

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA CATARINENSE - CÂMPUS CAMBORIÚ CURSO DE SISTEMAS DE INFORMAÇÃO (BACHARELADO)

LARISSA ROZZA PELUSO

USO DA ROBÓTICA PARA O ESTUDO DE ALGORITMOS

CAMBORIÚ (SC) 2014

LARISSA ROZZA PELUSO

USO DA ROBÓTICA PARA O ESTUDO DE ALGORITMOS

Trabalho de Conclusão de Curso submetido ao Instituto Federal Catarinense — Câmpus Camboriú, para obtenção dos créditos de disciplina com nome equivalente no curso de Sistemas de Informação - Bacharelado.

Orientação: Profa. Cátia dos Reis Machado, Dra.

CAMBORIÚ (SC) 2014

LARISSA ROZZA PELUSO

USO DA ROBÓTICA PARA O ESTUDO DE ALGORITMOS

ESTE RELATÓRIO, DO TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO PARA OBTENÇÃO DOS CRÉDITOS DA DISCIPLINA DE TRABALHO DE CONCLUSÃO DE CURSO, DO 8°. SEMESTRE, OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM SISTEMAS DE INFORMAÇÃO

Camboriú (SC), 02 de Dezembro de 2014 (data da defesa)

Profa. Cátia dos Reis Machado, Dra. **Orientador**

BANCA EXAMINADORA:

Profa. Ana Elisa Ferreira Schmidt, Dra. **IFC-CAMBORIÚ**

Prof. Wagner Faria de Souza, Esp. **IFC-CAMBORIÚ**

Prof. Rogério Gonçalves Bittencourt, M.Sc.

Professor de TCC

Prof. Angelo Augusto Frozza, M.Sc. Coordenador de Curso

Dedico este Trabalho à minha família, por não me deixar desistir dos meus objetivos, e por me apoiar nos momentos de dificuldade, mesmo quando estes tornaram-se grandes obstáculos.

Agradeço, primeiramente, a Deus, por me enfrentar dar forças para todos obstáculos e prosseguir. A todos os meus professores, rico pelo conhecimento principalmente transmitido, minha a orientadora, Profa. Dra. Cátia dos Reis Machado, que me acompanhou desde o início deste trabalho, me incentivando e auxiliando nas pesquisas realizadas. Ao Instituto Federal Catarinense - Câmpus Camboriú, pelo apoio financeiro ao projeto pesquisa "Desenvolvimento de do raciocínio lógico pela inserção da robótica", no qual fiz parte, e que deu origem a este Trabalho de Conclusão de Curso. Aos meus colegas e amigos de curso, pelos anos de companheirismo, amizade e união. E a minha família, por me apoiar e permanecer ao meu lado nos momentos mais difíceis. Muito obrigada a todos!

Para se ter sucesso, é necessário amar de verdade o que se faz. Caso contrário, levando em conta apenas o lado racional, você simplesmente desiste. É o que acontece com a maioria das pessoas. (Steve Jobs)

LISTA DE ILUSTRAÇÕES

FIGURA 2 - Modelos de placas Arduino	FIGURA 1 – IDE de programação Arduino	24
FIGURA 3 - Ambiente de programação do Scratch		
FIGURA 4 - Ambiente de programação do S4A		
FIGURA 5 - Software Alice 3D		
FIGURA 6 - Software Fritzing		
FIGURA 8 – Matriz	FIGURA 6 - Software Fritzing	30
FIGURA 9 - Estrutura do método	FIGURA 7 - Selecionando a propriedade cor do LED no software Fritzing	31
FIGURA 10 - Exemplo de projeto utilizando estrutura de decisão	FIGURA 8 – Matriz	36
FIGURA 11 - Opções Verify e Upload no software Arduino	FIGURA 9 - Estrutura do método	37
FIGURA 12 - Montagem do projeto Pressionamento de botão para acender LED4 FIGURA 13 - Dados sobre LEDs	FIGURA 10 - Exemplo de projeto utilizando estrutura de decisão	42
FIGURA 12 - Montagem do projeto Pressionamento de botão para acender LED4 FIGURA 13 - Dados sobre LEDs	FIGURA 11 - Opções Verify e Upload no software Arduino	44
FIGURA 13 - Dados sobre LEDs		
FIGURA 15 - Montagem do projeto Pisca LED com 10 repetições		
FIGURA 15 - Montagem do projeto Pisca LED com 10 repetições	FIGURA 14 - Exemplo de projeto utilizando estrutura de repetição	46
FIGURA 17 - Montagem do projeto Reprodução de som com notas musicais		
FIGURA 18 - Exemplo de projeto utilizando matrizes	FIGURA 16 - Exemplo de projeto utilizando vetores	48
FIGURA 18 - Exemplo de projeto utilizando matrizes	FIGURA 17 - Montagem do projeto Reprodução de som com notas musicais	49
FIGURA 19 - Montagem do projeto Ligação de LEDs na matriz RGB		
QUADRO 1 - Pesquisas2		
•		
TABELA 1 - Dados das notas musicais50	QUADRO 1 - Pesquisas	21
	TABELA 1 - Dados das notas musicais	50

LISTA DE ABREVIATURAS E SIGLAS

E-TIC - Encontro de Tecnologia e Informação

GPS - Global Positioning System IDC - International Data Corporation

IDE - Integrated Development EnvironmentIFSC - Instituto Federal de Santa Catarina

LCD - Liquid Crystal Display LED - Light Emitting Diode

MIT - Massachussetts Institute of Technology

SO - Sistema OperacionalTI - Tecnologia da Informação

TIC - Tecnologia da Informação e Comunicação

UPF - Universidade de Passo Fundo

USB - Universal Serial Bus

V - Volts

RESUMO

O presente trabalho propõe a elaboração de um método de ensino voltado para o estudo da programação de computadores, denominado MAAlg – Método Aplicado ao Ensino de Algoritmos, que constitui-se de conceito, problema, solução prática e avaliação, e que utiliza a aplicação de projetos com Arduino para demonstração dos conceitos básicos da disciplina de Algoritmos e Programação de Computadores, sendo estes: estruturas de decisão, estruturas de repetição, vetores e matrizes. Durante o desenvolvimento da pesquisa, e visando obter a percepção dos alunos quanto à utilização da robótica no ambiente escolar, foi realizado a aplicação de questionário nas oficinas de Introdução a Arduino, ministradas no 5º e-TIC (Encontro de Tecnologia e Informação) do Instituto Federal Catarinense - Câmpus Camboriú, em que a maioria foi favorável, no qual, esta foi apenas uma forma preliminar de avaliar a aceitação dos alunos frente ao uso da robótica. O leitor encontrará no decorrer deste trabalho, alguns exemplos de projetos com Arduino, que devido a sua simplicidade poderão ser aproveitados tanto por professores quanto por alunos, para entender cada um dos conceitos mencionados e aprender o estudo da lógica e da programação de computadores de uma forma mais atrativa.

Palavras-chave:

Arduino; Processo ensino-aprendizagem; Algoritmos e Programação de Computadores; Método Aplicado.

ABSTRACT

This paper proposes the development of a teaching method focused on the study of computer programming, called MAAlg – Applied Method Algorithms to teaching, which consists of concept, problem, solution and evaluation practice, and using the application Arduino projects for demonstration of the basic concepts of the discipline of Algorithms and Computer Programming, namely: decision-making structures, repetition structures, vectors and matrices. During the development of research, and to obtain the students' perception regarding the use of robotics in the school environment, was carried out questionnaire in the Introduction to Arduino workshops, held on the 5° m-TIC (Meeting of Technology of Information) Instituto Federal Catarinense – Câmpus Camboriú, where the majority was favorable, in which this was only a preliminary way to evaluate the acceptance of students towards the use of robotics. The reader will find in this paper, some examples of projects with Arduino, which due to its simplicity can be availed by both teachers and pupils to understand each of the mentioned concepts and learn the study of logic and computer programming of a more attractive way.

Keywords:

Arduino; Teaching-learning process; Algorithms and Computer Programming; Applied method.

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Apresentação	
1.2 Descrição do problema	
1.3 Justificativa	
1.4 Objetivo geral	
1.5 Objetivos específicos	
1.6 Metodologia	
2 FUNDAMENTAÇÃO TEÓRICA	
2.1 Problemática do processo ensino-aprendizagem de algoritmos e programa	
computadores	
2.2 Estudos relacionados	
2.2.1 Pesquisas realizadas	
2.2.2 Iniciativas de projetos com robótica educativa	22
2.3 Ferramentas	24
2.3.1 Arduino	24
2.3.2 Scratch	28
2.3.3 Alice 3D	29
2.3.4 Fritzing	30
2.4 Conceitos trabalhados	31
2.4.1 Estruturas de decisão	32
2.4.2 Estruturas de repetição (Loopings ou Loops)	
2.4.3 Vetores	
Representação de vetor (em português estruturado):	
var numero: vetor [15] de inteiro	
2.4.4 Matrizes	
Representação de matriz (em português estruturado):	
var matriz: vetor [130, 14] de inteiro	36
3 MÉTODO PROPOSTO	37
3.5.1 Estruturas de decisão	
3.5.2 Estruturas de repetição	46
3.5.3 Vetores	
3.5.4 <i>Matrizes</i>	
3.6.1 Apresentação dos resultados	53
4 CONSIDERAÇÕES FINAIS	57

REFERÊNCIAS BIBLIOGRÁFICAS	59
APÊNDICES6	i 3
ANEXOS6	59

1 INTRODUÇÃO

1.1 Apresentação

A complexidade envolvida na área de Ciências Exatas e da Terra e nos cursos de TI em geral, incluindo as dificuldades de abstração de problemas lógicos e de adaptação às disciplinas de programação de computadores, tem dificultado a permanência de alunos nos cursos relacionados à informática.

Segundo Ribeiro, Martins e Bernardini (2011):

A disciplina de programação é considerada a base para o curso de Ciência da Computação, estando presente também no plano curricular de outros cursos, como os de Engenharia, por exemplo. Esta disciplina aborda os princípios de lógica de programação, a fim de desenvolver a capacidade de raciocínio dos alunos, permitindo que estes consigam analisar e resolver problemas por meio da sua representação na forma de algoritmos, e costuma apresentar altos índices de evasão e reprovação, sendo um dos gargalos existentes nos cursos de graduação na área de computação e engenharias.

Ainda, de acordo com Deters *et. al.* (2008): "Tais disciplinas são consideradas desafiadoras pelos alunos, pois exigem o desenvolvimento de estratégias de solução de problemas com base lógico-matemática", tendo como consequência disso o elevado número de problemas de aprendizagem, que favorece a ocorrência de reprovações e desistências.

Frente a problemática que envolve o ensino-aprendizagem da programação de computadores, tornam-se necessários a existência de novos métodos de ensino que possam atrair mais os alunos para essa área. A robótica é uma grande aliada nesse processo.

Para Werner e Gaspar Jr. (2001), "Os equipamentos de robótica permitem criar ambientes de aprendizagem atrativos onde os alunos interagem com materiais

concretos e computadores".

A robótica no meio educacional possibilita uma maior interação professor/aluno, "permitindo que estes experimentem um aprendizado constante por meio de busca e investigação" (RIBEIRO; MARTINS; BERNARDINI, 2011).

Desta forma, esta pesquisa propõe a elaboração do método MAAlg (Método Aplicado ao Ensino de Algoritmos), baseado em conceito, problema, solução prática e avaliação, que consiste na aplicação de projetos com Arduino para o ensino da lógica e da programação de computadores, sendo o trabalho estruturado em 4 capítulos, com seções e subseções, no qual:

- o capítulo 1 apresenta brevemente os problemas que abrangem as disciplinas de programação de computadores e que desmotivam os alunos na permanência em cursos de TI, descrevendo o problema da temática, a justificativa do trabalho, seus objetivos (geral e específicos) e a metodologia utilizada para o desenvolvimento da pesquisa;
- o capítulo 2 explica de modo mais aprofundado as questões que envolvem os problemas do processo ensino-aprendizagem da disciplina de Algoritmos e Programação de Computadores, mencionando algumas pesquisas realizadas, iniciativas de projetos com robótica educativa, ferramentas (Arduino, Scratch, Alice 3D e Fritzing) e os conceitos trabalhados (estruturas de decisão, estruturas de repetição, vetores e matrizes);
- o capítulo 3 inicia a explicação da estrutura do método, demonstrando sua aplicação prática por meio de exemplos de projetos com Arduino, para cada um dos conceitos trabalhados, além de incluir as descrições do local de aplicação das oficinas de Introdução a Arduino e do questionário de percepção dos alunos quanto ao estudo da programação de computadores com o uso da robótica (e-TIC, escola de rede pública federal e resultados obtidos do questionário aplicado);
- o capítulo 4 apresenta as conclusões.

Todos os sketches dos projetos apresentados poderão ser encontrados ao final deste trabalho (Apêndices A, B, C e D), incluindo os anexos de alguns exemplos relevantes.

1.2 Descrição do problema

O estudo de Algoritmos nos cursos de graduação em informática, e cursos de tecnologia em geral, é um dos grandes problemas enfrentados pelos estudantes devido à dificuldade em entender questões de lógica e traduzi-las em uma linguagem computacional.

Por ser a base dos cursos de informática, e os alunos considerarem um obstáculo difícil de ser superado, os mesmos acabam desistindo de seus objetivos acadêmicos, sendo que a evasão gerada reflete na falta de profissionais qualificados na área de TI. Os professores, por sua vez, deparam-se com o desafio de tornar o aprendizado da disciplina de Algoritmos e Programação de Computadores mais interessante, estimular a criatividade e a busca por novos conhecimentos de modo mais atrativo ao aluno.

1.3 Justificativa

Diversos problemas que envolvem o processo ensino-aprendizagem da disciplina de Algoritmos e Programação de Computadores, como elaborar procedimentos lógicos e organizar a sequência de ideias corretamente, tem intensificado o número de desistências dos estudantes em cursos de Informática. Segundo Neto (2007):

Os cursos da área de computação e informática enfrentam um grande problema com as disciplinas de introdução à programação de computadores, as quais visam ensinar como utilizar o computador para solucionar problemas. Acadêmicos iniciantes, ao se depararem com a disciplina, sentem-se incapazes de programar, devido ao conjunto de habilidades que a programação exige como capacidade

para solucionar problemas, raciocínio lógico, habilidade matemática, capacidade de abstração, entre outras.

Nesse contexto, propor novos métodos de ensino-aprendizagem, que sejam mais atrativos aos alunos, têm se tornado uma necessidade, visto que inúmeros problemas são advindos das disciplinas que envolvem programação de computadores.

Para este trabalho optamos por escolher o Arduino como plataforma de prototipagem eletrônica, por ser de baixo custo e de fácil utilização.

1.4 Objetivo geral

Criar um modelo para demonstrar a aplicação dos conceitos básicos da disciplina de Algoritmos e Programação de Computadores por meio da criação de projetos com Arduino.

1.5 Objetivos específicos

- a) Proporcionar maior atratividade ao aluno no estudo de algoritmos;
- b) Aplicar os conceitos vistos em aula em projetos práticos.

1.6 Metodologia

Para este projeto foi utilizado como metodologia à pesquisa bibliográfica, compondo a revisão de literatura, primeira fase do trabalho, que consistiu na busca de informações sobre os seguintes temas: dificuldade dos estudantes de computação na disciplina de Algoritmos e Programação de Computadores, métodos atrativos de ensino-aprendizagem e aplicações de robótica que auxiliam nesse processo.

A pesquisa foi desenvolvida em cinco etapas: levantamento bibliográfico, compra de material, proposta do método, elaboração de um conjunto de oficinas para serem utilizadas nas aulas e aplicação das oficinas.

Na etapa de levantamento bibliográfico inicialmente, realizou-se a coleta de todo material necessário para o desenvolvimento do projeto, no qual, o conteúdo sobre robótica divide-se em duas partes específicas: pesquisa sobre a plataforma Arduino e pesquisa sobre a programação em *hardware* (com microcontroladores ATMEGA 328), que inclui a leitura de artigos científicos, livros, textos da internet e visualização de vídeos sobre aplicações práticas.

Após a realização da coleta de dados e informações sobre o assunto, também foram inclusos exemplos sobre cada um dos conceitos trabalhados na disciplina de Algoritmos e Programação de Computadores, conforme o método proposto, visando a melhor fixação desses conceitos dados em aula, pelo aluno, e demonstrar a sua aplicabilidade na educação.

O método proposto constitui-se em quatro partes, sendo estas: conceito, problema, solução prática e avaliação, conforme as descrições a seguir:

<u>Conceito:</u> Assuntos abordados na disciplina de Algoritmos e Programação de Computadores, como por exemplo, estruturas de decisão, estruturas de repetição, vetores e matrizes.

<u>Problema:</u> Para cada conceito foi definido um problema a ser resolvido, que em termos práticos, consiste no que o programa deve fazer.

<u>Solução prática</u>: Elaboração dos exemplos práticos e disponibilização do código-fonte. Refere-se à aplicação final, composta pelos exemplos elaborados com o *software* e o *hardware* Arduino, de acordo com o problema a ser resolvido.

Avaliação: Consiste na aplicação do método com algumas turmas de alunos, em forma de provas, exercícios de fixação ou trabalhos, de modo a avaliar o aprendizado do aluno em disciplinas que envolvem algoritmos, raciocínio lógico e programação de computadores.

A compra do material necessário obteve-se com recurso aprovado no projeto de Pesquisa e Extensão Edital 195/2013 PROEX/PROPI/IFC, além do que foi adquirido particularmente. Esses materiais foram utilizados para elaboração dos

exemplos práticos apresentados em cada uma das oficinas.

As etapas de elaboração e aplicação das oficinas foram organizadas seguindo o modelo proposto, selecionando exemplos de projetos para cada um dos conceitos, definindo os problemas e demonstrando cada um deles com o uso do *software* e do *hardware* Arduino.

Por envolver a resolução de problemas concretos de modo prático, esta pesquisa caracteriza-se como sendo de caráter aplicado.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Problemática do processo ensino-aprendizagem de algoritmos e programação de computadores

O processo de ensino-aprendizagem da disciplina de Algoritmos e Programação de Computadores tem sido um desafio enfrentado tanto por alunos quanto por professores. Isto ocorre devido a inúmeros fatores que envolvem a capacidade de transmissão do conhecimento aos estudantes, pelo docente, de modo claro e de fácil compreensão, e também, as dificuldades apresentadas pelos alunos, que muitas vezes "culminam em desmotivação, reprovação ou evasão" (HINTERHOLZ JR., 2009). Ainda segundo o mesmo autor, podem ser citados diversos problemas, como por exemplo: dificuldade de adaptação no desenvolvimento do raciocínio lógico; falta de motivação e desânimo pela disciplina, por acreditarem ser um obstáculo difícil de ser superado.

De acordo com Piva Jr. e Freitas (2011):

Por ser a disciplina de algoritmos o primeiro contato do aluno ingressante com o mundo da computação, em termos acadêmicos, caso exista algum problema de entendimento ou absorção de tal conteúdo, na maioria dos casos ocorre uma desmotivação por parte dos estudantes, fazendo-os pensar em desistir ou mudar de curso. O que se observa, de forma geral, é que o aluno ingressante em cursos superiores da área de computação e informática e que não possui ou não consegue desenvolver a competência de abstração de problemas, não consegue um bom desempenho em disciplinas iniciais do curso, tais como algoritmos, cálculo, álgebra entre outras.

Essa dificuldade enfrentada pelos alunos faz com que tenhamos uma carência de profissionais qualificados frente a um mercado em expansão na área de TI no Brasil. Segundo os dados realizados em estudo da consultoria independente IDC

("Habilidades em Redes e Conectividade na América Latina" - *Networking Skills Latin America*) encomendado pela Cisco na América Latina, a demanda por profissionais de Tecnologia da Informação e Comunicação (TIC) no Brasil excederá a oferta em 32% para o ano de 2015, chegando a uma lacuna de 117.200 trabalhadores especializados em redes e conectividade (CISCO, 2013).

Observa-se que várias pesquisas são realizadas no sentido de contribuir com essa problemática, entre estas, o ensino da robótica, que nos últimos anos tem se difundido. Barbosa (2008) realizou uma experiência didática aplicando a robótica com o objetivo de que cada estudante construísse seu próprio robô para realizar uma competição entre os alunos da mesma turma. Ao final da experiência, foi verificado que houve um maior interesse por parte dos alunos em realizar trabalhos de conclusão de curso em áreas relacionadas à robótica, controle e automação. Silva *et al.* (2009) realizaram uma pesquisa de campo a fim de verificar como ocorre a interdisciplinaridade mediante a robótica, se realmente ela auxilia no entendimento das outras disciplinas e como é feito este trabalho em relação à metodologia utilizada pelo professor. Romão e Sacchelli (2009) realizaram uma pesquisa aplicando os conceitos da Física com o auxílio da robótica educacional. Ribeiro *et al.* (2011) apresentaram os resultados de um estudo da utilização da robótica em âmbito educacional superior como uma ferramenta de apoio ao ensino de disciplinas de programação.

E ainda, iniciativas observadas recentemente em projetos estão sendo desenvolvidas, com o intuito de inserir a robótica para crianças no meio educacional. Dentre essas iniciativas estão: Escola SuperGeeks, robótica e programação voltada para crianças, Vila Mariana (São Paulo), que incentiva crianças a partir dos sete anos de idade a se tornarem criadores de tecnologia (desenvolvimento de programas, jogos eletrônicos e pequenos robôs) e não apenas consumidores (SUPERGEEKS, 2014). Prefeitura de Itajaí, com a realização de uma oficina de robótica para alunos de 12 e 14 anos da Escola Básica Francisco Celso Mafra, através de uma parceria entre o Instituto Federal de Santa Catarina e a unidade escolar, no qual um professor do IFSC tem a ajuda de um estagiário para mostrar a prática da montagem de robôs (ITAJAI, 2014).

2.2 Estudos relacionados

Nesta seção, serão apresentados os estudos relacionados à temática da inclusão da robótica no ambiente educacional, como auxílio ao processo ensino-aprendizagem da programação de computadores.

2.2.1 Pesquisas realizadas

QUADRO 1 - Pesquisas

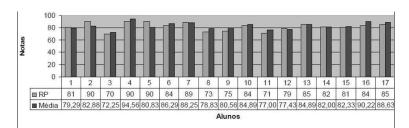
carro-robô Consiste na construção de um carro-robô, da placa controladora e do Utilização construído com componentes de baixosoftware, para utilização no ensino da programação orientada a custo como ferramenta de apoio ao objetos, sendo, os conceitos de orientação a objetos trabalhados no dehardware, colocando os alunos frente a novos desafios, e levando-os a aprendizado conceitos dos pesquisar equações da Física, comunicação em baixo nível orientação a objetos (hexadecimal) e a utilizar seus próprios conhecimentos de programação orientada a objetos (WERNER; GASPAR, 2001).



Uso do carro-robô em sala de aula (FONTE: WERNER; GASPAR, 2001)

programação de computadores

Utilização da robótica pedagógica no Apresenta uma investigação da utilização da robótica pedagógica no processo de ensino-aprendizagem de processo ensino-aprendizagem da programação de computadores, realizando uma pesquisa através de um estudo de caso em uma instituição de ensino superior de Belo Horizonte, no qual, o curso de Sistemas de Informação é ministrado com foco em uma disciplina que aplica a robótica pedagógica como instrumento de apoio ao processo ensino-aprendizagem da programação de computadores. Adota uma abordagem qualitativa que emprega como instrumentos de coleta de dados uma extensa observação de campo, entrevistas semiestruturadas e pesquisa documental. Professores aplicam a montagem de robôs em sala de aula, desafiando os alunos a construírem uma esteira rolante, inicialmente, sem utilizar a programação, com a ideia de que os alunos aprendessem os conceitos de robótica e mecânica, e que soubessem a importância da programação de computadores para tornar o projeto mais interessante. Para esta atividade foram utilizados os kits MINDSTORMS LEGO³⁸ (ROCHA, 2006).



Comparativo entre a nota de robótica pedagógica e a média das notas na disciplina de programação (FONTE: ROCHA, 2006)

programação

O desafio de trabalhar com alunos Aborda a problemática envolvida no processo de ensino**repetentes na disciplina de algoritmos e**aprendizagem da disciplina de Algoritmos e Programação de Computadores, desde os motivos que contribuem para a reprovação do aluno e possível desistência da disciplina ou do curso, até os motivos pelos quais tornam as aulas mais cansativas aos professores, devido à complexidade em transmitir o conhecimento aos estudantes da melhor forma possível para que desenvolvam o raciocínio lógico e a prática da programação (DETERS et. al., 2008).

Nome da Disciplina	Curso	Índice de repetência	Recursos utilizados e temas ensinados	Estratégias de Ensino e Avaliação
[1] Algoritmos e Programação (1º período)	Ciência da Computação	56,41%	Meta-linguagem Portugol Ambiente Virtual ALICE Editor WebPortugol IDE CodeBlocks	Aulas expositivas com exercícios práticos (listas de exercícios), trabalhos em grupo e avaliações individuais
[2] Algoritmos e Programação (2º período)	Ciência da Computação	50%	Ambiente Virtual ALICE IDE CodeBlocks	Aulas expositivas com exercícios práticos (listas de exercícios), trabalhos em grupo e avaliações individuais
[3] Programação 1 (3º período)	Ciência da Computação	20%	Ambiente Virtual genérico Linguagem JAVA NetBeans	Aulas expositivas com exercícios práticos (listas de exercícios) e projeto final
[4] Lógica e Algoritmos (1º período)	Tecnologia em Sistemas para Internet	37,78%	Meta-linguagem Portugol Ambiente Virtual ALICE Editor WebPortugol	Aulas expositivas com exercícios práticos (listas de exercícios) e avaliações individuais
[5] Programação 1 (1º período)	Tecnologia em Sistemas para Internet	40%	Ambiente Virtual genérico Linguagem de Web PHP	Aulas expositivas com exercícios práticos (listas de exercícios) e avaliações individuais
[6] Algoritmos e Programação (1º período)	Curso Técnico de Programação	50%	Ambiente Virtual da IES VisualG	Aulas expositivas com exercícios práticos (listas de exercícios) e avaliações individuais

Cenários das disciplinas (FONTE: DETERS et. al., 2008) (FONTE: Própria autora, 2014)

2.2.2 Iniciativas de projetos com robótica educativa

Prefeitura de Passo Fundo: Em parceria com a prefeitura de Passo Fundo (Rio Grande do Sul), os cursos de Ciência da Computação e de Física da UPF (Universidade de Passo Fundo) promovem um evento sobre a utilização da programação de computadores e da robótica livre no ensino fundamental e médio, a 1ª Olimpíada de Robótica Educativa Livre, em 29 de abril de 2014, de modo a criar novas formas de utilizar os recursos de informática das instituições de ensino, despertar o interesse para as áreas de física e informática, proporcionar situações reais de aplicação de conceitos da Física e aproximar a Universidade das redes públicas de ensino (ONACIONAL, 2013).

- b <u>Prefeitura de Itajaí</u>: Em parceria com a prefeitura de Itajaí e o Instituto Federal de Santa Catarina (IFSC), a Escola Básica Francisco Celso Mafra recebe uma Oficina de Robótica, com cerca de 32 estudantes participantes, de idade entre 12 e 14 anos. Um professor do IFSC conta com a ajuda de um estagiário, para mostrar a montagem de um robô aos alunos da Escola na prática, sendo distribuídos oito kits de robótica entre eles (ITAJAI, 2014).
- c <u>Escola SuperGeeks</u>: Escola criada e inaugurada em Vila Mariana (São Paulo), visa incentivar crianças a partir dos sete anos de idade a serem criadores de tecnologia e não apenas consumidores. Dentre os conteúdos de ensino da escola estão o desenvolvimento de programas, jogos eletrônicos e pequenos robôs, sendo algumas das ferramentas utilizadas: Scratch, Arduino, Blender, Alice 3D e HTML + CSS + JS (SUPERGEEKS, 2014).

Segundo os estudos apresentados é possível observar a busca por novas formas de ensinar lógica nos diferentes estágios de aprendizagem, desde crianças, adolescentes e adultos, sendo a robótica uma ferramenta muito utilizada nesse processo. Dentre as quais, pôde-se perceber muitos resultados positivos com sua aplicabilidade no ambiente educacional. Essas informações fortaleceram a ideia de trabalhar a robótica no ensino da lógica e da programação de computadores.

2.3 Ferramentas

2.3.1 Arduino

a O que é Arduino?

Arduino é uma plataforma de prototipagem eletrônica *open-source* (código aberto), que pode ser utilizado para programar o processamento de entradas e saídas entre um dispositivo e os componentes externos conectados a ele. É conhecido, também, por plataforma de computação física ou embarcada, isto é, um sistema que interage com seu ambiente por meio de *hardware* e *software* (MCROBERTS, 2011), podendo ser considerado os dois, com as placas e a IDE de programação (FIGURA 1), utilizadas em conjunto.

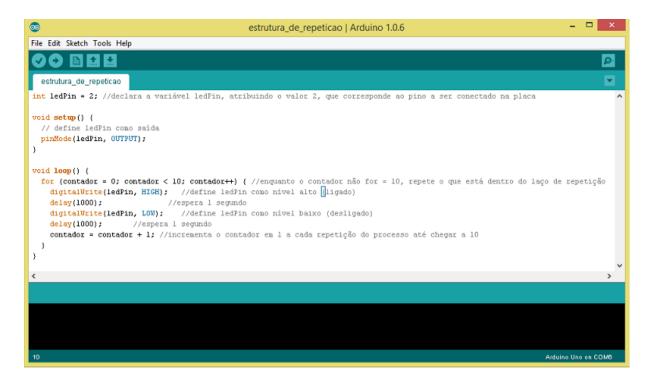


FIGURA 1 – IDE de programação Arduino (FONTE: Própria autora, 2014)

A família Arduino (FIGURA 2) possui uma ampla gama de placas existentes, e vem crescendo a cada dia, pois, por ser um *hardware* livre, permite-se que sejam criadas as denominadas placas-clone com base no Arduino, e conforme a necessidade de cada aplicação. Porém, a palavra "Arduino" é reservada, não sendo permitido utilizá-la para as novas placas desenvolvidas. Daí a existência dos nomes das placas-clone, como Freeduino, Roboduino. O Arduino é também um *software* livre, no qual, o código, os esquemas, o projeto, disponibilizados podem ser aproveitados livremente por qualquer pessoa e com qualquer propósito (MCROBERTS, 2011). Para se ter acesso a funcionalidades adicionais, é possível estender o Arduino utilizando *shields*, que são placas de circuito acessórias, conectando-o a dispositivos como *displays* de LCD, receptores GPS, módulos de Ethernet, entre outros. Ainda, cada programa desenvolvido no *software* Arduino é conhecido por *sketch*. É desta forma que os programas feitos na plataforma são chamados.



FIGURA 2 - Modelos de placas Arduino (FONTE: http://ardufc.blogspot.com.br/2012/10/modelos-do-arduino.html)

b Instalação do software Arduino

Para dar início a criação dos projetos, é necessário fazer o *download* da IDE de programação no site oficial do Arduino (www.arduino.cc), na aba "*Download*", e

instalá-la, juntamente com os *drivers* de USB, dependendo da plataforma utilizada. Nesse site, também constam instruções completas de instalação do *software* Arduino em computadores com os sistemas operacionais Windows, Linux e Mac (MONK, 2013).

Para fazer os projetos foi utilizado um computador com o SO Windows, porém, o *software* Arduino também é compatível com Linux e Mac. Maiores informações sobre a instalação nesses dois últimos sistemas operacionais poderão ser encontradas em (MONK, 2013), no livro "Programação com Arduino – começando com *Sketches*", que foi uma das referências utilizadas para este trabalho.

Para instalar o Arduino no Windows precisam ser baixados dois arquivos: o instalador "Windows Installer", e o arquivo compactado "Windows ZIP File", que contém os drivers de instalação das placas (ANEXO B).

Ao baixar os dois arquivos, deve-se realizar a instalação do Arduino, executando o arquivo com a extensão ".exe" (ANEXO C), e também, descompactar a pasta que contém os *drivers*, utilizando algum descompactador de arquivos (WinRAR, WinZIP...).

Ao descompactar a pasta do Arduino, percebe-se que há outras pastas dentro dela. É a pasta "drivers", que possui o instalador do pacote de drivers de todas as placas Arduino. Para iniciar a instalação, precisa ser executado o arquivo com o nome "dpinst", conforme a versão do computador utilizado, 32 bits ou 64 bits (ANEXO D).

Concluindo os procedimentos de instalação, é possível inicializar o *software* Arduino para dar início aos projetos. Mas, antes disso, algumas configurações são requeridas.

c Configuração

Após ter instalado o *software* Arduino, é preciso fazer algumas configurações básicas nele, conforme a placa a ser utilizada. Para isso, inicialmente, deve-se conectar a placa ao computador através do cabo USB (do tipo A para B) e inicializar a IDE de programação, verificando se no item "*Serial Port*" do menu "*Tools*" é possível

selecionar a porta utilizada. Essa verificação da porta poderá ser feita, também, seguindo os passos a seguir:

No Windows 8: Campo de pesquisa da tela Iniciar → Digitar a palavra "Gerenciador de Dispositivos" e selecioná-la → Portas - COM e LPT

No Windows 7: Menu Iniciar → Computador → Propriedades do Sistema → Gerenciador de Dispositivos → Portas - COM e LPT

Ao abrir o menu "Iniciar" e selecionar "Computador" (ANEXO F), será aberto a tela que contém informações sobre as Unidades de Disco Rígido e os Dispositivos com Armazenamento Removível (Unidade de DVD).

Na parte superior da tela deverá ser selecionada a opção "Propriedades do sistema" (ANEXO G), que consta o "Gerenciador de Dispositivos", opção exata a mostrar as portas do Arduino.

Ao clicar em "Gerenciador de Dispositivos" (ANEXO H), será aberta uma nova tela com as informações sobre o computador e seus periféricos, incluindo a opção "Portas - COM e LPT", que precisa ser selecionada. Lembrando que só é possível selecioná-la se a placa estiver conectada ao computador.

Com a seleção da opção "Portas - COM e LPT" (ANEXO I), ao abri-la, e se tudo ocorrer corretamente aparecerá à palavra "Arduino Uno - COM4". A porta COM pode variar dependendo das configurações do computador, então, não há problemas quanto ao número que a acompanha.

No entanto, a porta que consta na opção "Portas - COM e LPT" deverá ser a mesma a ser selecionada em "Serial Port" no menu "Tools" da IDE de programação Arduino (ANEXO J).

Concluindo todos esses procedimentos, deve-se selecionar a placa que irá ser utilizada para os projetos (ANEXO K). Nesse caso, será utilizado o Arduino Uno. Para isso, a placa precisa ser selecionada em "*Board*" do menu "*Tools*".

Com todos os procedimentos de instalação e configuração do Arduino

realizados, pode-se iniciar os projetos práticos, mas antes, serão explicadas brevemente algumas outras ferramentas que podem ser utilizadas no ensino da lógica de programação de computadores, e alguns conceitos a serem trabalhados no método proposto.

2.3.2 Scratch

Scratch é um *software* educativo voltado para o ensino da programação de computadores. Possui uma linguagem de fácil entendimento, que utiliza o conceito de programação em blocos, no qual são combinados, a lógica com a percepção visual, e através do encaixe de diversos blocos é possível criar várias funções. Foi desenvolvido no *Massachussetts Institute of Technology* (MIT) *Media Lab*, com o objetivo de facilitar o aprendizado e torná-lo mais divertido (MARJI, 2014, p. 22).

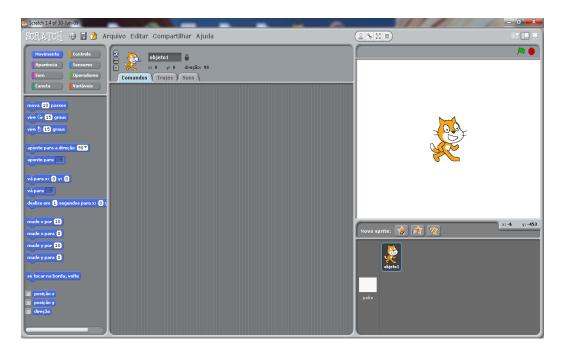


FIGURA 3 - Ambiente de programação do Scratch (FONTE: Própria autora, 2014)

Além do Scratch tradicional, há outra versão deste *software* adaptado para o Arduino, denominado de S4A (*Scratch for Arduino*). Possui a mesma estrutura do programa, porém, nos blocos de comandos são apresentados os comandos utilizados

na programação com microcontroladores, da IDE Arduino (FIGURA 4). Esta é uma ótima ferramenta para quem está iniciando projetos com robótica.

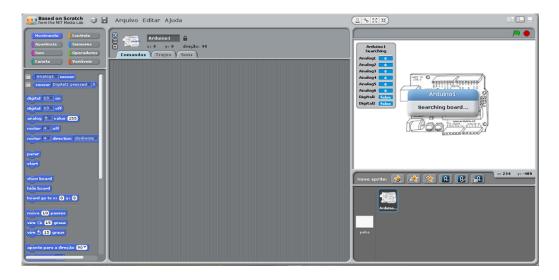


FIGURA 4 - Ambiente de programação do S4A (FONTE: Própria autora)

2.3.3 *Alice 3D*

Alice é um software livre que introduz o aluno a programação de computadores, permitindo criar animações e pequenos jogos através de um ambiente gráfico interativo (FIGURA 5), com o clique e arrasto das instruções para criação de um programa, sendo estas semelhantes às de linguagens de programação como Java e C++. Permite também, que após a adição de uma instrução, a mesma seja executada imediatamente, possibilitando que se observe o efeito sobre o objeto do cenário. Assim, o aluno vai adquirindo conhecimentos sobre programação sem se preocupar com detalhes (UFMS, 2013).

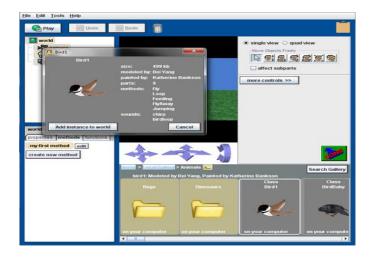


FIGURA 5 - *Software* Alice 3D (FONTE: UFMS, 2013)

2.3.4 Fritzing

Fritzing é um programa em ambiente gráfico utilizado para criação de desenhos dos projetos eletrônicos com Arduino. Com ele é possível visualizar de modo virtual como seria uma montagem física em protoboard, bem como seu circuito elétrico (MULTILÓGICA SHOP).

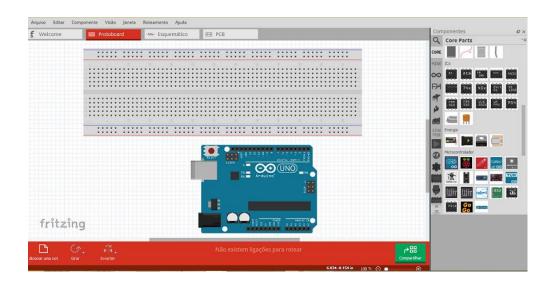


FIGURA 6 - *Software* Fritzing (FONTE: Própria autora)

O *software* Fritzing também permite personalizar os tipos de componentes a serem utilizados para o projeto, definindo propriedades como cor do LED, rotação de um objeto, tipo de Arduino, valor do resistor, dentre outras (FIGURA 7).

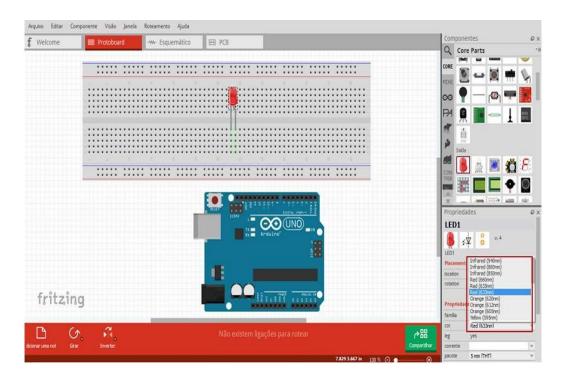


FIGURA 7 - Selecionando a propriedade cor do LED no software Fritzing

2.4 Conceitos trabalhados

No desenvolvimento de algoritmos utilizam-se diversos conceitos, como dados de entrada, processamento lógico e saída, variáveis, constantes, atribuições, expressões lógicas, relacionais e aritméticas, bem como da tradução desses conceitos por comandos de modo a representar um conjunto de ações, sendo que para sua viabilidade, é necessário existir uma perfeita relação lógica intrínseca pelo qual essas ações são executadas, à maneira em que é regido o fluxo de execução do algoritmo (FORBELLONE; EBERSPÄCHER, 2005, p. 30).

Através de estruturas básicas de controle do fluxo de execução podem ser criados algoritmos para solução de problemas. Algumas dessas estruturas são:

estruturas de decisão e estruturas de repetição. Mais adiante serão explicadas cada uma delas, incluindo os conceitos de vetores e matrizes.

É importante ressaltar que a disciplina de Algoritmos e Programação de Computadores abrange diversos conceitos, porém, este trabalho apresenta apenas alguns deles, para a demonstração da aplicação do método.

2.4.1 Estruturas de decisão

Estruturas de decisão são estruturas utilizadas por computadores para realização da tomada de decisões por meio de processamento lógico, e com base em determinada condição, no qual é estabelecida uma ação de desvio na operação do fluxo do programa, sendo as estruturas de decisão, divididas em três tipos: decisão simples, decisão composta e decisão múltipla.

a Decisão simples: define-se pela execução de uma condição (entre os comandos **se** e **então**) apenas se esta for verdadeira.

Exemplo de estrutura de decisão simples (em português estruturado):

se (<condição>) então

[instruções são executadas se a condição for verdadeira]

fim_se

b Decisão composta: define-se pela execução das instruções independentemente de o resultado ser falso ou verdadeiro, ou seja, são executadas tanto as instruções com o valor falso, quanto às com valor verdadeiro.

Exemplo de estrutura de decisão composta (em português estruturado):

se (<condição>) então

[instruções são executadas se a condição for verdadeira]

senão

[instruções são executadas se a condição for falsa]

fim_se

[instruções são executadas após a condição ser falsa ou ser verdadeira]

c Decisão múltipla: utilizada em situações onde há um grande número de verificações lógicas a serem realizadas.

Exemplo de estrutura de decisão múltipla (em português estruturado):

```
caso <variável>
seja <opção 1> faça
[ação se a condição 1 for verdadeira]
seja <opção 2> faça
[ação se a condição 2 for verdadeira]
seja <opção 3> faça
[ação se a condição 3 for verdadeira]
senão
[ação se nenhuma condição for satisfeita]
fim_caso
```

2.4.2 Estruturas de repetição (Loopings ou Loops)

Estruturas de repetição, também conhecidas por laços de repetição, são estruturas de programação utilizadas nas ocasiões em que precisam ser repetidos determinados trechos de código, reduzindo o trabalho do programador, principalmente quando é necessário repetir diversas vezes alguma ação importante do programa. Existem três tipos de estruturas de repetição, são elas: repetição com teste no início, repetição com teste no final e repetição com variável de controle.

a Repetição com teste no início: é uma estrutura que controla o fluxo de execução, permitindo repetir várias vezes um mesmo trecho do algoritmo, porém, sempre fazendo a verificação antes de cada execução se é permitido executar o mesmo trecho. Para a realização desse tipo de teste, utiliza-se a estrutura **enquanto**, que possibilita repetir um bloco ou uma ação enquanto uma determinada <condição> for verdadeira.

Exemplo de estrutura de repetição com teste no início (em português estruturado):

enquanto (<condição>) faça[instruções são executadas enquanto a condição for verdadeira]

fim_enquanto

b Repetição com teste no final: a repetição com teste no final é realizada utilizando a estrutura **repita**, que permite repetir um bloco ou ação **até** que uma determinada condição seja verdadeira.

Exemplo de estrutura de repetição com teste no final (em português estruturado):

↓ a**nit**a

repita

[instruções são executadas no período em que a condição for falsa, isto é, até que esta seja verdadeira]

até (<condição>)

c Repetição com variável de controle: para a repetição com variável de controle, utiliza-se a estrutura **para**, que sempre executa o bloco repetidamente em um número predeterminado de vezes, por não prever uma condição e possuir limites fixos.

Exemplo de estrutura de repetição com variável de controle (em português estruturado):

1

35

para (<variável>) de (<início>) até (<fim>) faça

[instruções são executadas durante o ciclo de contagem da variável de controle]

fim_para

2.4.3 Vetores

Vetores são estruturas de dados unidimensionais que armazenam um conjunto de valores em uma única variável e em determinadas posições utilizando o conceito de índices. Porém, todos esses valores precisam ser do mesmo tipo de dado (inteiro, real, caracter ou lógico). Possui apenas uma dimensão, dividindo-se em dois tipos de vetores: vetores linha e vetores coluna. No qual, os vetores linha se referem a uma linha e várias colunas, e os vetores coluna, consistem em uma coluna e várias linhas.

Exemplo de vetor linha

Exemplo de vetor coluna

$$\left[\begin{array}{c} 12345 \end{array}\right]$$

Representação de vetor (em português estruturado):

V

var numero: vetor [1..5] de inteiro

2.4.4 Matrizes

Matrizes são estruturas de dados bidimensionais, compostas por duas variáveis que representam linhas e colunas. Da mesma forma como os vetores, permitem que seja armazenado mais de um valor. Por ser uma estrutura de dados homogênea, todos os dados armazenados na matriz devem ser do mesmo tipo (LOPES; MELO, 2002).

Exemplo de matriz que armazena quatro notas de uma turma de trinta alunos

	1	2	3	4
1	7.5	8.0	7.0	9.0
2	6.0	7.0	5.5	8.0
3	9.5	9.2	10.0	8.7
			:	
		:	:	:
			:	:
30	4.0	5.8	7.0	6.5

FIGURA 8 – Matriz

(FONTE: Ricardo Sobjak, 2011)

Representação de matriz (em português estruturado):

 \downarrow

var matriz: vetor [1..30, 1..4] de inteiro

3 MÉTODO PROPOSTO

Conforme literatura apresentada verifica-se que a aplicação da robótica no ambiente escolar tem motivado muitos alunos para essa área, e nos diferentes estágios, desde crianças, a adolescentes e adultos, com algumas pesquisas indicando melhoria no ensino da programação de computadores. Sendo assim, o método proposto visa integrar a teoria com a prática de uma forma atrativa, e ao final, fazer uma avaliação no sentido de verificar o aprendizado do aluno. A seguir, a FIGURA 9 apresenta sua estrutura.

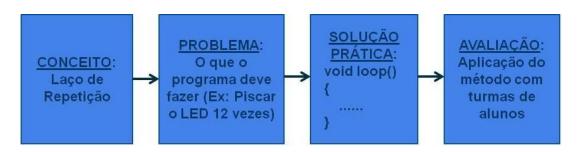


FIGURA 9 - Estrutura do método

(FONTE: Própria autora, 2014)

Esse método, denominado MAAlg (Método Aplicado ao Ensino de Algoritmos), divide-se, basicamente, em quatro partes, sendo elas: conceito, problema, solução prática e avaliação, no qual, cada parte compõe o resultado final de sua elaboração.

No decorrer deste trabalho, serão explicados, de modo mais detalhado, os procedimentos do método proposto.

38

3.1 CONCEITO

Propósito: Definir o conceito a ser trabalhado em cada aula ou oficina, sendo os

conceitos: estruturas de decisão, estruturas de repetição, vetores e matrizes.

Atores envolvidos: Professores, ministrantes de oficinas práticas e instrutores de

robótica.

Atividades de cada ator: Qualquer um dos atores envolvidos nesta etapa precisarão

organizar os dias, horários e qual o conceito a ser aplicado em cada uma das aulas.

Resultados esperados: Dentre os resultados esperados, estão: para cada conceito,

sejam propostos alguns projetos, de modo a visualizar esse conceito no hardware,

explicando, inicialmente, sua descrição e demonstrando exemplos de aplicação.

Modo de aplicação: A etapa de conceito deve ser aplicada por um, ou no máximo

dois atores, se for turmas muito grandes de alunos. Em caso de haver dois atores, as

tarefas podem ser divididas da seguinte forma: o primeiro ator explica a descrição do

conceito a ser trabalhado e demonstra algum exemplo simples relativo a esse conceito,

e o segundo ator demonstra o mesmo exemplo, porém, mais aprimorado e utilizando

mais recursos. Os projetos a serem demonstrados deverão ser combinados entre os

dois atores antes do início da aula, para evitar que ocorram problemas inesperados.

3.2 PROBLEMA

Propósito: Definir o que o programa deve fazer, com sua demonstração prática no

hardware, conforme o conceito aplicado.

39

Atores envolvidos: Professores, ministrantes de oficinas práticas e instrutores de

robótica.

Atividades de cada ator: Qualquer um dos atores envolvidos precisarão definir

alguns problemas a serem trabalhados em cada conceito, com o uso de projetos

Arduino.

Resultados esperados: Espera-se como resultado, que os problemas e a criação dos

projetos definidos para cada conceito seja simples, de modo que estes possam ser

assimilados aos conteúdos da disciplina de Algoritmos e Programação de

Computadores.

Modo de aplicação: A etapa de problema deverá ser aplicada por qualquer um dos

atores envolvidos.

3.3 SOLUÇÃO PRÁTICA

Propósito: Aplicar o conceito trabalhado em cada aula na prática, utilizando para isso,

o hardware Arduino e componentes eletrônicos.

Atores envolvidos: Professores, ministrantes de oficinas práticas, instrutores de

robótica e alunos.

Atividades de cada ator: Professores, ministrantes de oficinas práticas e instrutores

de robótica deverão definir o conceito, o problema a ser resolvido (o que o programa

deve fazer) e demonstrar alguns exemplos de projetos, propondo atividades ou

40

exercícios de fixação aos alunos, de forma a aplicar o que foi passado. Essas

atividades/exercícios serão definidas por quem dará a aula. Os alunos deverão realizar

essas tarefas solicitadas.

Resultados esperados: Os resultados esperados na etapa de solução prática são: o

aluno aplicar cada conceito explicado na aula em projetos práticos com Arduino, e

assimilar os conceitos utilizados nos projetos com os conteúdos da disciplina de

Algoritmos e Programação de Computadores.

Modo de aplicação: A etapa de solução prática deverá ser realizada pelos alunos, com

o acompanhamento de um professor, ministrante de oficina prática ou instrutor de

robótica. Nessa etapa, poderão ser utilizados como base para a criação dos projetos,

tanto por quem dará a aula quanto pelos alunos, os sketches contidos na própria IDE

Arduino. Uma sugestão que os aplicadores do método poderão utilizar seria escolher

um dos exemplos de projetos da IDE Arduino e propor modificações ou adaptações

aos alunos conforme os conceitos estudados.

3.4 AVALIAÇÃO

Propósito: Avaliar a aplicabilidade e viabilidade do método.

Atores envolvidos: Professores, ministrantes de oficinas práticas, instrutores de

robótica e alunos.

Atividades de cada ator: Para avaliar o método proposto, qualquer um dos atores

envolvidos deverá aplicá-lo com no mínimo três turmas de alunos da disciplina de

Algoritmos e Programação de Computadores, explicando os conceitos de Algoritmos e demonstrando-os em projetos práticos com Arduino.

Resultados esperados: Dentre os resultados esperados com a aplicação da avaliação, estão: os alunos conseguirem assimilar os conceitos da programação de computadores com os projetos Arduino, e ficarem motivados em aprender Algoritmos.

Modo de aplicação: A avaliação do método será realizada em forma de provas, no qual, uma delas utilizará o método proposto MAAlg, e a outra, o método de ensino tradicional, visando comparar seus resultados (positivos ou negativos). Essas provas poderão ser realizadas com turmas diferentes ou com a mesma turma, ficando a critério de quem as aplicará.

3.5 APLICAÇÃO DO MÉTODO

Esta seção apresenta alguns exemplos de projetos que podem ser utilizados para o estudo da programação de computadores. Para a realização dos testes, é necessário adquirir os materiais informados em cada projeto, fazer as montagens e conexões corretamente, conforme as instruções apresentadas, além de copiar os *sketches* que constam ao final deste trabalho para o *software* Arduino, fazendo o "*Upload*" para a placa.

A aplicação do método pode ser realizada em forma de oficinas práticas de robótica, minicursos e nas próprias aulas de Algoritmos, lógica e programação de computadores.

Se for utilizar a própria IDE Arduino para a explicação dos conceitos, é importante que os aplicadores do método expliquem, primeiramente, sobre o uso de algumas funções específicas, tais como, PinMode e dados de entrada e saída (INPUT, OUTPUT, INPUT_PULLUP), considerando que os alunos não possuem conhecimento prévio a respeito. Outras alternativas que podem ser utilizadas são, o Scratch, para o

aprendizado da lógica de programação, e o S4A (versão do Scratch adaptada para a plataforma Arduino), para o ensino dos conceitos básicos.

3.5.1 Estruturas de decisão

- a Conceito: (Veja a descrição do conceito "Estruturas de decisão" na página 32).
- b <u>Problema</u>: Pressionar um botão para acender o LED.
- c Solução prática:

Projeto: Pressionamento de botão para acender LED

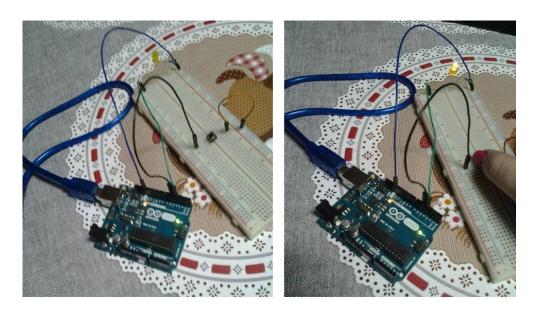


FIGURA 10 - Exemplo de projeto utilizando estrutura de decisão (FONTE: Própria autora, 2014)

Materiais necessários:

- 1 placa Arduino Uno;
- 1 LED amarelo;

- 1 resistor de 200 Ω (ohm) ou com valor maior mais próximo (resistor utilizado = 220 Ω);
- 1 protoboard;
- 4 jumpers;
- 1 cabo USB, do tipo A para B (para conectar a placa ao computador);
- 1 botão (pushbutton).

Como fazer:

Primeiramente, inicie a montagem do projeto fazendo as ligações dos componentes eletrônicos na protoboard, conforme os seguintes procedimentos:

- 1°) Encaixar o LED amarelo na protoboard;
- 2°) Conectar o resistor ao polo negativo do LED (cathodo), e à coluna negativa da protoboard, que localiza-se à parte mais externa dela;
- 3°) Conectar o jumper ao polo positivo do LED (anodo) e ao pino 4 da placa;
- 4°) Conectar o jumper à coluna negativa da protoboard e ao pino GND da placa;
- 5°) Encaixar o botão entre as duas colunas que possui uma linha divisória, ao meio da protoboard;
- 6°) Fazer a ligação da linha em que consta o botão, ao pino 5 da placa com o jumper, e no lado oposto, conectar o botão com o jumper na coluna negativa da protoboard.

Ao terminar todos os procedimentos de montagem do projeto, é necessário conectar a placa ao computador através do cabo USB. A partir daí, se no menu "*Tools*" do Arduino, a placa (*Board*) e a porta utilizada (*Serial Port*) estiverem sido selecionadas corretamente, pode-se incluir o código do projeto, copiando-o do "Apêndice A" (página 64) e colando-o no *software* Arduino. Feito isso, é possível verificar se há algum erro no código em "*Verify*" e fazer o "*Upload*" do código para a placa.



FIGURA 11 - Opções Verify e Upload no *software* Arduino (FONTE: Própria autora, 2014)

Caso tenha ocorrido algum problema, verifique se as conexões de placa e porta do menu "*Tools*" da IDE Arduino, e as ligações da protoboard com a placa (FIGURA 12) foram feitas corretamente.

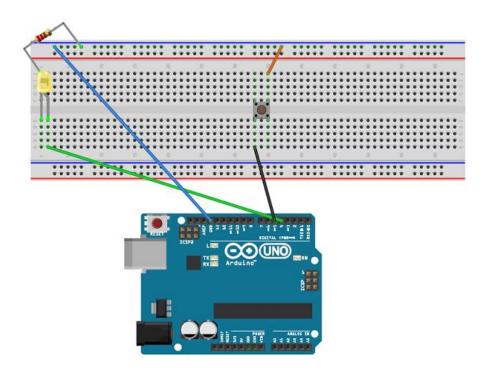


FIGURA 12 - Montagem do projeto Pressionamento de botão para acender LED (FONTE: Própria autora, 2014 – Ferramenta Fritzing)

Observação: Existe um tipo de resistor adequado a ser utilizado com cada LED, no qual, sua principal função é protegê-lo para que não queime. Para saber qual o resistor correto a utilizar, é preciso fazer o cálculo do resistor, através da seguinte fórmula:

Resistor = (Tensão da fonte - Tensão do LED) / Corrente Máxima do LED

Cada LED possui um valor de tensão específico para que o mesmo funcione (acenda), incluindo o valor de sua corrente máxima, conforme os dados da FIGURA 13.

Côr	Queda de Tensão	Corrente Máxima
Vermelho	1.8 V	0.02 A
Verde	2.1 V	0.02 A
Amarelo	2.0 V	0.015 A
Laranja	2.0 V	0.02 A
Azul	3.1 V	0.02 A
Branco	3.1 V a 4.0V (depende do fabricante)	0.02 A
Infra- vermelho	1.1 V	0.02 A

FIGURA 13 - Dados sobre LEDs

(FONTE: Claudio Dias - http://aplnx.blogspot.com.br/2011/11/led-circuito.html)

O valor da tensão da fonte do Arduino Uno é de 5V. Então, para saber qual resistor utilizar com o LED amarelo é preciso realizar o seguinte cálculo:

Resistor = (Tensão da fonte – Tensão do LED) / Corrente Máxima do LED

Resistor = (5V - 2V)/0.015

Resistor = 3/0.015

Resistor = 200Ω

Caso não haja o resistor com o valor exato para o LED, pode ser utilizado um com o valor aproximado, porém, sempre maior que o valor obtido. Ou seja, entre escolher um resistor de 199 Ω ou 201 Ω , é recomendável ficar com o resistor de 201 Ω , pois, um resistor com valor menor permitiria a passagem de corrente elétrica pelo LED, deixando-o propenso a queimar. Se o valor for muito distante do cálculo, como 400 Ω , a intensidade do brilho do LED é reduzida. Para o projeto, foi utilizado um resistor de 220 Ω , por ser o resistor com valor menor dentre todos os resistores que a autora possui.

3.5.2 Estruturas de repetição

- a Conceito: (Veja a descrição do conceito "Estruturas de repetição" na página 33).
- b <u>Problema</u>: Acender LED por um período de repetição de 10 vezes, com intervalo de tempo de 1 segundo.

c Solução prática:

Projeto: Pisca LED com 10 repetições

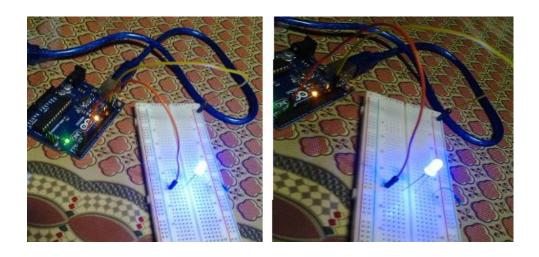


FIGURA 14 - Exemplo de projeto utilizando estrutura de repetição (FONTE: Própria autora, 2014)

Materiais necessários:

- 1 placa Arduino Uno;
- 1 protoboard;
- 1 LED azul;
- 1 cabo USB, do tipo A para B (para conectar a placa ao computador);
- 1 resistor de 95 Ω (ohm) ou com valor maior mais próximo (resistor utilizado =

220 Ω);

• 2 jumpers.

Como fazer:

Para o início deste projeto, deverão ser conectados todos os componentes eletrônicos na protoboard, conforme os seguintes procedimentos:

- 1°) Encaixar o LED azul na protoboard;
- 2°) Colocar o resistor no polo negativo do LED e na coluna negativa da protoboard;
- 3°) Conectar o jumper no polo positivo do LED e no pino 2 da placa;
- 4°) Conectar o jumper na coluna negativa da protoboard e no pino GND da placa.

Terminando todos os procedimentos, a placa deverá ser conectada ao computador através do cabo USB, e o *sketch* que consta no "Apêndice B" (página 65), ao final deste trabalho, precisará ser copiado para o *software* Arduino, fazendo o "*Upload*" do código para a placa.

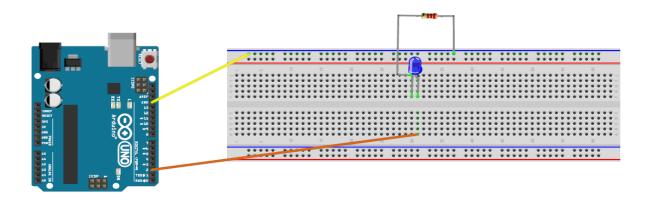


FIGURA 15 - Montagem do projeto Pisca LED com 10 repetições (FONTE: Própria autora, 2014 – Ferramenta Fritzing)

Cálculo do resistor para o LED azul:

Resistor = (Tensão da fonte - Tensão do LED) / Corrente Máxima do LED

Resistor = (5V - 3.1V)/0.02

Resistor = 1.9/0.02

Resistor = 95Ω

3.5.3 Vetores

a Conceito: (Veja a descrição do conceito "Vetores" na página 35).

b <u>Problema</u>: Reproduzir som com as notas musicais DÓ – RÉ – MI – FÁ – SOL – LÁ - SI.

c Solução prática:

Projeto: Reprodução de som com notas musicais

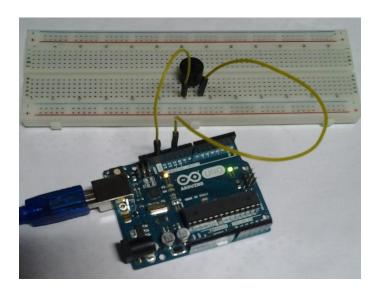


FIGURA 16 - Exemplo de projeto utilizando vetores (FONTE: Própria autora, 2014)

Materiais necessários:

- 1 placa Arduino Uno;
- 1 cabo USB, do tipo A para B (para conectar a placa ao computador);
- 1 buzzer;
- 1 protoboard;
- 2 jumpers.

Como fazer:

A montagem do circuito elétrico deste projeto deverá ser feita conforme os seguintes procedimentos:

- 1°) Encaixar o buzzer na protoboard;
- 2°) Conectar o jumper ao polo negativo do buzzer e ao pino GND da placa;
- 3°) Conectar o jumper ao polo positivo do buzzer e ao pino 11 da placa.

Após a realização de todos os procedimentos, a placa deverá ser conectada ao computador através do cabo USB, e o *sketch* contido no "Apêndice C" (página 66), precisará ser copiado para o *software* Arduino, e transferido para a placa, com o clique na opção "*Upload*".

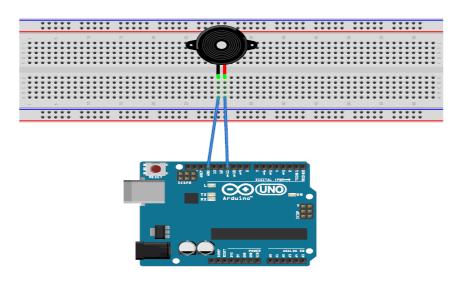


FIGURA 17 - Montagem do projeto Reprodução de som com notas musicais (FONTE: Própria autora, 2014 – Ferramenta Fritzing)

Observação: Para fazer a reprodução de som com as notas musicais, é necessário conhecer a frequência e a duração de cada uma delas. A frequência possui valor definido. Já a duração precisa ser calculada. A fórmula utilizada para esse cálculo é: **duração** = **período/2** = **1**, no qual, a expressão "= 1" quer dizer que o resultado dessa fórmula precisa ser um valor inteiro.

TABELA 1 - Dados das notas musicais

Nota	Frequência	Período	Duração
DÓ	261 Hz	3830	1915
RÉ	294 Hz	3400	1700
MI	329 Hz	3038	1519
FÁ	349 Hz	2864	1432
SOL	392 Hz	2550	1275
LA	440 Hz	2272	1136
SI	493 Hz	2028	1014

(FONTE: Adaptado de um dos exemplos de projeto com som no Site Oficial do Arduino, 2014 - playground.arduino.cc/Portugues/LearningKeyboardSerial)

3.5.4 *Matrizes*

- a Conceito: (Veja a descrição do conceito "Matrizes" na página 36).
- b Problema: Acender todos os LEDs da matriz de LED RGB.
- c Solução prática:

Projeto: Ligação de LEDs na matriz RGB



FIGURA 18 - Exemplo de projeto utilizando matrizes (FONTE: Própria autora, 2014)

Materiais necessários:

- 1 placa Arduino Uno;
- 1 cabo USB, do tipo A para B (para conectar a placa ao computador);
- 1 matriz de LED RGB;
- 6 jumpers.

Como fazer:

Para montar o projeto com matrizes, é preciso seguir os seguintes procedimentos:

- 1°) Conectar o pino VCC da matriz de LED ao pino de 5V da placa (ou ao pino de 3.3V);
- 2°) Conectar o pino 10 da placa ao pino MOSI da matriz de LED;
- 3°) Conectar o pino 8 da placa ao pino VCC da matriz de LED;
- 4°) Conectar o pino 12 da placa ao pino SCLK da matriz de LED;
- 5°) Conectar o pino 11 da placa ao pino CS da matriz de LED.
- 6°) Conectar a placa Arduino Uno ao computador, através do cabo USB;
- 7°) Copiar o sketch do "Apêndice D" (página 67) para o software Arduino;

8°) Fazer o "Upload" do código para a placa.

Observação: Todas as conexões são feitas com o jumper.

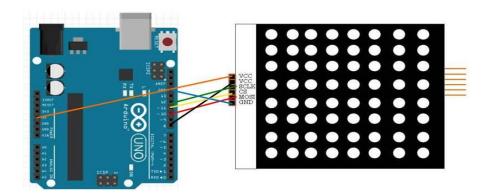


FIGURA 19 - Montagem do projeto Ligação de LEDs na matriz RGB (FONTE: Própria autora, 2014)

3.6 APLICAÇÃO DAS OFICINAS COM OS PROJETOS PRÁTICOS

a <u>E-TIC</u>: É um evento de caráter acadêmico que ocorre anualmente no Instituto Federal Catarinense – Campus Camboriú, objetivando proporcionar o intercâmbio entre pesquisadores, professores e alunos dos cursos técnicos e superiores relacionados à área de Tecnologia da Informação. Neste evento são apresentadas as mais diversas atividades, como: torneio de RoboCode, apresentação de palestras e oferta de oficinas práticas, no qual, os estudantes podem participar da competição de RoboCode e concorrer a prêmios (1°, 2° e 3° lugar), participar/ministrar palestras ou participar/ministrar oficinas.

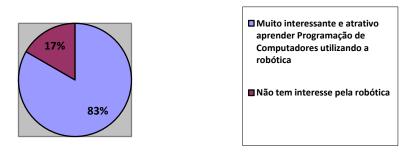
b <u>Escola da rede pública federal</u>: Foram ministradas duas oficinas no Laboratório de Informática Geral I do Instituto Federal Catarinense — Campus Camboriú, com duração de 4 horas cada (2 primeiras horas com conteúdo teórico e 2 últimas horas com atividade prática), nos dias 19 e 21 de maio de 2014, em computadores com o sistema operacional Windows, devido à facilidade de instalação e configuração da plataforma Arduino nesse tipo de sistema. O público-alvo participante dessas oficinas abrangeu tanto alunos dos cursos técnicos quanto dos cursos superiores, sendo ambos,

da área de informática. Foi aplicado, também, um questionário aos alunos da primeira oficina, realizada no dia 19 de maio de 2014, obtendo-se 12 questionários respondidos, correspondente à quantidade de participantes presentes. Esse questionário visou obter a percepção dos alunos em relação à utilização da robótica no estudo de programação de computadores e foi apenas uma forma preliminar de avaliá-la. Por serem oficinas com tempo limitado, e a maioria dos estudantes não possuírem muito conhecimento sobre eletrônica, optou-se pela demonstração de exemplos mais simples, para que os mesmos pudessem acompanhar o que estava sendo feito.

3.6.1 Apresentação dos resultados

Com a aplicação dos questionários, pôde-se perceber que grande parte dos alunos de nível médio/técnico consideram interessante e atrativo aprender Programação de Computadores utilizando a robótica. Dos 6 questionários respondidos, 5 afirmaram ter interesse pela robótica (83%), e apenas 1 (17%) obteve-se a resposta de que não há interesse, conforme o Gráfico 1.

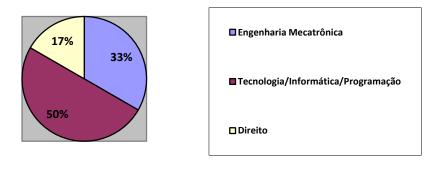
Gráfico 1 – Respostas dos alunos de nível médio sobre suas opiniões em relação à utilização da robótica em Programação de Computadores



(FONTE: Própria autora, 2014)

Foi possível perceber também, que a maior parte desses alunos pretende continuar com os estudos na área de tecnologia/informática/programação, sendo dois no curso de Engenharia Mecatrônica, que está relacionado com a robótica. Apenas um optou pelo curso de Direito, como pode-se notar no Gráfico 2.

Gráfico 2 – Respostas dos alunos de nível médio sobre a área profissional/curso que pretendem seguir



(FONTE: Própria autora, 2014)

Dentre as respostas obtidas sobre a utilização de robô, notou-se que os alunos de nível médio/técnico tiveram mais contato com o uso de robôs do que os estudantes do ensino superior. A partir desses resultados, é possível saber o nível de conhecimento do aluno, e em que grau de dificuldade é viável aplicar a robótica no ensino da programação de computadores, para que as aulas não se tornem tão cansativas. Mas, para que o professor possa considerar essa análise, é necessário que todos os alunos de uma turma respondam o questionário. A seguir, serão apresentados os resultados quanto à utilização de robô pelos estudantes de nível médio (Gráfico 3) e pelos acadêmicos do ensino superior (Gráfico 4).

Gráfico 3 – Alunos de nível médio que já utilizaram robô



(FONTE: Própria autora, 2014)

Conforme consta no Gráfico 3, metade dos alunos do ensino médio responderam que já utilizaram robô, sendo que apenas um deles mencionou o Arduino, e a outra metade responderam que não. Enquanto que no Gráfico 4, pode-se perceber que a maior parte dos alunos do ensino superior afirmaram não ter utilizado robô. Essas informações complementam a questão sobre o conhecimento do aluno em relação à plataforma ou ao *hardware* a ser trabalhado nas aulas.

17%

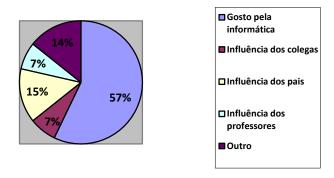
☐ Já utilizou robô
☐ Não utilizou robô

Gráfico 4 – Alunos de Ensino Superior que já utilizaram robô

(FONTE: Própria autora, 2014)

Outra análise que pôde-se realizar foi quanto a motivação do aluno em fazer um curso na área de informática. De modo geral, dos 12 questionários respondidos, entre alunos do nível médio e superior, 8 deles obtiveram-se como respostas que o "Gosto pela informática" foi um fator importante na escolha do curso profissional, dentre esses, alguns alunos incluíram também, que tiveram influência na opção pelo curso da área tecnológica. Todas essas respostas, e as demais, podem ser observadas no Gráfico 5.

Gráfico 5 – Motivação do aluno na escolha do curso profissional



(FONTE: Própria autora)

Através das análises do questionário aplicado foi possível perceber que grande parte dos alunos participantes da oficina possui interesse pela robótica, e que, sendo assim, seria interessante que houvesse mais investimentos nessa área, que tende a crescer com o avanço da tecnologia.

4 CONSIDERAÇÕES FINAIS

Esta pesquisa apresentou um método lúdico de ensino da programação de computadores, denominado pela autora de MLAlg, abordando alguns exemplos práticos de projetos com Arduino que podem ser aplicados por professores nas aulas de Algoritmos e Programação de Computadores.

Devido à extensão do trabalho em relação à quantidade de conteúdo, e as diversas pesquisas que precisaram ser feitas para elaboração do mesmo, não foi possível aplicar as oficinas com algumas turmas de alunos, no entanto, vale ressaltar que esse é um processo muito importante para validação do método de ensino proposto. Porém, foram ministradas duas oficinas práticas de Introdução a Arduino no Instituto Federal Catarinense — Campus Camboriú, com a aplicação de questionário sobre a percepção dos alunos quanto à utilização da robótica em programação de computadores, no qual, pôde-se perceber que a maior parte desses alunos possui interesse em utilizar a robótica como ferramenta de apoio ao estudo da disciplina.

Os projetos com Arduino apresentados, para cada conceito das disciplinas de lógica e programação de computadores, visaram demonstrar a aplicabilidade da robótica no âmbito escolar, e podem ser utilizados tanto por professores que queiram tornar suas aulas mais atrativas e interessantes, quanto por alunos, que desejam experimentar novas formas de aprender a programação de computadores e, talvez, tornar-se um futuro profissional da área.

Para avaliar o aprendizado dos alunos, e também, o interesse pela disciplina de Algoritmos e Programação de Computadores, os professores poderão aplicar provas, trabalhos e exercícios de fixação, comparando as notas dos alunos nas atividades realizadas, com as notas obtidas utilizando o método tradicional de ensino

da programação.

Trabalhos futuros

Tendo em vista que as oficinas realizadas foram apenas para verificar a aceitação do método proposto, sugere-se que este seja aplicado em forma de oficinas em outras escolas da região com a finalidade de verificar sua aplicabilidade.

E, ainda a adoção do método em um piloto na disciplina de Algoritmos e Programação de Computadores do Instituto Federal Catarinense – Campus Camboriú, para avaliar o aprendizado dos alunos frente a outras turmas que estejam usando o procedimento tradicional de ensino da lógica.

REFERÊNCIAS BIBLIOGRÁFICAS

ARDUINO. **Arduino** (**Site**). 2014. Disponível em: <www.arduino.cc>. Acesso em: 05 set. 2014.

BARBOSA, L. F. W. A Utilização da Robótica como Ferramenta Multidisciplinar no Ensino da Engenharia Elétrica. In: CBA: Congresso Brasileiro de Automática, n. 17: 2008, Juiz de Fora, MG. Disponível em: http://www.labplan.ufsc.br/congressos/cba2008/textos/CBA_2008_Artigos/37941.pd f>. Acesso em: 09 ago. 2014.

CASTILHO, M.; SILVA, F.; WEINGAERTNER, D. **Algoritmos e Estrutura de Dados I**. 2011. Disponível em: <www.inf.ufpr.br/cursos/ci055/apostila.pdf>. Acesso em: 03 set. 2014.

CISCO. Estudo revela demanda crescente por profissionais de tecnologia no Brasil, mas há falta de mão de obra qualificada. 2013. Disponível em: http://globalnewsroom.cisco.com/pt/br/release/Estudo-revela-demanda-crescente-por-profissionais-de-tecnologia-no-Brasil-mas-h%C3%A1-falta-1688801). Acesso em: 16 mai. 2014.

DETERS, J. I *et. al.* **O Desafio de Trabalhar com Alunos Repetentes na Disciplina de Algoritmos e Programação**. Florianópolis: Serviço Nacional de Aprendizagem Industrial – SENAI, 2008; Itajaí: Universidade do Vale do Itajaí – UNIVALI, 2008; São José: Universidade do Vale do Itajaí – UNIVALI, 2008. Disponível em: http://proativa.virtual.ufc.br/sbie/CD_ROM_COMPLETO/workshops/workshop%20/2/O%20Desafio%20de%20Trabalhar%20com%20Alunos%20Repetentes%20na.pdf>. Acesso em: 21 abr. 2014.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. **Lógica de programação:** a construção de algoritmos e estruturas de dados. 3ª ed. São Paulo: Prentice Hall, 2005.

HINTERHOLZ JR., O. **Tepequém:** uma nova ferramenta para o ensino de algoritmos nos cursos superiores em computação. Roraima (Boa Vista): Faculdade Atual da Amazônia – FAA, 2009. Disponível em: http://csbc2009.inf.ufrgs.br/anais/pdf/wei/st02_04.pdf >. Acesso em: 21 abr. 2014.

ITAJAI. **Escola de Itajaí recebe Oficina de Robótica**. 2014. Disponível em: http://www.itajai.sc.gov.br/noticia/8807/escola-de-itajai-recebe-oficina-de-robotica#.VFa1OvnF8R4. Acesso em: 03 ago. 2014.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. de. **Algoritmos:** lógica para desenvolvimento de programação de computadores. 24ª ed. rev. São Paulo: Érica, 2010.

MARJI, M. **Aprenda a programar com scratch:** uma introdução visual à programação com jogos, arte, ciência e matemática. 1ª ed. São Paulo: Novatec, 2014.

MCROBERTS, M. **Arduino básico**. Tradução: Rafael Zanolli. São Paulo: Novatec Editora, 2011.

MONK, S. **Programação com Arduino: começando com Sketches**. Tradução: Anatólio Laschuk. Porto Alegre: Bookman, 2013. 148 p.

MULTILOGICASHOP. **Fritzing, um programa livre para projetos e testes com Arduino**. Disponível em: http://multilogica-shop.com/content/fritzing-um-programa-livre-para-projetos-e-testes-com-arduino. Acesso em: 02 out. 2014.

NETO, W. C. B. Ferramenta computacional de apoio ao processo de ensino-aprendizagem dos fundamentos de programação de computadores. 2007. Disponível em: http://www.lbd.dcc.ufmg.br/colecoes/sbie/2007/0022.pdf >. Acesso em: 02 out. 2014.

ONACIONAL. 1ª Olimpíada de Robótica Educativa Livre da UPF está com inscrições abertas. 2013. Disponível em:

http://www.onacional.com.br/geral/cidade/44004/1+0limpiada+de+robotica+educativa+livre+da+upf+esta+com+inscricoes+abertas. Acesso em: 03 ago. 2014.

PIVA JR., D.; FREITAS, R. L. **Estratégias para melhorar os processos de Abstração na disciplina de Algoritmos**. 2011. Disponível em: http://www.dimap.ufrn.br/csbc2011/anais/eventos/contents/WEI/Wei_Secao_1_Artigology.html o_1_Piva_Jr.pdf>. Acesso em: 15 mai. 2014.

PRIETCH, S. S.; PAZETO, T. A. **Estudo sobre a Evasão em um Curso de Licenciatura em Informática e Considerações para Melhorias**. 2010. Disponível em: http://www.cesmac.com.br/erbase2010/papers/weibase/65258.pdf>. Acesso em: 15 mai. 2014.

RIBEIRO, P. C.; MARTINS, C. B.; BERNARDINI, F. C. A Utilização da Robótica no Ensino de Disciplinas de Programação em Cursos de Computação e Engenharia. In: ENINED: Encontro Nacional de Informática e Educação, n. 2: 2011, Cascavel, PR. Disponível em:http://200.201.81.50/enined/anais/artigos_enined/A35.pdf>. Acesso em: 09 ago. 2014.

ROMÃO, L. M.; SACCHELLI, C. M. Uma Proposta Construtivista na

Aprendizagem dos Conceitos da Física com o Auxílio da Robótica Educacional.

In: SEMINFO: Seminário de Informática, n. 8: 2009, Torres, RS. Disponível em: http://www.seminfo.com.br/anais/2009/pdfs/WEI_Tche/63217_1.pdf. Acesso em: 09 ago. 2014.

SILVA, A. C.; LUCHE, F. D.; GOULART, E.; AGUIAR, V. P. **Aplicação da Robótica no Ensino Fundamental:** um estudo de caso. Revista de Informática Aplicada, São Caetano do Sul, SP, v.5, n.2, p.1-18, 01 nov. 2009. Bimestral. Disponível em:

http://seer.uscs.edu.br/index.php/revista_informatica_aplicada/article/view/991/805. Acesso em: 07 jun. 2014.

SOBJAK, R. **Algoritmos e lógica da programação**. 2011. Disponível em: <sobjak.files.wordpress.com/2011/07/apostila-de-algoritmos-e-lc3b3gica-da-programac3a7c3a3o.pdf>. Acesso em: 03 set. 2014.

SUPERGEEKS. **SuperGeeks** (**Site**). Disponível em: http://supergeeks.com.br>. Acesso em: 19 abr. 2014.

UFMS. **DESTACOM**: tutorial de alice. 2013. Disponível em: http://destacom.ufms.br/wiki/Tutorial_de_Alice#Introdu.C3.A7.C3.A3o>. Acesso em: 20 ago. 2014.

WERNER, H.; GASPAR JR., G. P. Utilização de um carro-robô construído com componentes de baixo custo como ferramenta de apoio ao aprendizado dos conceitos de orientação a objetos. 2001. Disponível em:

http://lsm.dei.uc.pt/ribie/docfiles/txt200373118616paper-201.pdf. Acesso em: 05 mai. 2014.

APÊNDICES

APÊNDICE A - SKETCH ESTRUTUR	A DE
DECISÃO64	
APÊNDICE B – SKETCH ESTRUTUR	A DE REPETIÇÃO3
APÊNDICE C - SKETCH VETOR	4
A PÊNDICE D _ SKETCH MATRIZ	5

APÊNDICE A – SKETCH ESTRUTURA DE DECISÃO

```
int ledAmarelo = 4; //declara a variável ledAmarelo, atribuindo o
valor 4, que corresponde ao pino a ser conectado na placa
int pinoBotao = 5; //declara a variável pinoBotao, atribuindo o
valor 5, que corresponde ao pino a ser conectado na placa
int estadoBotao = 0; //declara e inicializa a variável estadoBotao
com o valor 0
void setup()
 pinMode(ledAmarelo, OUTPUT); //define o ledAmarelo como dado de
saída (OUTPUT)
 pinMode(pinoBotao, INPUT PULLUP); //define o pinoBotao como dado
de entrada do tipo INPUT PULLUP
}
void loop()
  estadoBotao = digitalRead(pinoBotao); //faz a leitura do
estadoBotao em pinoBotao
 //ledAmarelo permanece apagado (LOW) enquanto não for pressionado
o botão
  if(estadoBotao == LOW) //verifica se estadoBotao não mudou (se foi
pressionado)
    digitalWrite(ledAmarelo, HIGH); //acende ledAmarelo
  }else{
   digitalWrite(ledAmarelo, LOW); //apaga ledAmarelo
  }
}
```

APÊNDICE B - SKETCH ESTRUTURA DE REPETIÇÃO

```
int ledPin = 2; //declara a variável ledPin, atribuindo o valor 2,
que corresponde ao pino a ser conectado na placa
void setup() {
 // define ledPin como saída
 pinMode(ledPin, OUTPUT);
void loop() {
 for (contador = 0; contador < 10; contador++) { //enquanto o</pre>
contador não for = 10, repete o que está dentro do laço de repetição
   digitalWrite(ledPin, HIGH);    //define ledPin como nível alto
(ligado)
   delay(1000);
                          //espera 1 segundo
   (desligado)
  delay(1000);
                   //espera 1 segundo
  contador = contador + 1; //incrementa o contador em 1 a cada
repetição do processo até chegar a 10
 }
}
```

APÊNDICE C – SKETCH VETOR

```
//define constante para cada nota musical
#define DO 261
#define RE 294
#define MI 329
#define FA 349
#define SOL 392
#define LA 440
#define SI 493
//define o pino 11 a ser utilizado na placa
int pino = 11;
//cria um vetor com todas as 7 notas musicais
int notas musicais[] = {DO, RE, MI, FA, SOL, LA, SI};
//cria um vetor com a duração de cada uma das 7 notas musicais
int duracaodecadanota[] = {100,100,100,100,100,100};
void setup() {
   //"for" utilizado para garantir que as 7 notas sejam tocadas,
começando no 0 até 7
        for (int nota musical = 0; nota musical < 7; nota musical++)</pre>
{ //enquanto o valor for menor que 7, incrementa-se o contador
              //atribui um tempo de duração para cada nota musical
              int duracao nota = duracaodecadanota[nota musical];
              //gera o som no pino
              tone(pino, notas musicais[nota musical], duracao nota);
              //pausa depois das notas
              int pausa nota musical[]
=\{1000,1000,1000,1000,1000,1000,1000\};
              delay(pausa nota musical[nota musical]);}
void loop() {
}
```

APÊNDICE D – SKETCH MATRIZ

```
//Define as cores normais (contidas no datasheet do controlador)
#define BLACK 0
#define RED 0xE0
#define GREEN 0x1C
#define BLUE 0x03
#define ORANGE 0xFC
#define MAGENTA 0xE3
#define TEAL 0x1F
#define WHITE OxFF
//Define os pinos do Arduino a serem conectados na matriz de LED
int saida = 10;
int entrada = 8;
int sclk = 12;
int cs = 11;
//Define as variáveis que precisam conter no programa
char cor buffer [64]; //64 representa a quantidade de LEDs da matriz
(8x8=64)
void setup()
  //configuração SPI (representa os dados de entrada e saída da
matriz de LED)
 SPCR = (1 << SPE) | (1 << MSTR) | (1 << SPR1); // Habilita SPI em nível HIGH,
Modo Master, e divide clock por 16
  //Define o modo do pino para a matriz RGB
  pinMode(saida, OUTPUT);
  pinMode(entrada, INPUT);
  pinMode(sclk,OUTPUT);
  pinMode(cs,OUTPUT);
  //Faz a matriz RGB estar desativada
  digitalWrite(cs, HIGH);
}
void loop()
  //Carrega cores dentro da primeira linha de cor do buffer array.
  //Este é um array de cores definidas para a matriz RGB.
  cor buffer[0]=BLACK; //primeiro valor do array inicia em 0
  cor buffer[1]=RED;
  cor buffer[2]=GREEN;
  cor buffer[3]=BLUE;
```

```
cor buffer[4] = ORANGE;
  cor buffer[5] = MAGENTA;
  cor buffer[6]=TEAL;
  cor buffer[7]=WHITE;
  //Ativação da matriz RGB
  digitalWrite(cs, LOW);
  //Envia a cor do buffer para a matriz RGB
  for(int LED=0; LED<64; LED++) {</pre>
    spi transferencia(cor buffer[LED]);
  }
  //Desativação da matriz RGB
  digitalWrite(cs, HIGH);
  while (1);
}
//Este comando é usado para enviar um único valor de cor para a
matriz RGB (tem 64 valores de cores que são enviados para a matriz
RGB)
char spi transferencia(volatile char dado)
  SPDR = dado;
                                   //Começa a transmissão
  while (!(SPSR & (1<<SPIF)))
                                  //Espera até o fim da transmissão
  {
  };
  return SPDR; // retorna o byte recebido
```

(FONTE: Código adaptado de "https://www.robocore.net/modules.php?name=GR_LojaVirtual&prod=179", em "Código para Arduino")

ANEXOS

ANEXO A - PÁGINA DE DOWNLOAD DA IDE ARDUINO NO SITE	
OFICIAL	
70	
ANEXO B – LOCALIZAÇÃO DOS ARQUIVOS A SEREM BAIXADOS NO	\mathbf{C}
SITE OFICIAL DO ARDUINO (PARA WINDOWS)	
ANEXO C – ARQUIVO DE INSTALAÇÃO DO ARDUINO	
ANEXO D – ARQUIVO DE INSTALAÇÃO DO PACOTE DE DRIVERS DA	AS
PLACAS ARDUINO (VERSÃO 64 BITS)	
ANEXO E – TELA INICIAR DO WINDOWS 8	
ANEXO F – MENU INICIAR DO WINDOWS 7	
ANEXO G – SELEÇÃO DA OPÇÃO "PROPRIEDADES DO SISTEMA"	8
ANEXO H – SELEÇÃO DA OPÇÃO "GERENCIADOR DE DISPOSITIVO	
ANEXO I – SELEÇÃO DA OPÇÃO "PORTAS (COM E LPT)"	
ANEXO J – SELEÇÃO DA PORTA COM	
ANEXO K – SELEÇÃO DA PLACA A SER UTILIZADA	
ANEXO L – PROJETO MÚSICA TEMA MARIO	
ANEXO M – LETREIRO DE NATAL	

ANEXO A – PÁGINA DE DOWNLOAD DA IDE ARDUINO NO SITE OFICIAL



Download the Arduino Software

The open-source Arduino environment makes it easy to write code and upload it to the i/o board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing, avr-gcc, and other open source software.

THE ARDUINO SOFTWARE IS PROVIDED TO YOU "AS IS," AND WE MAKE NO EXPRESS OR IMPLIED MARRANTIES WHATSOEVER WITH RESPECT TO ITS FUNCTIONALITY, OPERABILITY, OR USE, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR INFRINCEMENT. WE EXPRESSLY DISCLAIM ANY LIABILITY WHATSOEVER FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST REVENUES, LOST PROFITS, LOSSES RESULTING FROM BUSINESS INTERRUPTION OR LOSS OF DATA, RECARDLESS OF THE



(FONTE: www.arduino.cc)

ANEXO B – LOCALIZAÇÃO DOS ARQUIVOS A SEREM BAIXADOS NO SITE OFICIAL DO ARDUINO (PARA WINDOWS)



Arduino IDE

Arduino 1.0.6

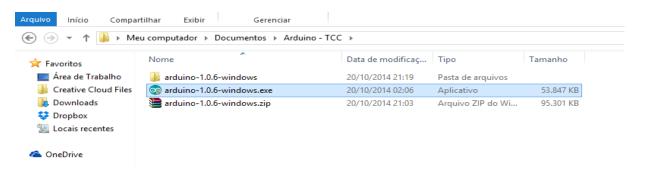
· source

Download Next steps Arduino 1.0.6 (release notes): Getting Started Windows Installer, Windows ZIP file (for non-administrator installe) Reference - Mac OS X Environment - Linux: 32 bit, 64 bit Examples

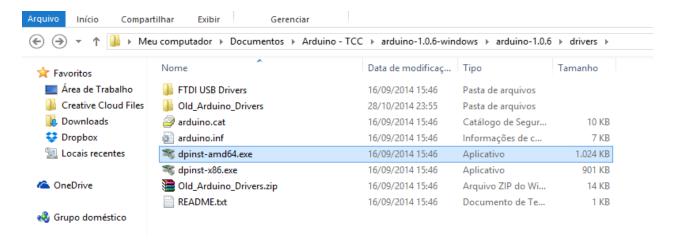
(FONTE: www.arduino.cc)

Foundations

ANEXO C – ARQUIVO DE INSTALAÇÃO DO ARDUINO



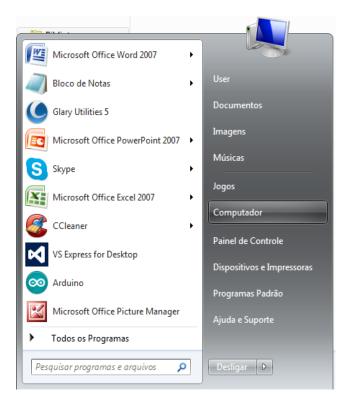
ANEXO D – ARQUIVO DE INSTALAÇÃO DO PACOTE DE DRIVERS DAS PLACAS ARDUINO (VERSÃO 64 BITS)



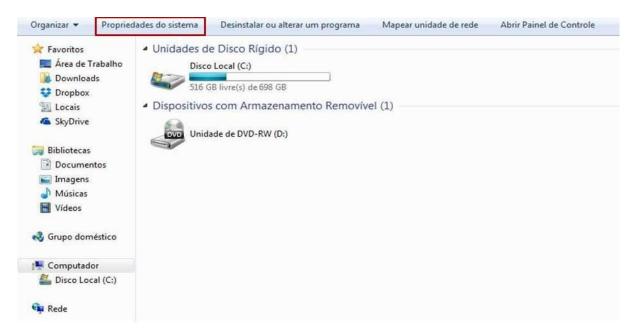
ANEXO E – TELA INICIAR DO WINDOWS 8



ANEXO F – MENU INICIAR DO WINDOWS 7



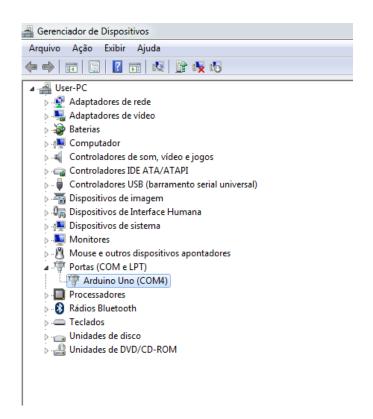
ANEXO G – SELEÇÃO DA OPÇÃO "PROPRIEDADES DO SISTEMA"



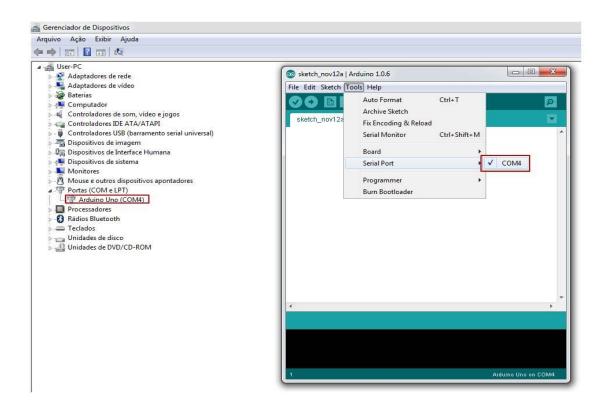
ANEXO H – SELEÇÃO DA OPÇÃO "GERENCIADOR DE DISPOSITIVOS"



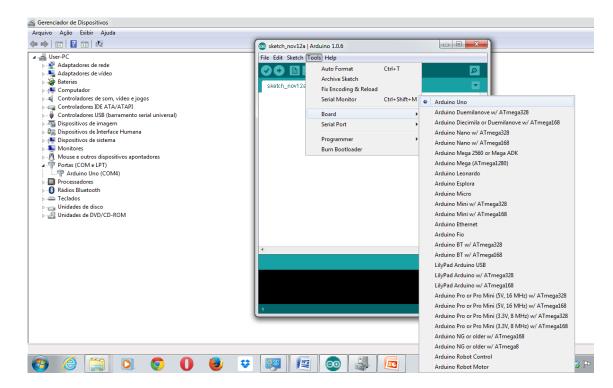
ANEXO I – SELEÇÃO DA OPÇÃO "PORTAS (COM E LPT)"



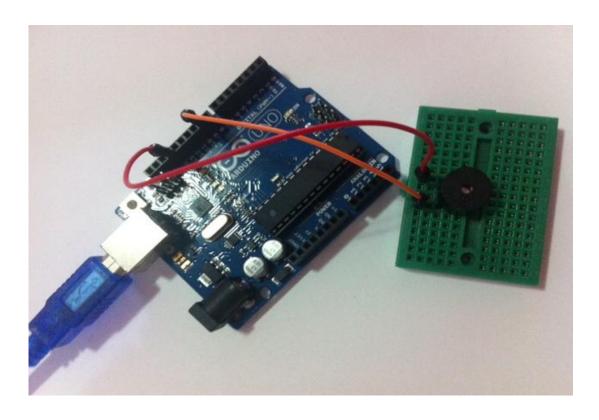
ANEXO J – SELEÇÃO DA PORTA COM



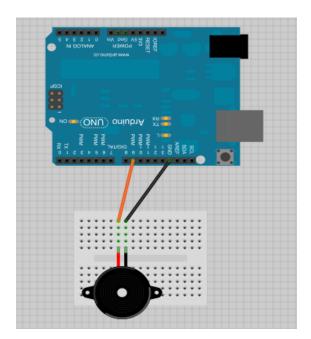
ANEXO K – SELEÇÃO DA PLACA A SER UTILIZADA



ANEXO L – PROJETO MÚSICA TEMA MARIO



FONTE: Núbia Souza (www.nubiasouza.com.br/musica-tema-mario-arduino-speaker/)



FONTE: Núbia Souza

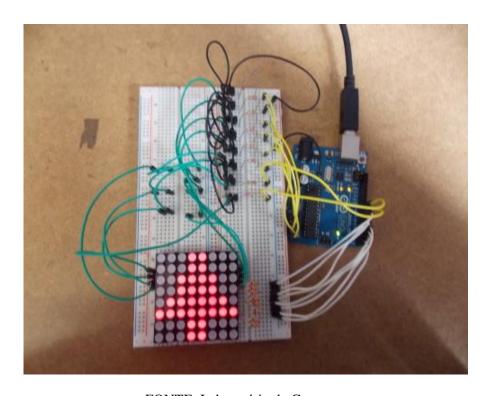
(www.nubiasouza.com.br/musica-tema-mario-arduino-speaker/)

```
int portSpeak(9); //porta ligada no speaker
//melodia do MARIO THEME
int melodia[] =
,380,430,500,430,500,570,500,760,720,680,620,650,1020,1020,1020,1020,780,760,720,680,620,650,380,430,500,430,500,430,500,570,585,550,500,380,500,500,500,700,700,680,620,650,380,430,500,430,500
60,500,500,500,500,580,660,500,430,380,660,660,660,510,660,770,380};
//duraçao de cada nota
int duracaodasnotas[] =
00,100,100};
void setup() {
//for para tocar as 156 notas começando no 0 ate 156 ++ incrementado
  for (int nota = 0; nota < 156; nota++) {
    int duracaodanota = duracaodasnotas[nota];
    tone(portSpeak, melodia[nota],duracaodanota);
//pausa depois das notas
    int pausadepoisdasnotas[] =
,300,350,150,350,150,300,150,600,150,300,300,100,300,550,575};
    delay(pausadepoisdasnotas[nota]);}
    noTone(portSpeak);
void loop() {
```

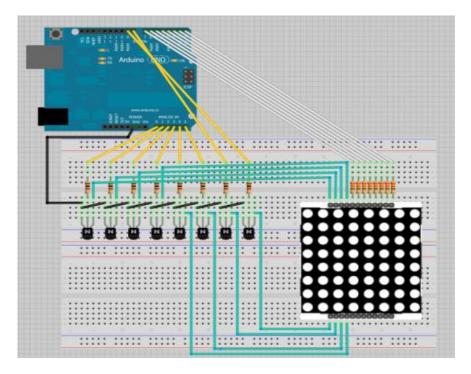
FONTE: Núbia Souza

(www.nubiasouza.com.br/musica-tema-mario-arduino-speaker/)

ANEXO M – LETREIRO DE NATAL



FONTE: Laboratório de Garagem (http://labdegaragem.com/profiles/blogs/letreiro-de-natal-display-8x8-enviando-sua-mensagem-de-natal)



FONTE: Laboratório de Garagem (http://labdegaragem.com/profiles/blogs/letreiro-de-natal-display-8x8-enviando-sua-mensagem-de-natal)

```
byte texto[] = {//Matriz onde o texto é colocado para que possa aparecer no display
  0x0, 0x0,
0x0, 0xFF,
                                                                     Ox0,//Espaço
                   CocC.
                             OxeO,
                                       OxeO,
                                                  Coco,
                                                           OxeO,
                                                          0x3,
                  Oxff,
                            0x3,
                                      Ox3,
                                                Ox3,
  0x0, 0xE,
0x0, 0x7E,
                            0x33.
                                      0×33.
                                                Owers.
                                                          Ower.
                  Ox1F.
                                                                    0x0_//d.
                  OMFF,
                                                OxDF,
                                                           OxSE,
                                                0x15,
                            0x15,
                                                          0x15,
  0x0, 0x15,
0x0, 0xff,
                  0x15,
                                       0x15.
                                                                    Oxe0_//-
                                                Cocco,
                  Codes.
                            CodD5,
                                       OxD5,
                                                           OscCO,
                                                                    0x0,//3
                  OxFF,
  OMO, OMFF,
                            CocDB,
                                      CocDB,
                                                OscES,
                                                          CocCS,
                                                                    0x0,//E
                                                Ox3,
  OscO, OscFF,
                  Codes,
                                                           Ox3,
                            Ox3,
                                      Ox3,
                                                                    0x00,//L
                                                          OxeD,
  0x0, 0x0,
0x0, 0x07,
                  Oxo,
OxoEF,
                            CodEF,
                                      OxDF,
                                                Owo.
                            OxDF.
                            0x0s,
0x0,
0x70, 0x38,
0x8C,
                                     OxFB,
                                                OxF3,
                                                           OxES,
                                                                   0x0,//z
                                                0x0,//Espaço
  Case C.
         Coco,
                  CocC,
                                        0x55, 0x
0x55, 0x
0x55,
                    OxEO, Cx.
OxEO, Cx.
                                                            0xff, 0xu,...
0x0,//A
           OxFF,
                                                                      Osc0 . / /80
                                                   Ox1C,
  OscO.
                  Oxeco,
                                                          Oxeco,
  0x0, 0x00,
                  Oxff,
  0x0, 0x7F,
                            OxSC,
                                     OxeSC,
                                                Oxff,
                                                          Ox7F,
                                                                    0x0,//A
                 Oxff,
                                     0x3,
0x0,
                                                          0x3,
0x0,
  Omo, Omff,
                            0x3,
                                                0x3,
                                                                    0x0
                              Osco,
                                                  Osco,
                                                                      0x0,//Espaço
  Osco,
         CocC.
                    Coco,
int ponteiro;
int columa;
Intiquadros.
Void setup()
 DDRC - B00011111; //configura os pinos de A0 até A5 como salda digital.
 DDRB - 800000011; //configura os pinos 8 e 9 como salda digital.
 DDRD - B11111111; //configura os pinos de 0 até 7 como salda digital.
ì
vold floop()
 PORTC = 800000000; //Mantem as colunas da matriz desigadas
 PORTB - B00000000; //Mantem as colunas da matriz desligadas
 for(ponteiro=0; ponteiro < 126;ponteiro++); //Carrega o ponteiro em zero e limita ele para o tamanho da matriz
  for (quadros =0; quadros < 5;quadros++ ){//inida o contador de quadros por passagem
   for(coluna=0; coluna < 8; coluna++) // Inicia Contador das Colunas
    f (coluna-s=5)
      PORTB - B00000000;//Delxa pinos 8 e 9 desigados
      PORTC - B00000001 < coluna;//Seta o bit da coluna que será acessa.
// OBS: No lugar do " " "da linha adma, coloque outro sinal de menor que " < "
    else
    4
      PORTC - B00000000;//Delxa pinos de A0 a A5 desigados
      PORTB - B00000001 <" coluna-6;//Seta o bit da coluna que será acessa
// OBS: No lugar do " " "da linha adma, coloque outro sinal de menor que " < "
   PORTD - (texto[pontelro-coluna])://Carrega o PORTD(pinos 0 a 7 do Arduino) com o valor indicado pela matriz
   delay(3);// Espera para que o LED Acenda e em seguida apaga os LEDs.
PORTO - 0x00:
   3
  3
 3
```

FONTE: Laboratório de Garagem

(http://labdegaragem.com/profiles/blogs/letreiro-de-natal-display-8x8-enviando-sua-mensagem-de-natal)