workshop sobre Educação em Computação (SBC), 2004.

LEMOS, M. A., BARROS, L. N., LOPES, R. D. **Um Biblioteca Cognitiva para o Aprendizado de Programação**. XI Workshop de Educação em Computação – WEI 2003. Campinas, SP, Brasil,

MCROBERTS, Michael. **Arduíno Básico**. São Paulo, 2011.

MOREIRA, R. A.; SILVA, C. R.; FERREIRA, L. F. G. **A utilização da plataforma arduíno em sala de aula: Um estudo de caso**. Minas Gerais, 9ª Jornada científica e tecnológica e 6º simpósio da pós-graduação do ifsuldeminas, 2017.

SOUZA, Fabio. **Placas arduíno – história até o arduíno UNO**, 21/11/2013. Disponível em:< <https://www.embarcados.com.br/placas-arduino/>> Acesso em: 25/05/2018.

VERGARA. **Projeto e relatórios de pesquisa em administração**. 14. Ed. São Paulo: Atlas, 2013.

WING, J.M. **Computational Thinking**. Communications of the ACM, Vol. 49, No.3. March 2006.

WING, J.M. **Computational thinking and thinking about computing**. Philosophical Transactions of the Royal Society. Published online 31 July 2008.

## **APÊNDICE A – QUESTIONÁRIO DE VERIFICAÇÃO DO ENSINO DE PROGRAMAÇÃO**

UNIVERSIDADE DO ESTADO DA BAHIA – UNEB – CAMPUS II  
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA  
COLEGIADO DE SISTEMAS DE INFORMAÇÃO

O objetivo da aplicação desse questionário é traçar um perfil do aluno e verificar como está sendo eficiente o ensino da programação no curso de Sistemas de Informação na Universidade do Estado da Bahia – Campus II.

1 – Informe seu sexo:

Masculino             Feminino

2 – Informe sua idade:

Entre 16 – 20 anos       Entre 21 – 25 anos       Maior que 25 anos

3 – Informe seu semestre:

2º Semestre

4º Semestre

6º Semestre

8º Semestre

Outro (Desemestralizado)

3 – Como você considera o ensino de programação no curso de sistemas de informação?  Muito Fraco  Fraco  Médio  Bom  Muito Bom

4 – Você tem alguma dificuldade para assimilar os conceitos computacionais?  Sim  Não

5 – Você acha que as metodologias aplicadas pelos professores estimulam seu estudo?  Sim  Não

6 – Qual a frequência aproximada você tem aulas práticas?  20%  40%  50%  60%  80%

7 – Como você considera seu nível de programação?

Muito Fraco  Fraco  Médio  Bom  Muito Bom

8 – Você acha que a quantidade de aulas práticas é adequada para as disciplinas?  Sim  Não

9 – Você acha que a quantidade de disciplinas de programação é adequada para o curso?  Poucas Disciplinas  Quantidade de disciplinas adequada  Muitas disciplinas

10 – Quando você terminar uma disciplina de programação, qual sua sensação em relação ao aprendizado?

Muito Fraco  Fraco  Médio  Bom  Muito Bom

11 – De que forma você acha que pode melhorar o ensino e a aprendizagem de programação? Marque somente a que você considera mais importante.

Mais aulas práticas

Livros com metodologias diferentes

Novas formas de abordagem do assunto pelo professor

- Uma metodologia de ensino diferente  
 Outro. Qual ?
- 

## **APÊNDICE B – QUESTIONÁRIO DE VERIFICAÇÃO DA METODOLOGIA EMPREGADA**

UNIVERSIDADE DO ESTADO DA BAHIA – UNEB – CAMPUS II  
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA  
COLEGIADO DE SISTEMAS DE INFORMAÇÃO

O objetivo desse questionário é verificar uma possível nova metodologia baseada no uso de sistemas embarcados no ensino de programação no curso de Sistemas de Informação na Universidade do Estado da Bahia – Campus II.

- 1 – O que você achou da aula utilizando a placa arduino  
 Muito Fraco  Fraco  Médio  Bom  Muito Bom

2 – Você acha que a aula utilizando sistemas embarcados podem melhorar os conceitos computacionais?

Sim       Não

3 – Quais estruturas foram trabalhadas na aula prática? Marque um **X** em todas abordadas na aula.

- Criação de variáveis
- Programação Sequencial
- Estrutura de decisão
- Estrutura de repetição
- Vetores
- Matrizes
- Funções

4 – Na sua opinião a utilização do arduíno em sala de aula:

- Não contribuiu para a assimilação dos conteúdos propostos na aula prática.
- Pouco contribuíram para assimilação dos conteúdos propostos na aula prática.
- Contribuíram de forma mediana para assimilação dos conteúdos propostos na aula prática.
- Contribuíram muito para assimilação dos conteúdos propostos na aula prática.

5 – Você acredita que ensinar programação através de resolução de problemas pode melhorar a compreensão dos sistemas computacionais?

Sim       Não