



ISSN: 1989-4155

## PLATAFORMA ARDUINO PARA O ENSINO DE LÓGICA DE PROGRAMAÇÃO

Gilmar Ferreira de Aquino Filho<sup>1</sup>

Luiz Henrique Amaral<sup>2</sup>

Juliano Schimiguel<sup>3</sup>

### Resumo

O presente trabalho tem como objetivo demonstrar uma nova prática pedagógica que está sendo utilizado em vários cursos de tecnologia, o intuito de modo geral trate-se de uma atividade facilitadora de aprendizado e motivadora para o estudo de Lógica de Programação em especial linguagem C, isso a partir da utilização da plataforma Arduino como ferramenta de apoio didático.

**Palavras-chave:** Ensino – Tecnologia – Linguagem de Programação – Arduino – Didática.

### Abstract

This paper aims to demonstrate a new pedagogical practice that is being used in various technology courses, the general order of addresses is a facilitator of learning and motivating activity for the study of Logic Programming especially C language, this from using the Arduino platform as a teaching support tool.

**Keywords:** Education - Technology - Programming Language - Arduino - Teaching

---

<sup>1</sup>Professor da Faculdade de Tecnologia de São Vicente - FATEF e Mestrando Bolsista CAPES em Ensino de Ciências pela Universidade Cruzeiro do Sul. e-mail: g.aquinofilho@gmail.com

<sup>2</sup>Professor Dr. da Universidade Cruzeiro do Sul - UNICSUL email: luiz.amaral@cruzeirosul.edu.br

<sup>3</sup>Professor Dr. da Universidade Cruzeiro do Sul - UNICSUL email: juliano.schimiguel@cruzeirosul.edu.br

<sup>1</sup> Professor da Faculdade de Tecnologia de São Vicente - FATEF e Mestrando Bolsista em Ensino de Ciências pela Universidade Cruzeiro do Sul. e-mail: g.aquinofilho@gmail.com

<sup>2</sup>Professor do IFGoiano e Mestrando em Ensino de Ciências e Matemática pela Universidade Cruzeiro do Sul. e-mail: jonatas.ifgoiano@gmail.com

<sup>3</sup>Professor Dr. da Universidade Cruzeiro do Sul - UNICSUL email: luiz.amaral@cruzeirosul.edu.br

<sup>4</sup>Professor Dr. da Universidade Cruzeiro do Sul - UNICSUL email: juliano.schimiguel@cruzeirosul.edu.br

## INTRODUÇÃO

A lógica de programação é uma disciplina que está presente em grande parte da grade curricular de cursos técnicos e de graduação voltados para a área de Informática/Tecnologia, bem como nas engenharias, através disso vê-se a necessidade de fazer com que ela tenha o seu conceito teórico aplicado de forma prática em níveis que vão além de *softwares* criados para computador.

Nos últimos três séculos, a tecnologia foi desenvolvida de tal maneira que é inconcebível uma sociedade globalizada sem a presença de artefatos que visam o conforto do ser humano, desenvolvidos através do avanço da tecnologia. Seja na forma de Celulares/*Smartphones*, Computadores, Eletrodomésticos ou até mesmo meios de transporte, ela é intrínseca ao cotidiano da sociedade e faz parte do desenvolvimento da mesma. Porém, nem todos dominam a produção da tecnologia e aqueles que o conseguem e destacam no mercado de trabalho. Neste contexto, conhecer as linguagens de programação é uma ferramenta muito útil para aqueles que desejam produzir e inovar em tecnologia.

A principal dificuldade no aprendizado da programação é o fato de programas de computadores serem objetos virtuais, exigindo que o estudante dessa área detenha uma alta capacidade de abstração para entender o que está sendo proposto, como no exemplo abaixo:

```
/* Exemplo de código Linguagem C para computador */  
  
// Mostrando mensagem na tela  
  
void main()  
{  
  
printf("Olá Mundo!!!\n"); // mostra mensagem  
  
system("PAUSE"); // para a execução da tela  
  
}
```

Assim, a utilização de um *kit* de componentes eletrônicos, juntamente com microcontroladores mostra-se de grande ajuda na compreensão de conceitos básicos de programação, pois permite ao aluno observar fisicamente aquilo que foi programado, gerando um resultado palpável.

Exemplo com o código com Arduino:

```
/* Exemplo de código para Arduino */  
// Piscando Led  
  
void setup()  
{  
  // Configura porta do led para escrita / saída  
  pinMode(13, OUTPUT);  
}  
void loop()  
{  
  digitalWrite(ledPin, HIGH); // liga o LED  
  delay(1000); // espera um segundo  
  digitalWrite(ledPin, LOW); // desliga o LED  
  delay(1000); // espera um segundo  
}
```

Nota-se nesses códigos uma diferença em relação ao seu tamanho, porém os dois apresentam comandos de saída, no que é feito em linguagem C para computadores irá imprimir na tela o resultado ao contrário da plataforma Arduino que piscará o LED no intervalo de 1 (um) segundo.

Porém o código da plataforma Arduino é mais completo em virtude de ter que preencher o número da porta em que o LED (Diodo Emissor de Luz) irá acender, definir se a saída será feita em uma porta digital ou analógica e delimitar tempo para a saída. Isso ajudaria o aluno a pensar nos componentes eletrônicos e não somente em um programa de computador.

O fluxo de execução de qualquer programa do Arduino começa na função *setup()*, em que definimos quais pinos iremos usar e se eles serão de entrada ou de saída. Para isso, usamos a função *pinMode(pino, modo)*, onde *pino* é um inteiro representando o número do pino que estamos configurando e *modo* é o tipo de pino que ele será, no caso entrada ou saída - *Input* ou *Output*, respectivamente. Por exemplo, *pinMode(13, OUTPUT)* indica que o pino 13 será usada como uma saída; ou seja, ele fornecerá sinal ao circuito.

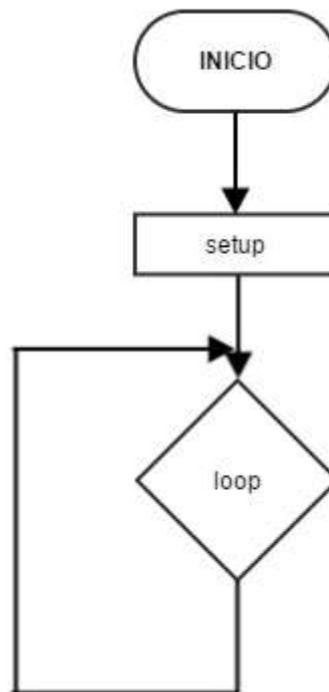


Figura 1 - Fluxograma

Após a execução da função *setup()*, o programa passa a executar a função *loop()*, e continuará a repeti-la até a execução do programa terminar. Destarte, ela é uma das funções mais importantes que iremos utilizar na construção de projetos, pois é nela que iremos descrever tudo que o circuito irá fazer e como ele vai se comportar.

Este artigo trata de uma nova abordagem no ensino da programação: a utilização da plataforma Arduino para o ensino de lógica de programação como ferramenta de assimilação das estruturas condicionais.

## DESENVOLVIMENTO

A presente seção fornece uma visão geral do Arduino, principal ferramenta utilizada na execução dos projetos, e descreve a elaboração e o desenvolvimento da oficina.

Arduino é uma plataforma de fonte aberta destinada a prototipação de sistemas digitais interativos. Ela é capaz tanto de receber e enviar dados de diversos tipos de sensores e a diversos atuadores, viabilizando a construção rápida de sistemas de controle em diversos domínios de aplicação.

Segundo Vasiljčić, Gabriel sobre Arduino (2013, p. 3)

É bastante flexível e não requer um domínio profundo de eletrônica, o que o fez ser bastante popular entre os artistas e iniciantes, além de desenvolvedores experientes que não têm acesso a plataformas mais complexas.

Dentre os vários modelos de Arduino, para a demonstração foi escolhido o modelo UNO (Figura 2), que utiliza o microcontrolador ATmega328. Este modelo possui seis entradas analógicas e catorze entradas/saídas digitais, das quais seis podem ser utilizadas como saídas PWM. Pode ser alimentada por fontes, baterias ou até mesmo pela conexão USB via computador.

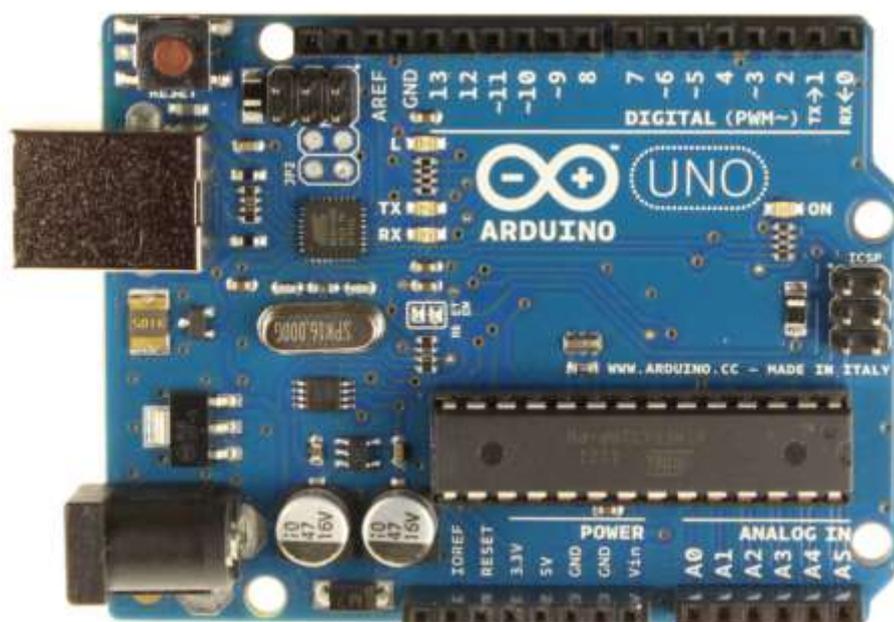


Figura 2 - Arduino UNO

A linguagem de programação utilizada em todos os *kits* Arduino é a "C", utilizando a plataforma *open-source* fornecido pelo site <http://arduino.cc>, sendo que a plataforma foi desenvolvida em Java, possibilitando ao aluno entrar em contato com fundamentos de linguagens largamente empregadas no mercado de trabalho. Para desenvolver os programas, compilar arquivos e fazer o *upload* para o microcontrolador, utiliza-se a IDE *Arduino development environment* (Figura 3, 4), que é multiplataforma funcionando em Windows, MAC OS e Linux.



Figura 3 - IDE do Arduino

The image is a screenshot of the Arduino IDE interface. At the top, there is a menu bar with the following items: "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". Below the menu bar is a toolbar with several icons: a checkmark, a right-pointing arrow, a document with a plus sign, an upload icon, a download icon, and a speech bubble icon. The main area of the IDE is a text editor showing a sketch named "sketch\_jan23a". The code in the sketch is as follows:

```
void setup() {  
  // Escreva o código que será executado apenas uma vez.  
  pinMode(13, OUTPUT); // DECLARANDO QUE O PINO 13 É SAÍDA.  
}  
  
void loop() { // INICIO DO LOOP  
  // Escreva o código que será executado infinitas vezes  
  digitalWrite(13, HIGH); //HIGH = 1 : LIGA LED  
  delay(500); // TEMPO DE ESPERA  
  digitalWrite(13, LOW); //LOW = 0 : DESLIGA LED  
  delay(500); // TEMPO DE ESPERA  
} // FIM DO LOOP
```

At the bottom of the IDE, there is a status bar with the number "13" on the left and "Arduino Uno on COM5" on the right.

Figura 4 - IDE do Arduino

Segundo Vasiljčić, Gabriel sobre Arduino (2013, p. 11)

O IDE do Arduino é bastante simples. Foi projetado para ser uma interface amigável à pessoas que nunca tiveram contato com desenvolvimento de software, e portanto é bastante intuitiva. Foi desenvolvido em Java e possui recursos simples de realce de palavras-chave e uma base com diversos códigos prontos para servir como exemplo.

Para o auxílio na montagem do circuito poderá ser utilizado o Fritzing, Fritzing é um programa em ambiente gráfico que facilita o aprendizado e as primeiras montagens com Arduino, ele mostra de modo virtual como seria uma montagem física em protoboard, bem como o circuito elétrico.

É fácil de instalar e usar, vem com interface em português e possui mais de 45 exemplos de projetos que podem ser facilmente visualizados, montados ou até mesmo modificados.

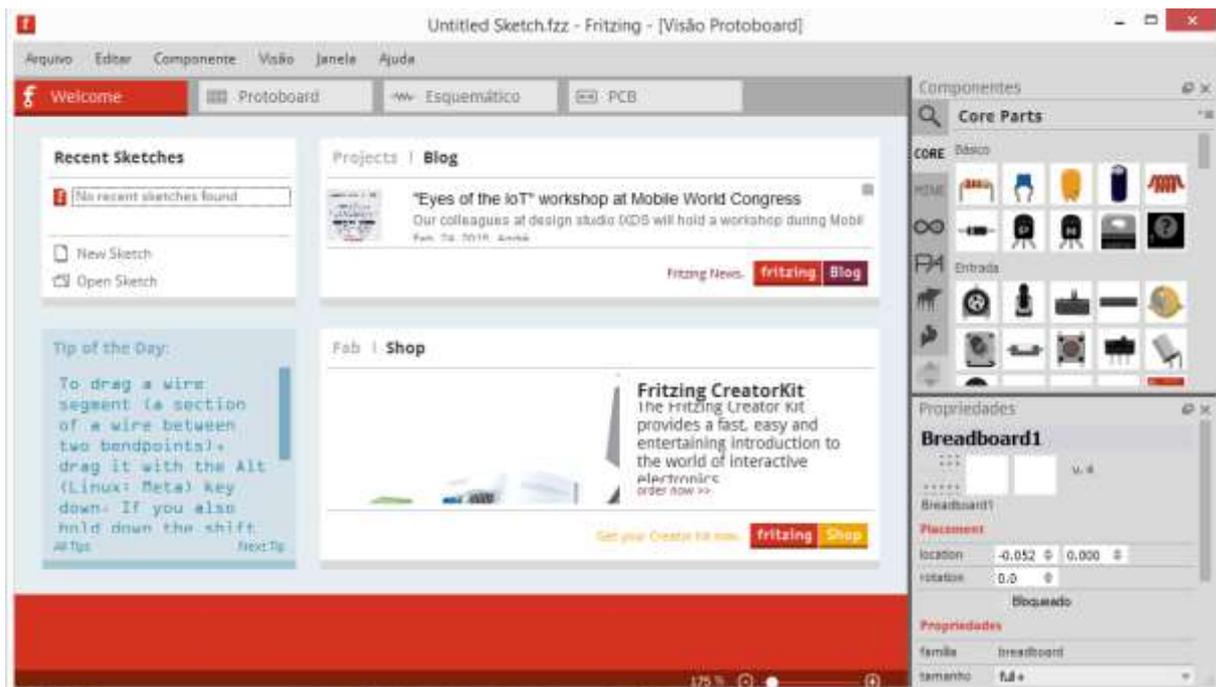


Figura 5 - Ambiente Fritzing

## **CONSIDERAÇÕES FINAIS**

O Arduino tem proporcionado ao ensino meios de facilitar o conhecimento de lógica de programação, eletrônica e robótica, além de ser um *hardware* de custo baixo em comparação a outras tecnologias e tem o diferencial que proporciona uma maior facilidade assemelhando-se com as linguagens utilizadas no mercado de trabalho.

Com perspectiva para o próximo semestre uma pesquisa com objetivo de consolidar a prática de ensino com a plataforma Arduino.

## **REFERÊNCIAS**

AQUINO FILHO, Gilmar, Curso Técnico em Mecatrônica, 2015. Apostila do curso oferecida pela FORTEC, São Vicente, 2015.

VASILJECIĆ. Gabriel, Apostila de Arduino, 2013. Apostila oferecida pela UFRN, Natal, 2013.

ARDUINO. Arduino, 2005. Disponível em: <<http://arduino.cc>>. Acesso em: 18/07/2015.

FRITZING. Fritzing, 2011. Disponível em: <<http://fritzing.org>>. Acesso em: 18/07/2015.