

DESENVOLVIMENTO DE BIBLIOTECA DE FUNÇÕES PARA ROBÔS CONSTRUÍDOS SOB A PLATAFORMA ARDUINO

**Joaquim Flávio Almeida Quirino Gomes¹, Diego de Castro Rodrigues², Marcos Dias da Conceição³,
Dêmis Carlos Fonseca Gomes³**

¹ Estudante do curso Técnico em Informática Integrado ao Ensino Médio do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins – IFTO, Campus Dianópolis.

² Professor do Ensino Básico, Técnico e Tecnológico do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins – IFTO, campus Dianópolis.

³ Professor do Ensino Básico, Técnico e Tecnológico do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins – IFTO, campus Dianópolis. Membro do grupo de Educação, Inovação e Tecnologia do Tocantins (GEDAITT).

Resumo: A robótica é uma área que precisa ser explorada desde a idade juvenil e em ambiente escolar. E assim, a partir da necessidade do desenvolvimento de robôs para competições e conseqüentemente das dificuldades relacionadas à programação no desenvolvimento destes, este estudo tem como principal objetivo desenvolver uma biblioteca de funções em C++ a fim de facilitar o trabalho de alunos participantes de competições de robótica que utilizam a plataforma Arduino. Objetiva-se ainda neste trabalho demonstrar a utilização e eficiência da referida biblioteca em diferentes formatos de robôs para competição (resgate, explorador e seguidor de linha). Os resultados mostram que, após a implementação da solução, é possível reduzir o tempo para programação dos mesmos robôs que não utilizam a referida biblioteca, além de desenvolvermos plataformas robóticas com códigos-fontes de fácil entendimento e mais intuitivo, auxiliando todos os grupos de robótica que esta solução utilizarem.

Palavras-chave: Arduino, Biblioteca de Funções, Competição, GuaráTeca, Robótica.

1. INTRODUÇÃO

A partir da revolução industrial, muitas atividades da vida diárias das pessoas foram facilitadas. Desde máquinas de linha de montagem para a fabricação de carros, até aquelas que auxiliam os humanos nos trabalhos médicos para salvar vidas, e ainda aquelas que trabalham de forma autônoma para salvar pessoas em ambiente de desastres, poupando profissionais e mitigando riscos.

Para (OBR, 2016), “[...]a robótica tende a se tornar uma das dez maiores áreas de pesquisa na próxima década”, logo, é uma área que precisa ser explorada, desde a idade juvenil e em ambiente escolar. Ainda de acordo com este autor “a robótica é uma tecnologia emergente que tem se tornado elemento praticamente obrigatório nas escolas modernas devido à sua possibilidade de atuação em diversas dimensões”. E a partir desses princípios, o grupo de estudos e pesquisas em programação e robótica (GuaráBots) iniciou seus trabalhos sobre essa temática ainda no ano de 2015 no campus Dianópolis do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins (IFTO) através da participação na Olimpíada Brasileira de Informática (OBI) e na Olimpíada Brasileira de Robótica (OBR). Ainda conforme (OBR, 2016), a OBR “é uma olimpíada científica e utiliza da temática da robótica para estimular jovens de todo o Brasil às carreiras científico-tecnológicas, identificar talentos e promover discussões no processo de ensino e aprendizagem no país”.

E, dentre as várias categorias desta competição, o grupo GuaráBots tem participado da modalidade prática, nível 2 (categoria de resgate), destinada a alunos do ensino de nível médio e/ou técnico. Segundo (OBR, 2015, p. 5), “a missão da OBR prática caracteriza-se por simular um ambiente de desastre em mundo real onde o resgate de vítimas precisa ser feito por robôs. Em um ambiente hostil, muito perigoso para o ser humano, um robô desenvolvido pela equipe de estudantes recebe uma tarefa muito difícil: construir um robô completamente autônomo para resgatar vítimas sem interferência humana”.

E assim, a partir da necessidade do desenvolvimento desses robôs para a OBR iniciou-se os estudos sobre as mais variadas plataformas robóticas para esta finalidade e as linguagens de programação a serem utilizadas. Dentre as plataformas estudadas e trabalhadas podemos citar o Arduino, o qual, para (McRoberts, 2015), trata-se de “um computador minúsculo que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos que conectar a ele”, além das linguagens de programação: Arduino (baseada na linguagem C), C, C++, C# e NXT-G®.

Por conseguinte, ao longo dos trabalhos, verificou-se uma ligeira necessidade de capacitação e aprimoramento dos membros do grupo de estudos no menor tempo possível e sem perda na qualidade de ensino. Assim também como foram detectadas dificuldades relacionadas à programação no desenvolvimento desses robôs, principalmente em relação às bibliotecas de funções (as quais são escritas em língua inglesa) para acesso às suas funcionalidades, como ações dos motores, sensores, luzes, placas de expansão e outras, utilizando a linguagem de programação nativa do Arduino.

Desta feita, estas dificuldades motivaram este estudo, no sentido de criar uma alternativa que pudesse tornar a experiência em programação mais fácil, prática e intuitiva a partir da ocultação de detalhes técnicos da linguagem de programação utilizada para o Arduino.

E assim, tomou-se por base a pergunta: que solução desenvolver para alunos iniciantes programarem de forma mais fácil e intuitiva robôs construídos com Arduino?

Logo, esta problemática foi o principal ponto de partida para o avanço do estado da arte da temática ora trabalhada. Conforme (Aristóteles, 1984) há mais saber e conhecimento na arte do que na experiência. Para este autor os homens de arte são mais sábios que os empíricos. E assim, o estado da arte indica o ponto em que o produto ou técnica deixa de ser apenas um projeto para se tornar uma obra-prima. Desta feita, para o avanço do estado da arte e solução da problemática exposta neste trabalho, foi desenvolvida uma biblioteca de funções (métodos), que, para (Schildt, 1997, p. 277) é uma coleção de funções, onde, para este trabalho é capaz de simplificar o processo de programação de robôs para competição, a “GuaráTeca”.

O trabalho aqui proposto, denominado “GuaráTeca”, tem seu nome baseado na denominação do grupo de estudos e pesquisas em programação e robótica do campus Dianópolis do IFTO, o GuaráBots, e tem como propósito, controlar as mais diversas estruturas robóticas de Arduino para competição. Com isso, é possível que o usuário (aluno) possa criar robôs para competições diversas, de forma mais fácil e intuitiva, pois esta solução facilita o trabalho eliminando a necessidade da inclusão de diversas linhas de códigos as quais já foram implementadas na biblioteca. Utiliza ainda termos em língua portuguesa, tornando mais dinâmica a atividade de programar os robôs, tendo o resultado final (robô pronto) mais rápido. E assim, tornando a “GuaráTeca” um produto único no mercado.

Ressalta-se que, durante a execução deste estudo identificou-se apenas uma solução disponível no mercado cujo objetivo seja o de facilitar o trabalho do aluno ao se utilizar Arduino com motores e sensores diversos de forma completa, a *Robot Library*, uma biblioteca de funções produzida pela equipe que desenvolve o Arduino. Conforme (Arduino, 2016), trata-se de uma biblioteca projetada para acessar facilmente as funcionalidades do Arduino *Robot*, que, de acordo

com o mesmo autor, é o primeiro Arduino oficial sobre rodas, o qual possui motores e vários sensores. Contudo, esta biblioteca foi desenvolvida unicamente para a programação específica do *Arduino Robot*, o qual possui um custo relatividade elevado, dificultando o trabalho da equipe, já que para utilizar as funcionalidades da biblioteca é necessário a aquisição deste robô. Verificou-se ainda a ampla utilização por equipes de desenvolvimento de robôs de competição da biblioteca “AF_Motor Library”, produzida pela Adafruit, que, segundo a (Adafruit, 2016), é uma biblioteca de funções para controle apenas de motores, e, com uso de uma placa de expansão para o Arduino, denominada *motorshield*.

Sendo assim, diante do exposto até aqui, aliar eficiência, simplicidade e economia, podem ser indispensáveis ao sucesso de uma equipe de robótica em preparação para competições, adjetivos proporcionados após a criação da “GuaráTeca”, uma biblioteca de funções a ser utilizada em qualquer robô construído sob a plataforma Arduino, um dos grandes diferenciais deste trabalho.

Partindo desse pressuposto, o objetivo geral desta pesquisa foi: desenvolver uma biblioteca de funções (métodos) a fim de facilitar o trabalho de alunos participantes de competições de robótica que utilizam a plataforma Arduino. E como objetivos específicos: demonstrar a utilização e eficiência da referida biblioteca em diferentes formatos de robôs para competição (resgate, explorador e seguidor de linha) e analisar a experiência dos alunos em relação ao uso da “GuaráTeca”.

1.1 IMPLEMENTAÇÃO E TESTES

A biblioteca tem como principal finalidade o controle dos componentes conectados ao Arduino, tendo sido implementados na biblioteca controladores para: motores DC’s e servos, sensores ultrassônicos, sensores de refletância, sensores de cor, sensores giroscópios, sensores de condução de energia e lâmpadas led.

Ressalta-se a necessidade da utilização da biblioteca “AF_Motor” para fazer um melhor uso quando for utilizado uma *motorshield* no Arduino, sendo necessário a inclusão dessa biblioteca nos programas para controlar os robôs.

E assim, chegou-se aos dois arquivos que compõem a “GuaráTeca”: “GuaraTeca.h”, arquivo onde estão implementadas todas as classes para controle de um robô (interface) e a *GuaraTeca.cpp*, o qual estão implementadas todas as ações de um robô em ambiente de competição, como andar para frente, para trás, para esquerda, para a direita, parar, acender um led, manipular uma garra, ler valores dos sensores ultrassônico, giroscópio, de refletância, de cor e de condução de energia.

Vale ressaltar que foi necessária alteração no arquivo da biblioteca “AF_Motor.h” para que fosse possível controlar os motores independente da ordem que forem escritos pelo programador no código de cada robô.

Abaixo temos partes dos códigos dos arquivos “GuaraTeca.h” e “GuaraTeca.cpp”, respectivamente.

```

GuaraTeca.cpp*  GuaraTeca.h*
Miscellaneous Files  HRobot

#ifndef GUARATECA_H
#define GUARATECA_H
#include <Arduino.h>
#include <string.h>

class MRobot{
public:
    MRobot(int conection1, int conection2,
           int velocidade, int tempDLCMotor = 0);
    void frente(float temp = 0);
    void tras(float temp = 0);
    void esq(float temp = 0);
    void dir(float temp = 0);
    void stop(float temp = 0);

    void setVelocidade(int tempV1, int tempV2,
                      int tempV3, int tempV4);
    float getV(int OP);
private:
    int DLCMotor;
    float V1, V2, V3, V4;
};

```

Figura 1 – Trecho de código do arquivo “GuaraTeca.h” (interface)

```

GuaraTeca.cpp*  GuaraTeca.h*
Miscellaneous Files  MRF

void MRobot::frente(float temp){
    motorDefaultOfMRobot1.run(FORWARD);
    motorDefaultOfMRobot2.run(FORWARD);
    if(DLCMotor){
        motorDefaultOfMRobot3.run(FORWARD);
        motorDefaultOfMRobot4.run(FORWARD);
    }
    delay(temp);
}

void MRobot::tras(float temp){
    motorDefaultOfMRobot1.run(BACKWARD);
    motorDefaultOfMRobot2.run(BACKWARD);
    if(DLCMotor){
        motorDefaultOfMRobot3.run(BACKWARD);
        motorDefaultOfMRobot4.run(BACKWARD);
    }
    delay(temp);
}

void MRobot::esq(float temp){
    motorDefaultOfMRobot1.run(FORWARD);
    motorDefaultOfMRobot2.run(BACKWARD);
    if(DLCMotor){
        motorDefaultOfMRobot3.run(FORWARD);
        motorDefaultOfMRobot4.run(BACKWARD);
    }
    delay(temp);
}

```

Figura 2 – Trecho de código do arquivo “GuaraTeca.cpp”

Na figura 1 é possível ver a classe “MRobot”, a qual faz o elo de ligação entre uma placa de expansão *motorshield* e o usuário, a chamada interface. Já a figura 2, demonstra parte do código-fonte para a manipulação de um robô utilizando Arduino com *motorshield*. É possível visualizar nesta figura as funções (métodos) para um robô andar para frente, direita, esquerda e para trás, códigos esses que seriam escritos em cada robô caso o programador não utilize a “GuaráTeca”.

Após a implementação de todas as classes e métodos, partiu-se para os testes no laboratório de robótica com três robôs construídos para finalidades distintas. A figura 3 mostra um código-fonte que não utiliza a biblioteca e, na figura 4, código-fonte utilizando a biblioteca *GuaraTeca.h* descrita no escopo do programa, escrito por alunos para um robô que desvia de obstáculos (*explorer*), o robô “BMO”.

```

float valorDistancia;
int pino1A = 2;
int pino1B = 3;
int pino1V = 9;
int pino2A = 4;
int pino2B = 5;
int pino2V = 10;
int trigPin = 13;
int echoPin = 12;
void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(pino1A, OUTPUT);
  pinMode(pino1B, OUTPUT);
  pinMode(pino2A, OUTPUT);
  pinMode(pino2B, OUTPUT);
  pinMode(pino1V, OUTPUT);
  pinMode(pino2V, OUTPUT);
  analogWrite(pino1V, 255);
  analogWrite(pino2V, 255);
}
void loop() {
  distancia();
  if(valorDistancia < 12){
    digitalWrite(pino1A, HIGH);
    digitalWrite(pino1B, LOW);
    digitalWrite(pino2A, HIGH);
    digitalWrite(pino2B, LOW);
    delay(1000);
    digitalWrite(pino1A, LOW);
    digitalWrite(pino1B, LOW);
    digitalWrite(pino2A, LOW);
  }
}
void distancia(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long microsec = pulseIn(echoPin, HIGH);
  valorDistancia = (microsec / 27.6233 / 2.0);
}
  
```

```
#include<GuaraTeca.h>

HRobot robo(2, 3, 9, 4, 5, 10, 100);

Sensor Ultrasonico(13, 12);

void setup() {
  Serial.begin(9600);
}

void loop() {
  if(Ultrasonico.distancia() 12){
    robo.frente(1000);
  }else{
    robo.tras(1000);
  }
}
```

Figura 4 – código-fonte do robô “BMO” com a utilização da “GuaráTeca”

É possível verificar na figura 4 a grande economia de código em relação ao código-fonte visto na figura 3, além de um melhor entendimento e simplicidade após a utilização da “GuaráTeca”.

Foram escritos ainda códigos-fontes e testados nos robôs seguidor de linha (“Bigodin”), e de resgate (“Bigodon d’Cabron”), com e sem o uso da “GuaráTeca”. Em todos os casos foram feitos dez testes com a utilização da solução aqui proposta, onde, em todos eles, a biblioteca se mostrou eficiente, de mais fácil entendimento para os alunos e com uma grande economia de código.

Após a utilização desta solução pelo grupo, foi aplicado um questionário de opinião sobre a experiência da utilização da mesma. Oito dos dez alunos do grupo GuaráBots responderam um questionário de opinião com perguntas fechadas e abertas sobre a experiência na utilização.

2. MATERIAL E MÉTODOS

Quanto aos materiais utilizados para desenvolvimento da solução aqui proposta, foi utilizada a linguagem de programação C++ através da ferramenta *Visual Studio Code*, versão 1.2.0 (Code, 2016) e o Arduino IDE, versão 1.6.9 (Arduino, 2016) para efetuar os testes.

Após o desenvolvimento da biblioteca com funções para controlar os motores do tipo DC e servos (com *motorshield* e/ou ponte H), sensores de refletância do tipo infravermelho, sensor ultrassônico e sensor giroscópio, partiu-se para a construção dos robôs de teste.

Já em se tratando de hardware para os testes da solução aqui sugerida, foi utilizada a plataforma Arduino UNO R3, a qual controlou e foi responsável pela integração entre o computador e componentes conectados a ela.

Já os chassis (estruturas) utilizados para os testes foram: *Zumo Pololu* (Robotics & Electronics, 2016) para o robô “Bigodin”, equipado com Arduino UNO R3, uma placa de expansão *motorshield* para controlar os motores, dois micromotores do tipo DC 100:1, um sensor ultrassônico, um sensor giroscópio, dois sensores de refletância, uma lâmpada led, uma *protoboard*¹ e fios diversos, conforme figura 5.

¹ Conforme (Eletrônica Didática, 2016) uma *protoboard*, também conhecida como matriz de contatos, é utilizada para fazer montagens provisórias, teste de projetos, além de inúmeras outras aplicações. É constituída por uma base plástica, contendo inúmeros orifícios

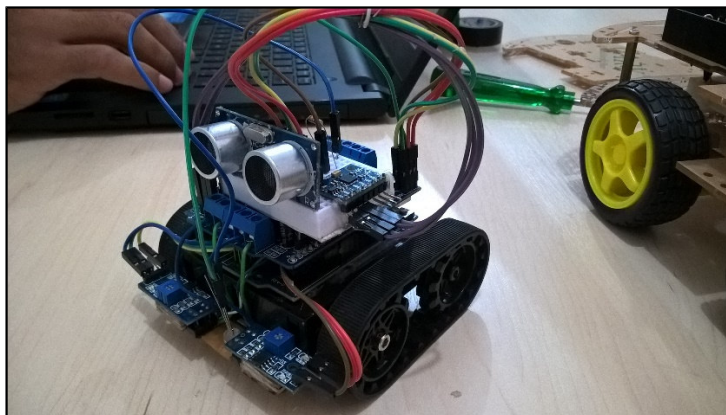


Figura 5 – robô “Bigodin”

O robô “BMO” foi equipado sobre um chassi 4x4, com um Arduino UNO R3, uma ponte H para controle dos motores, quatro motores DC 48:1, dois sensores de refletância, uma case para pilhas, uma *protoboard* e fios para ligação dos componentes, conforme figura 6.

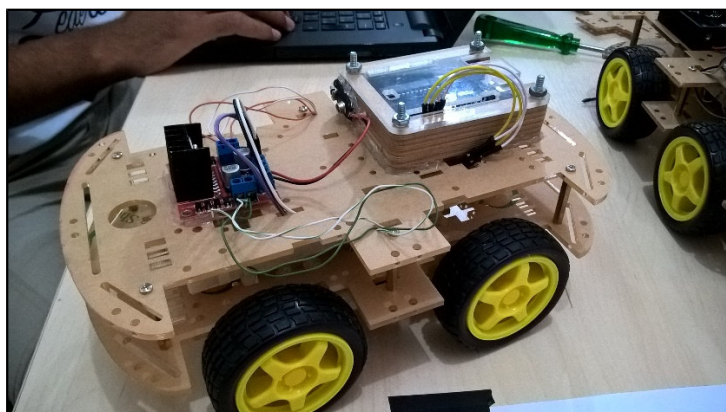


Figura 6 – robô “BMO”

Já o robô “Bigodon d’Cabron” foi construído sobre um chassi 4x4, sendo utilizado um Arduino UNO R3, uma placa de expansão *motorshield*, quatro motores DC 48:1, dois servos motores (garra e ultrassônico), uma garra, um sensor ultrassônico, dois sensores de cor, uma case para pilhas, fios para conexões dos componentes e uma *protoboard*, conforme figura 7.

destinados à inserção de terminais de componentes eletrônicos. Internamente existem ligações determinadas que interconectam os orifícios, permitindo a montagem de circuitos eletrônicos sem a utilização de solda.

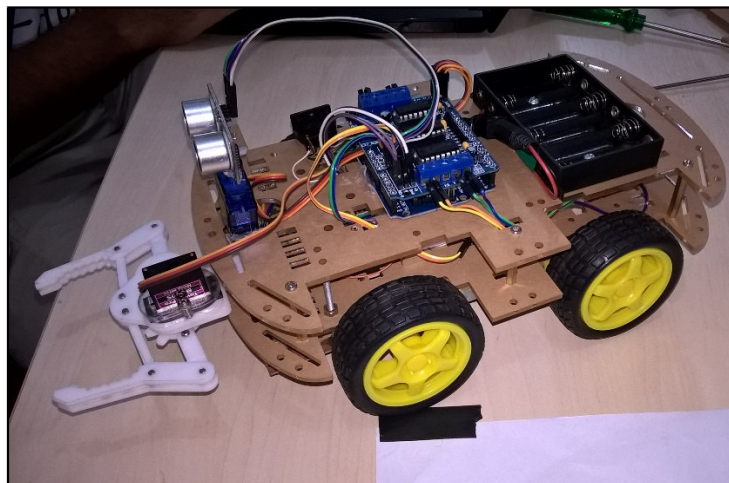


Figura 7 – robô “Bigodon d’Cabron”

Já em relação ao método utilizado, esta pesquisa teve uma abordagem predominantemente qualitativa na análise das respostas dos alunos aos formulários, por possibilitar múltiplas interpretações do que escreveram, porém, dados quantitativos foram utilizados por meio de porcentagem em gráficos para enfatizar algumas informações cruciais nas análises.

Para Kauark, Manhães e Medeiros (2010, p. 26), uma pesquisa qualitativa considera que há uma relação entre o mundo real e o sujeito que não pode ser traduzido em números, ou seja, os pesquisadores tendem a analisar seus dados indutivamente, onde o processo e seu significado são os principais focos de abordagem.

As figuras inseridas neste trabalho pertencem aos arquivos dos autores, construídos durante a pesquisa.

Como instrumento de coleta de dados foi necessária uma pesquisa bibliográfica para descrever conceitos e teorias sobre o Arduino e linguagens de programação utilizadas para programá-los, além de uma pesquisa de opinião sobre a experiência da utilização da “GuaráTeca”. Oito dos dez alunos do grupo GuaráBots, os quais são estudantes do curso Técnico em Informática Integrado ao Ensino Médio do campus Dianópolis do IFTO, responderam um questionário com perguntas fechadas e abertas sobre a experiência na utilização da solução ora proposta. As respostas dos questionários foram organizadas em uma planilha para facilitar a análise dos dados.

3. RESULTADOS E DISCUSSÃO

A partir do questionário aplicado ao grupo de estudos em programação e robótica, GuaráBots, foi possível identificar a opinião dos usuários em relação à “GuaráTeca” em ambiente prático no laboratório de robótica.

Quanto ao nível de conhecimento em programação para Arduino, sete dos oito entrevistados (87,5%) afirmaram estar no nível iniciante. Já quanto ao uso e dificuldades na utilização da biblioteca, seis alunos (62,5%) responderam que já utilizaram a biblioteca e não tiveram nenhuma dificuldade.

Já em relação aos fatores que mais facilitaram a compreensão, estes afirmaram que a simplicidade (66,7%) e os termos (funções) escritos em português (33,3%) são os maiores diferenciais para esta ferramenta. Ainda quanto à utilização, 100% dos alunos que utilizaram a biblioteca afirmaram que, de fato, é possível reduzir o tempo gasto com programação em robôs de competição, e não só em competições de resgate, conforme afirmam 6 dos entrevistados (75%).

Vale ressaltar ainda que todos os entrevistados responderam desconhecer qualquer outra biblioteca disponível no mercado que tenha as mesmas (ou melhor) funcionalidades e objetivos da “GuaráTeca”.

E, de modo geral, 50% dos alunos que a utilizaram a avaliaram como “excelente”, 25% como “bom” e os 25% restantes ainda não utilizaram a biblioteca.

Ressalta-se neste momento que os 25% (dois alunos) que ainda não utilizaram a “GuaráTeca” se dá ao fato dos mesmos estarem empenhados em atividades com outras plataformas robóticas no grupo de estudos.

Acredita-se que a avaliação “bom” pode ser dada ao fato da ferramenta estar em construção, com alguns termos ainda em inglês, e com algumas funções ainda a ser implementadas.

Quanto aos testes práticos, foram desenvolvidos códigos-fontes em três plataformas robóticas distintas, com e sem a utilização da “GuaráTeca”. A tabela a seguir mostra se houve economia de código-fonte ao se usar a biblioteca nos robôs preparados para competições com finalidades diversas.

Tabela 1 – Teste de eficiência da “GuaráTeca”

Robô	Finalidade	Economia de Código-Fonte
Bigodon d’Cabron	Resgate	(de 71 para 23 linhas)
Bigodin	Seguidor de Linha	(de 59 para 34 linhas)
BMO	Explorer	(de 51 para 21 linhas)

A tabela 1 mostra a economia de código-fonte em robôs com três finalidades distintas, se mostrando bastante eficiente, com programação mais simples e intuitiva, principalmente para alunos em nível iniciante em programação e robótica, reduzindo-se o tempo gasto com a programação, além de facilitar o acesso na manipulação de motores, sensores, placas de expansão, luzes e outros componentes.

4. CONCLUSÕES

Esta pesquisa teve como objetivo geral desenvolver uma biblioteca de funções (métodos) a fim de facilitar o trabalho de alunos participantes de competições de robótica que utilizam Arduino. E como objetivos específicos demonstrar a utilização e eficiência da referida biblioteca em diferentes formatos de robôs para competição (resgate, explorador e seguidor de linha), além de analisar a experiência dos alunos em relação ao uso desta biblioteca. E assim, o desenvolvimento da “GuaráTeca” foi a solução desenvolvida para ajudar os membros do grupo GuaráBots na programação desses robôs.

Algumas dificuldades foram encontradas durante o desenvolvimento deste trabalho, como o entendimento do paradigma de orientação a objetos, assim como o uso de bibliotecas externas para o uso da solução ora proposta.

Após o desenvolvimento e os testes em ambiente prático no laboratório de robótica, a “GuaráTeca” se mostrou eficiente e de fácil utilização, reduzindo o tempo para programação dos mesmos robôs que não estavam utilizando a referida biblioteca. Quanto à experiência dos alunos que utilizam a solução proposta nesta pesquisa, os questionários mostram a boa aceitação da mesma por parte dos alunos.

Como projetos futuros recomenda-se uma melhor padronização dos códigos-fontes, além de oferecer suporte a componentes não abarcados nesta versão, e a independência de bibliotecas externas para controle de motores.

Destaca-se ainda que o presente trabalho, através de sua metodologia, proporcionou um aprendizado ainda maior em relação à programação, além de ser uma potencial solução para todos os grupos de robóticas que utilizam Arduino em robôs de competição.

REFERÊNCIAS

ADAFRUIT. **Adafruit**. Disponível em <<http://www.adafruit.com>> Acesso em 05 de julho de 2016.

ARDUINO. **Arduino**. Disponível em <<https://www.arduino.cc>> Acesso em 12 de abril de 2016.

ARISTÓTELES. **Metafísica, Livro I cap. I**. São Paulo: Abril Cultural, 1984.

CODE, V. S. **Visual Studio Code**. Disponível em <<http://code.visualstudio.com>> Acesso em 12 de maio de 2016.

DEITEL, H. M., & DEITEL, P. J. **C++ Como Programar**. 5a. ed. Porto Alegre: Bookman, 2016.

ELETRÔNICA DIDÁTICA. **Eletrônica Didática: um jeito fácil de aprender eletrônica**. Disponível em <<http://www.eletronicadidatica.com.br>> Acesso em 05 de agosto de 2016.

KAUARK, Fabiana; MANHÃES, Fernanda Castro; MEDEIROS, Carlos Henrique. **Metodologia da Pesquisa: guia prático**. Itabuna: Via Letterarum, 2010.

MCRBERTS, M. **Arduino Básico**. 2a Ed. São Paulo: Novatec Editora, 2015

OBR. **Regras e Instruções – Provas Regionais/Estaduais Modalidade Prática / 2015**, 1–42, 2015.

OBR. **Olimpíada Brasileira de Robótica**. Disponível em <<http://www.obr.org.br>> Acesso em 06 de julho de 2016.

ROBOTICS & ELETRONICS, P. **Pololu**. Disponível em <<http://www.pololu.com>> Acesso em 12 de maio de 2016.

SCHILDT, Herbert. **C Completo e Total**. 3ª Ed. São Paulo: Makron Books, 1997