

# ROBÓTICA EDUCATIVA: GUIA PRÁTICO DE/PARA PROFESSOR

*EDUCATIONAL ROBOTICS: PRACTICAL GUIDE TO/FROM TEACHER*

Kelling Cabral Souto<sup>I</sup> 

Helena Carla Castro<sup>II</sup> 

Cristina Maria Carvalho Delou<sup>III</sup> 

<sup>I</sup> Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro, IFRJ, Rio de Janeiro, RJ, Brasil. Doutora em Engenharia Nuclear. Docente do IFRJ. E-mail: [kelling.souto@ifrj.edu.br](mailto:kelling.souto@ifrj.edu.br)

<sup>II</sup> Universidade Federal Fluminense, UFF, Rio de Janeiro, RJ, Brasil. Doutora em Biologia e Modelagem Molecular. Docente do UFF. E-mail: [hcastrorangel@yahoo.com.br](mailto:hcastrorangel@yahoo.com.br)

<sup>III</sup> Universidade Federal Fluminense, UFF, Rio de Janeiro, RJ, Brasil. Doutora em Educação. Docente do PPG Ciências, Tecnologias e Inclusão. E-mail: [cristinadelou@id.uff.br](mailto:cristinadelou@id.uff.br)

**Resumo:** Com o advento da Educação 4.0 que se baseia no conceito do aprender fazendo, novas diretrizes educacionais vêm sendo encorajadas, assim como modificações no processo ensino-aprendizagem e na relação professor-aluno. Nesse contexto, o aluno passa a ser protagonista do seu próprio aprendizado e o professor atua como mediador de circunstâncias e ações que levem a aprendizagem do aluno, com a tecnologia naturalmente envolvida nessas etapas. As metodologias ativas, em especial o STEAM, e a robótica educativa surgem como fortes aliados nesse processo, aproximando estudantes das tendências tecnológicas do cotidiano e da filosofia do aprender fazendo. Sensíveis ao discurso de que a escola precisa se adaptar as novas demandas educacionais as autoras, leigas em robótica e em eletrônica, iniciam, por meios próprios, um trabalho com robótica educativa na perspectiva da Educação 4.0, com placa Arduino UNO. Ao longo de seis meses, uma oficina foi implementada com informações obtidas na *internet* e aplicada a dois grupos de alunos em níveis extremos (pré-escola, estudantes com Altas Habilidades ou Superdotação e graduação, licenciandos em matemática), com as devidas adaptações. A motivação dos alunos e as dificuldades vivenciadas pelas autoras professoras na aquisição de conhecimento mínimo para a implantação da oficina estimularam a elaboração de um guia básico e prático de robótica educativa. Por meio de uma linguagem simples, sem preocupação com o rigor técnico, esse guia destina-se a capacitação de professores, como ponto de partida para estimular a implantação de atividades de ciência e tecnologia nas escolas brasileiras, bem como a filosofia da Educação 4.0.

**Palavras-chave:** Educação 4.0. Tecnologia. Robótica educativa. Arduino.

DOI: <https://doi.org/10.31512/vivencias.v18i37.692>

Submissão: 29-10-2021

Aceite: 09-12-2021



Esta obra está licenciada com uma Licença Creative Commons Atribuição-NãoComercial-SemDerivações 4.0 Internacional.

**Abstract:** With the advent of Education 4.0, which is based on the concept of learning by doing, new educational guidelines have been encouraged, as well as changes in the teaching-learning process and in the teacher-student relationship. In this context, the student becomes the protagonist of their own learning and the teacher acts as a mediator of circumstances and actions that lead to student learning, with technology naturally involved in these stages. Active methodologies, especially STEAM, and educational robotics emerge as strong allies in this process, bringing students closer to everyday technological trends and the philosophy of learning by doing. Sensitive to the discourse that the school needs to adapt to new educational demands, the authors, lay people in robotics and electronics, started, by their own means, a work with educational robotics from the perspective of Education 4.0, with an Arduino UNO board. Over a six-month period, a workshop was implemented with information obtained from the internet and applied to two groups of students at extreme levels (preschool, students with High Skills or Giftedness and graduation, undergraduates in mathematics), with the necessary adaptations. The motivation of the students and the difficulties experienced by the author-teachers in the acquisition of minimal knowledge for the implementation of the workshop stimulated the development of a basic and practical guide to educational robotics. Using simple language, without concern for technical rigor, this guide is intended to train teachers, as a starting point to encourage the implementation of science and technology activities in Brazilian schools, as well as the philosophy of Education 4.0.

**Keywords:** Education 4.0. Technology. Educational robotics. Arduino.

## Introdução

Nos dias atuais diante das inovações tecnológicas são observadas transformações em diversos setores da sociedade (HUSSIN, 2018), fato que tem afetado diretamente o desenvolvimento humano, fazendo surgir a Educação 4.0 como resposta ao advento da Revolução Industrial 4.0<sup>1</sup>.

A Educação 4.0 que se baseia no conceito de *Learning by doing*, vem provocando novas diretrizes educacionais e forçando naturalmente modificações no processo ensino-aprendizagem (HUSSIN, 2018; ANDRADE, 2018) e na relação professor-aluno (PANAGIOTOPoulos e KARANIKOLA, 2020).

---

<sup>1</sup>Revolução Industrial 4.0 nasceu a partir da evolução da tecnologia e a necessidade de integração de processos em várias verticais da indústria com o objetivo de levar eficiência, agilidade e mais qualidade de entrega para os clientes finais (FOCCOERP, 2017).

Fisk (2017) explica que na nova visão trazida pela Educação 4.0, o aprendizado é construído em torno do aluno, onde se aprende não apenas habilidades e conhecimentos que são necessários, mas a fonte dessas habilidades e conhecimentos. Fisk (2017) afirma ainda que no processo de aprendizagem os pares se tornam muito significativos. Os alunos aprendem juntos uns com os outros, enquanto os professores assumem o papel de facilitadores no aprendizado desses.

A Educação 4.0 coloca o aluno como protagonista do seu próprio aprendizado e traz a tecnologia naturalmente inserida nesse processo. Andrade (2018) assegura que colaborar, criar, pesquisar e compartilhar são ações que necessitam fazer parte do processo ensino e aprendizagem atual. O aluno, com orientação do professor, vai se tornando autônomo em sua aprendizagem, com capacidade de continuar esse processo ao longo de sua vida. Ainda de acordo com Andrade (2018) na educação 4.0 as salas de aulas se transformam em espaços de desenvolvimento de competências colaborativas.

Na Educação 4.0 o papel do professor (PANAGIOTOPOLOS e KARANIKOLA, 2020), também, sofre transformação e ele deverá criar circunstâncias propícias às exigências de um novo ambiente educacional, assim como propor e mediar ações que levem à aprendizagem do aluno. Garofalo (2018) mostra claramente o papel do professor nesse contexto quando afirma que o mesmo deve ser aprendiz, mediador, orientador e pesquisador de forma a conseguir propor e gerenciar ações buscando a aprendizagem do aluno.

Aliado a essa filosofia surge à metodologia de pesquisa ativa que tem como principal objetivo a aprendizagem ativa, onde a responsabilidade de aprender é colocada nas mãos dos próprios alunos. A referida metodologia designa, ainda, o papel de facilitador para o professor. A tarefa do facilitador é permitir que os alunos aprendam sozinhos através do uso de diferentes atividades de treinamento ativo (MORAN, 2017; ANDRADE JUNIOR, SOUZA e SILVA, 2019; NASCIMENTO e CASTRO NETO, 2020).

Dentro das metodologias ativas destaca-se o STEAM – *Science, Technology, Engineering, Arts and Mathematics* – conceito que surge com base na tecnologia e na mudança em relação ao aprendizado dos estudantes. Um dos seus objetivos é formar indivíduos que possuam conhecimentos em diversas áreas de Exatas e de Humanas, além de prepará-los para se tornarem cidadãos criativos, autônomos, responsáveis, críticos, com habilidade para trabalhar em equipe e tomar decisões (SPONTE, 2018; MORAN, 2017).

Os países desenvolvidos e instituições renomadas estão investindo alto nessa nova demanda cientes que os avanços tecnológicos precisam ser acompanhados pelas escolas. Nesse cenário, observam-se as metodologias ativas (MORAN, 2017; ANDRADE JUNIOR, SOUZA e SILVA, 2019; NASCIMENTO e CASTRO NETO, 2020) e a robótica educativa (GOMES, SILVA, BOTELHO e SOUZA, 2010; FABRÍCIO, 2014; CAMPOS, 2017) como forte aliados nesse processo, aproximando estudantes das tendências tecnológicas do cotidiano e da filosofia do aprender fazendo.

Fabricio (2014) define robótica como a ciência dos sistemas que interagem com o mundo real com ou sem intervenção dos humanos, afirmando que a robótica na educação está em

expansão e sua utilização é multidisciplinar, permitindo a professores e alunos inter-relacionarem conteúdos curriculares de maneira harmônica.

Zilli (2004) defende que a robótica educacional pode desenvolver as seguintes competências: raciocínio lógico, habilidades manuais e estéticas, relações interpessoais e intrapessoais, integração de conceitos aprendidos em diversas áreas do conhecimento para o desenvolvimento de projetos, investigação e compreensão, representação e comunicação, trabalho com pesquisa, resolução de problemas por meio de erros e acertos, aplicação das teorias formuladas a atividades concretas, utilização da criatividade em diferentes situações e capacidade crítica.

No Brasil, já se observam esforços tímidos no sentido de inserção da Educação 4.0 por meio da robótica educativa nas escolas, entretanto, não representa regra geral principalmente quando se fala em escolas públicas. Silva (2014) aponta quatro motivos para tal fato, a saber: falta de professores capacitados para dar início e continuidade a esse tipo de atividade nas escolas; baixo interesse dos gestores (especialmente públicos) em implementar projetos que envolvam ciência e tecnologia; o pouco engajamento e divulgação com as comunidades locais o que acaba não disseminando os ganhos adquiridos por esse tipo de atividade; e o custo para aquisição de *kits* e material didático para as aulas de robótica educativa. Esse último motivo é um fator que atrapalha, mas não é determinante uma vez que já se consegue no mercado material de baixo custo e até mesmo o uso de sucatas.

Docentes há algumas décadas, as autoras desse artigo são testemunhas das transformações educacionais e mudanças de interesses do alunado nativo digital no que diz respeito ao conteúdo estudado e a forma de aprendizado. Sensíveis ao discurso de que a escola precisa se adaptar as novas demandas educacionais iniciam, por meios próprios, a aquisição de equipamento mínimo e de conhecimento para um trabalho com robótica educativa na perspectiva da Educação 4.0, na forma de oficina e nos moldes do que já vem ocorrendo em outros países. Optou-se por placa Arduino UNO (ARDUINO, 2020) devido ao baixo custo e a grande quantidade de informação disponível na *internet* e de domínio público.

É importante destacar que as autoras ao iniciarem o trabalho não possuíam nenhum conhecimento prévio de robótica educativa e muito menos de eletrônica. A *internet* apresenta informações de variados níveis de complexidade que possibilitam o trabalho com Arduino (ARDUINO, 2020), componentes eletrônicos e programação, necessários para projetos de robótica, entretanto, observa-se que as informações em grande parte encontram-se desconexas, em inglês e com o uso de linguagem técnica, o que dificulta o aprendizado por parte de professores leigos no assunto.

Tendo vivenciado dificuldades para seleção de informações e cientes da necessidade de capacitar professores para dar início a atividades de ciência e tecnologia nas escolas brasileiras, esse artigo traz um guia básico elaborado por autoras docentes e destinado a professores que queiram se aventurar na robótica educativa junto a seus alunos.

O guia reúne conhecimentos essenciais adquiridos na *internet* ao longo de seis meses de pesquisas, o que possibilitou a montagem de uma oficina de robótica educativa com Arduino

(ARDUINO, 2020) para alunos de diversos níveis de escolaridade, com devidas adaptações. Tal oficina foi aplicada a dois grupos de estudantes em níveis extremos, crianças com Altas Habilidades ou Superdotação (BRASIL, 2001 e 2013; RENZULLI, 2004 e 2014) da pré-escola e alunos de licenciatura em matemática, explorando conteúdos e habilidades diferenciadas conforme interesse e faixa etária.

A escolha por dois níveis tão diversificados de escolaridade se deu para avaliar a versatilidade de tal prática pedagógica, desde discentes prestes a iniciar a vida escolar até futuros professores de matemática. A aplicação às crianças com Altas Habilidades ou Superdotação (BRASIL, 2001 e 2013; RENZULLI, 2004 e 2014) promoveu um enriquecimento escolar (RENZULLI, 2004 e 2014) fundamental a esses estudantes. Já a aplicação junto aos licenciandos de matemática ocorreu como um processo de formação docente.

O nível de envolvimento despertado por ambos os grupos durante aplicação da oficina foi outra razão que motivou a elaboração do guia de robótica educativa voltado a professores que desejem utilizar essa prática educacional em suas aulas. O trabalho reúne informações de fontes diversas, referencia bons trabalhos e procura transcrever em linguagem simples, sem a preocupação com o rigor técnico, informações básicas sobre robótica educativa, com Arduino, que possibilitem professores leigos aplicar e aprender junto com seus alunos, na perspectiva da Educação 4.0, promovendo a educação por meio da tecnologia.

### **Metodologia: guia de robótica educativa – ambiente e componentes**

O Arduino é uma plataforma eletrônica de código aberto baseada em *hardware* e *software* fáceis de usar. As placas Arduino são capazes de ler entradas e transformá-la em uma saída, como ativando um motor ou acendendo um *LED* por exemplo. O processamento de entrada em saída é possível através do envio de um conjunto de instruções a placa Arduino, via *software* Arduino – IDE (*software* que permite escrever programas e carregá-lo na placa Arduino) (ARDUINO, 2020).

O Arduino (ARDUINO, 2020) nasceu como uma ferramenta fácil, de prototipagem rápida, voltada para alunos sem formação em eletrônica e programação. Assim que alcançou uma comunidade mais ampla, a placa Arduino começou a mudar para se adaptar às novas necessidades e desafios, oferecendo placas diversas para fins diversos. Todas as placas e *software* Arduino são de código aberto, permitindo que os usuários as criem de forma independente e, eventualmente, as adaptem às suas necessidades particulares.

Os programas escritos no ambiente de desenvolvimento do Arduino IDE estão em linguagem baseada no C/C++, usando uma estrutura simples e o fato dos códigos serem abertos permite a professores, mesmo sem conhecimento em programação, entender e adaptar de forma a atender suas necessidades. Esse guia irá tratar da placa Arduino UNO (ARDUINO, 2020).

### *Processo de instalação*

Para usar o Arduino é preciso configurar o *software* Arduino IDE para controlar sua placa. Existem duas opções de trabalho: *online*, quando se tem uma conexão confiável à *internet*, o que não é a realidade da maioria das escolas brasileiras e *offline*, foco desse guia, do qual é sugerido instalar a versão mais recente (ARDUINO, 2020).

O Arduino IDE está disponível para ser instalado a alguns tipos de sistemas operacionais, mas será apresentado apenas a versão *Windows* para *Personal Computer* (PC), por ser mais comumente encontrado nas escolas brasileiras.

Para instalação faça o *download* em ARDUINO (2020), do *software* Arduino IDE para computadores *Windows*. Use o instalador (.EXE), prossiga permitindo o processo de instalação do *driver* quando receber um aviso do sistema operacional. Em seguida, escolha todos os componentes e o diretório padrão para instalação. Com esse procedimento serão instalados todos os arquivos necessários para executar corretamente o *software* Arduino IDE. Posteriormente, é possível fazer a adição de bibliotecas disponíveis que podem ser instaladas e utilizadas facilmente (apresentado no tópico a seguir), estendendo a capacidade do seu Arduino.

### *Ambiente de desenvolvimento*

Uma vez instalado o *software* Arduino IDE antes de iniciar a programar é preciso verificar se a IDE está devidamente configurada para a placa e para a entrada USB a ser utilizada (a placa apresenta uma USB, através da qual se pode conectá-la a um computador).

A Figura 1 apresenta o ambiente de desenvolvimento Arduino IDE. Para configurar a placa e a entrada USB basta nesse ambiente clicar em Ferramentas > Borda (*Tolls > Board*) e certificar-se que a opção Arduino Uno esteja selecionada. Depois clicar em Ferramentas > Porta Serial (*Tools > Serial Port*) e certificar-se que a porta correta está selecionada, conforme seu computador, no *Windows* será algo como “COM3 Arduino Uno” (GER, 2016).

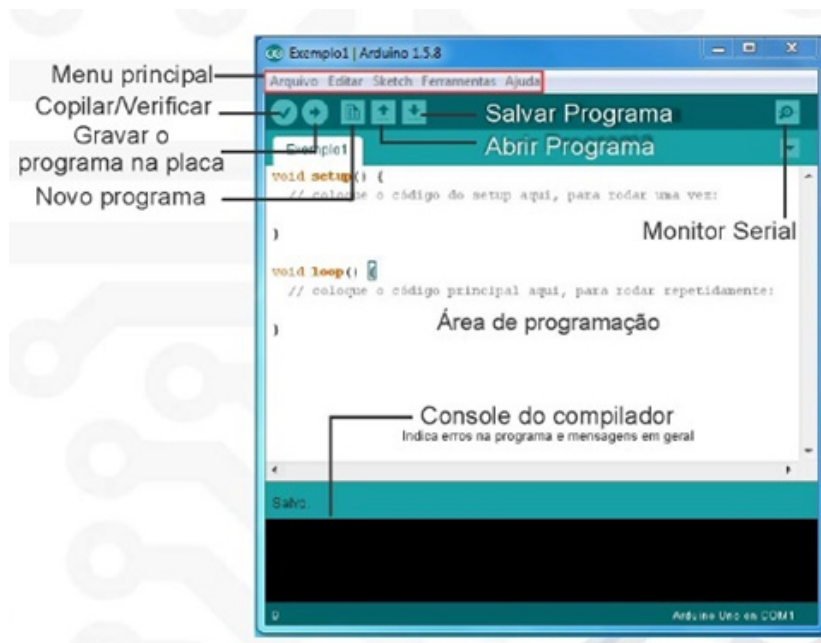
Uma das facilidades propostas pelo Arduino é uma IDE simples e fácil de usar. Ela possui vários exemplos prontos de código, um monitor serial (Figura 1) e uma ferramenta que simplifica a adição de novas bibliotecas (GER, 2016).

O monitor serial é o elo entre o computador e o seu Arduino. Com ele é possível enviar e receber informações na forma de texto, útil para depuração e também para controlar o Arduino pelo teclado do PC (CIRCUITAR, 2018).

Para acessar o monitor serial corretamente, você primeiramente deve conectar a placa ao computador, fazer *upload* do seu código e então clicar no símbolo da lupa (Figura 1) que se encontra no canto superior direito na tela do Arduino e aparece sempre que um código estiver aberto no ambiente de desenvolvimento (CIRCUITAR, 2018).

Basicamente, todo código Arduino deve possuir duas funções (Figura 1): *void setup* (), função chamada apenas uma vez, no início da execução do código. Ela deve ser usada para definir qualquer componente ou variável do código; e a função *void loop* () que deve ser implementada após a função *void setup* (). O código irá chamar esta função *void loop* () *ad infinitum*.

Figura 1 - Arduino IDE



Fonte: Mota (2018).

Os códigos prontos (exemplos de projetos) que o *software* Arduino IDE traz podem ser acessados no ambiente de desenvolvimento, clicando em Arquivo > Exemplos (Figura 1). A sugestão é iniciar por esses códigos prontos para se familiarizar com o ambiente, a execução e a linguagem de programação em si.

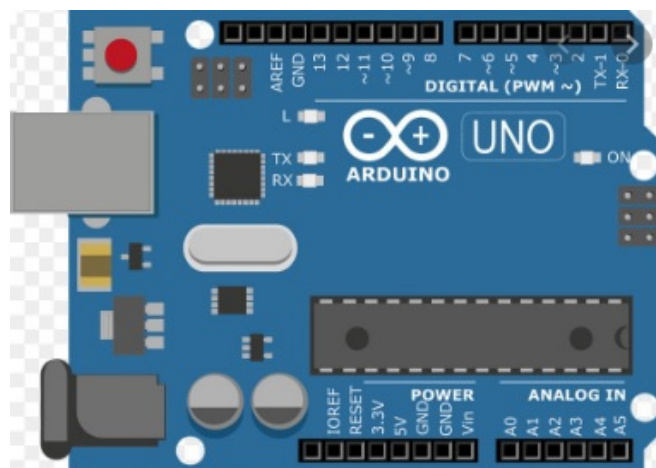
Silva (2014) traz os principais elementos da linguagem de programação Arduino, ou seja, os principais comandos e sintaxes que podem ser úteis a quem está aprendendo.

Quanto às bibliotecas, que permitem estender a capacidade do Arduino, são adicionadas no *software* Arduino IDE, por meios diversos. A forma considerada mais simples é a instalação por dentro do próprio Arduino IDE, para tal basta fazer *download* dos arquivos da biblioteca compactada que deseja adicionar e dentro do ambiente de desenvolvimento clicar em Esboço > Biblioteca de importação > Adicionar biblioteca (*Sketch > Import Library > Add Library*), selecionar a biblioteca desejada e utilizá-la em seu programa (MOTA, 2018).

### *Placa Arduino UNO*

Existem várias placas Arduino, mas a Arduino UNO é uma versão bastante popular e por representar um *hardware* simples, de baixo custo e eficiente será utilizado como referência nesse trabalho (ARDUINO, 2020). A Figura 2 ilustra uma placa Arduino UNO.

Figura 2 - Placa Arduino UNO



Fonte: Arduino (2020).

Para o uso da placa Arduino é importante se conhecer alguns aspectos fundamentais (GER, 2016; MOTA, 2018):

- a. Pinos A0 até A5 são conectores de entrada que recebem sinais analógicos. Pinos analógicos trabalham com tensões que variam de 0 à 5 volts;
- b. Pinos 0 até 13 são conectores de entrada e saída para sinais digitais. Pinos digitais trabalham com o valor de 0 volt (desligado) ou 5 volts (ligado);
- c. GND são pinos com função de aterramento, com tensão 0 volt;
- d. A placa apresenta uma interface USB, através da qual é possível conectá-la a um computador, possibilitando sua programação, alimentação e troca de informações;
- e. A tensão de alimentação do próprio Arduino é 5 volts, na maioria dos modelos de placas. Um computador já estabelece automaticamente uma tensão segura para a placa a partir da USB, ou seja, o computador como a própria fonte de energia da placa é na maior parte das aplicações satisfatório. Tensões maiores podem danificar a placa e o computador em que ela está conectada;
- f. 5 volts (e outros, como 3.3 volts) são conectores de tensão, utilizados para levar alimentação aos módulos conectados à placa;
- g. *RESET* reinicializa a placa quando energizado;
- h. Os códigos escritos no ambiente de desenvolvimento do Arduino, no geral, farão menção aos pinos analógicos ou digitais a serem conectados na placa.

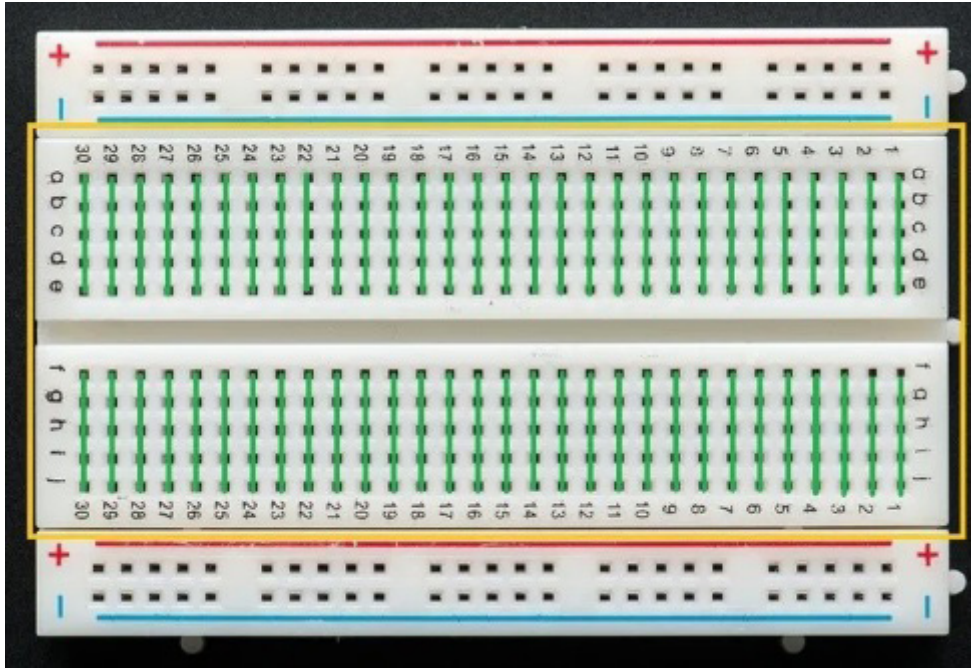
É sempre válido lembrar que grandezas digitais são aquelas que não variam continuamente no tempo, mas sim em saltos entre valores bem definidos. Um exemplo são os relógios digitais que apesar do tempo em si variar continuamente, o visor do relógio mostra o tempo em saltos de um em um segundo. Um relógio desse tipo nunca mostrará 12,5 segundos, pois para ele, só existem 12 e 13 segundos. Qualquer valor intermediário não está definido. Já as grandezas analógicas permitem valores intermediários (MOTA, 2018).



## Protoboard

É uma placa de ensaio muito útil na prototipagem de circuitos. Fisicamente a *protoboard* apresenta vários furos para fazer conexões entre componentes de um circuito elétrico. Ela é espelhada e tem duas partes: a central (barramentos verticais), para conexões entre os componentes elétricos, e a extremidade (barramentos horizontais), geralmente usada para conectar a alimentação e a terra. A Figura 3 ilustra uma *protoboard* e mostra suas faixas de barramentos.

Figura 3 - *Protoboard*



Fonte: Vida de silício (2021).


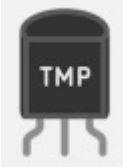
Os barramentos horizontais destinados a pinos de alimentação do circuito (pinos de VCC e GND) têm seus furos interligados horizontalmente. Os demais, representados dentro de um retângulo amarelo na Figura 3, destinam-se aos componentes eletrônicos e tem seus furos interligados verticalmente. Entender essa questão é fundamental para prototipagem dos circuitos (ROBOCORE, 2020).

## *Componentes eletrônicos*

A Tabela 1 apresenta alguns componentes eletrônicos mais comumente utilizados em projetos Arduinos básicos.

**Tabela 1** - Alguns componentes eletrônicos básicos (nome, função e ilustração)

Componente eletrônico e função	Ilustração
Cabos <i>jumper's</i> macho x macho: fundamentais para desviar, ligar ou desligar o fluxo elétrico, fazendo com que sejam realizadas as configurações específicas do seu projeto.	
Resistor: controlar a passagem de corrente elétrica em circuitos elétricos. Quando utilizado diminui a intensidade da corrente elétrica.	
LED: é um diodo emissor de luz que tem a propriedade de transformar energia elétrica em luz.	
Botão: estabelecer ou interromper a passagem de corrente elétrica em linhas de um circuito de comando.	
Potenciômetro: componente eletrônico que possui resistência elétrica ajustável. Geralmente, é um resistor de três terminais onde a conexão central é deslizante e manipulável, permitindo limitar o fluxo de corrente elétrica.	
Capacitor: armazena cargas elétricas em seu interior. Durante as descargas podem fornecer carga elétrica para um circuito.	
Interruptor deslizante: úteis para ligar e desligar circuitos elétricos.	
Motor servo: é um atuador eletromecânico utilizado para posicionar e manter um objeto em uma determinada posição, a partir de um sinal de controle.	

<p>Sensor de distância: função de medição de distância sem contato com o objeto.</p>	
<p>Sensor de temperatura: detectar a variação de temperatura no meio.</p>	

Fonte imagens: Tinkercad (2021).

Convencionou-se que cabos *jumper's* vermelhos e pretos são utilizados nas ligações positivas (+) e negativas (-), respectivamente, mas nada impede o uso de outras cores ou mesmo essas em outras ligações.

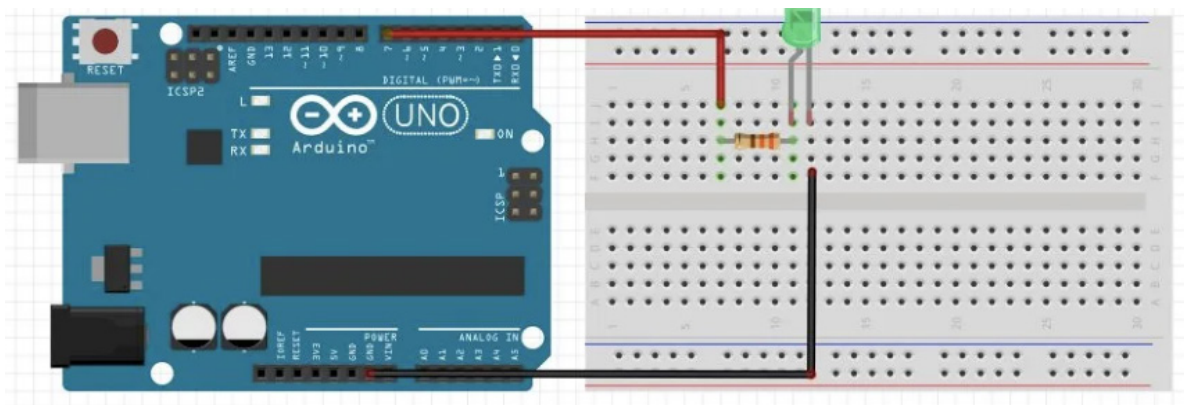
Potenciômetros são resistores variáveis e ajustáveis, e por isso são usados para controle analógico da funcionalidade de alguns aparelhos eletrônicos, tal como o volume de um aparelho de som (MOTA, 2018). Assim como outros componentes descritos acima, eles costumam possuir três pinos, os das extremidades fazem o papel de positivo e negativo e o pino central é o que se liga a placa Arduino.

## Resultados e discussões: Guia de robótica educativa – projetos práticos

Na sequência serão apresentados três projetos simples e completos baseados em exemplos do próprio Arduino (ARDUINO, 2020) com pequenas adaptações.

O primeiro projeto foi denominado piscar. Seu objetivo é fazer piscar um *led* a cada 1 segundo, repetidamente. O material necessário é: computador, placa Arduino, *protoboard*, cabos *jumper's* macho x macho (preto e vermelho), resistor e *LED*. O esquema de montagem ilustrado na Figura 4 conta com cabos *jumper's* vermelho e preto (ligando o positivo e negativo, respectivamente) e o código implementado encontra-se na Figura 5.

Figura 4 - Esquema de montagem projeto piscar



Fonte: Arduino (2020).

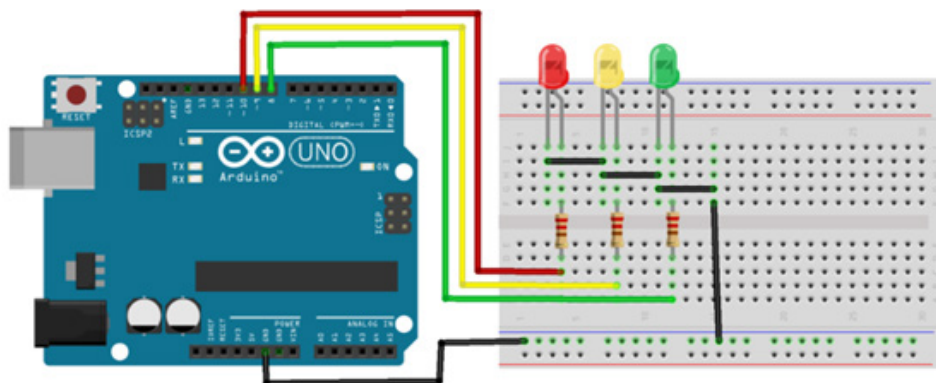
Figura 5 - Código do projeto piscar

```
void setup() { // a função setup é executada quando liga a placa ou preciona o reset.  
  pinMode(LED_BUILTIN, OUTPUT); // inicializa o pino digital LED_BUILTIN como uma saída.  
}  
  
void loop() { // a função loop é repetida indefinidamente.  
  digitalWrite(LED_BUILTIN, HIGH); // acende o led (nível de tensão alta).  
  delay(1000); // aguarde por 1 segundo. O led se manterá acesso por 1 segundo.  
  digitalWrite(LED_BUILTIN, LOW); // desliga o led (nível de tensão baixa).  
  delay(1000); // aguarde por 1 segundo. O led se manterá apagado por 1 segundo.  
}
```

Fonte: Arduino (2020), com adaptações.

O segundo projeto foi denominado semáforo *delay*, cujo objetivo é reproduzir um semáforo com tempos distintos para cada cor de lâmpada. Material necessário: computador, placa Arduino, *protoboard*, cabos *jumper's* macho x macho (vermelho, preto, amarelo, verde), *led* (vermelho, amarelo, verde) e 3 resistores. O esquema de montagem ilustrado na Figura 6 conta com cabos *jumper's* vermelho, amarelo e verde (ligando o positivo de cada *led* à placa Arduino, portas 8, 9 e 10); 3 cabos *jumper's* pretos (ligando o negativo de cada *led*, dentro da *protoboard*); e 1 cabo *jumper* preto (ligando o negativo da *protoboard* a placa Arduino). O código implementado é apresentado integralmente nas Figuras 7 e 8.

Figura 6 - Esquema de montagem projeto semáforo delay



Fonte: Arduino (2020).

Figura 7 - Código do projeto semáforo delay – parte 1

```
#include <avr/sleep.h> //Biblioteca adicionada.
// Constantes usadas para definir os LED's aos pinos no arduino.
const int ledPm1 = 13; // LED1 (vermelho) estará ligado ao pino 8 no arduino
const int ledPm2 = 12; // LED2 (amarelo) estará ligado ao pino 9 no arduino
const int ledPm3 = 11; // LED3 (verde) estará ligado ao pino 10 no arduino
int ledState1 = LOW; // variável que define o status inicial do LED1 como desligado
int ledState2 = LOW; // variável que define o status inicial do LED2 como desligado
int ledState3 = LOW; // variável que define o status inicial do LED3 como desligado
int contador = 1; // contador para controlar LEDs
// "unsigned long" é indicado para variáveis que controlam tempo
unsigned long previousMillis = 0; // irá guardar a ultima vez que o LED foi atualizado
const long interval = 200; //constante que indica o intervalo para piscar em milisegundos
void setup() { // a função setup é executada quando liga a placa ou preciona o reset.
// define o pino digital como saída:
pinMode(ledPm1, OUTPUT);
pinMode(ledPm2, OUTPUT);
pinMode(ledPm3, OUTPUT);
}
void loop() { // a função loop é repetida indefinidamente.
unsigned long currentMillis = millis();
previousMillis = currentMillis;
if (contador == 1) { // se contador for igual a 1, acende o LED3 (verde) por 3 segundos (delay (3000) e mantém os demais apagados.
ledState1 = LOW;
ledState2 = LOW;
ledState3 = HIGH;

```

Fonte: Arduino (2020), com adaptações.

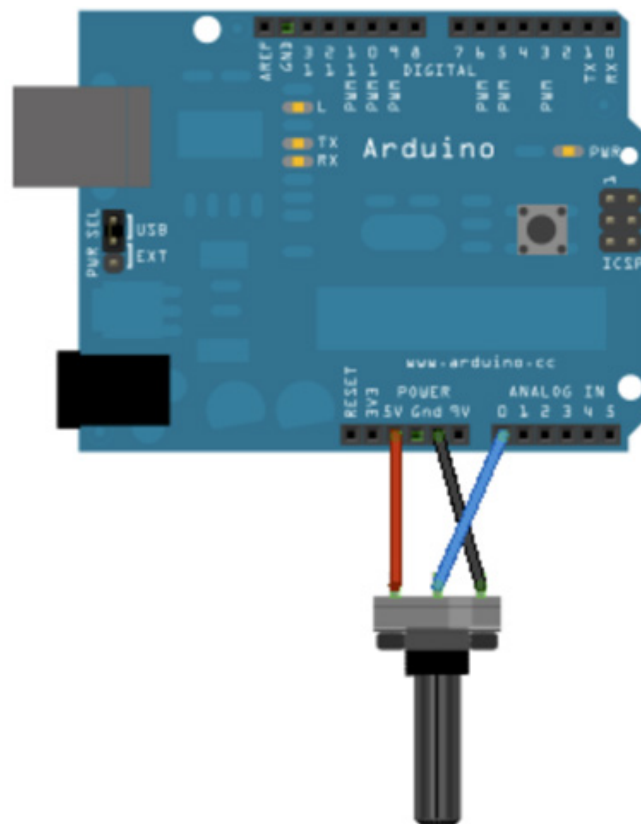
Figura 8 - Código do projeto semáforo delay – parte 2 (continuação)

```
contador = 2;
digitalWrite(ledPm1, ledState1);
digitalWrite(ledPm2, ledState2);
digitalWrite(ledPm3, ledState3);
delay(3000); }
else {
if (contador == 2) { //se contador = 2, acende o LED2 (amarelo) por delay(800) e mantém os demais apagados.
ledState1 = LOW;
ledState2 = HIGH;
ledState3 = LOW;
contador = 3;
digitalWrite(ledPm1, ledState1);
digitalWrite(ledPm2, ledState2);
digitalWrite(ledPm3, ledState3);
delay(800); }
else { // se contador = 3, acende o LED1 (vermelho) por delay(5000) e mantém os demais apagados.
ledState1 = HIGH;
ledState2 = LOW;
ledState3 = LOW;
contador = 1;
digitalWrite(ledPm1, ledState1);
digitalWrite(ledPm2, ledState2);
digitalWrite(ledPm3, ledState3);
delay(5000);
} } }
```

Fonte: Arduino (2020), com adaptações.

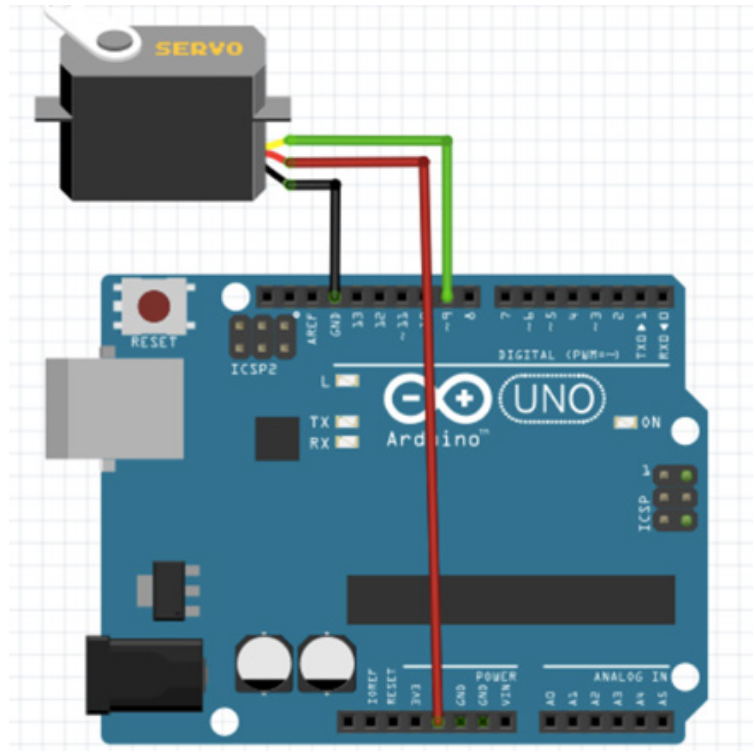
O terceiro projeto foi denominado braço servo motor com potenciômetro. Seu objetivo é fazer mover braço mecânico por meio de um motor servo comandado por um potenciômetro. O material necessário é: computador, placa Arduino, motor servo, braço, potenciômetro, cabo *jumper's* macho x macho (vermelho, preto e azul). Seu esquema de montagem (Figuras 9 e 10) conta com cabos *jumper's* vermelho e preto (ligando positivo e negativo do motor a placa Arduino); cabos *jumper's* vermelho e preto (ligando positivo e negativo da *proto-board* ao motor); cabos *jumper's* vermelho e preto (ligando positivo e negativo da *proto-board* ao potenciômetro); cabo *jumper* azul (ligando o pino central do potenciômetro ao pino A0 da placa Arduino); e cabo verde (ligando o motor ao pino 9 da placa Arduino). Seu código encontra-se na Figura 11.

Figura 9 - Esquema de montagem projeto braço servo motor com potenciômetro - Potenciômetro



Fonte: Arduino (2020).

Figura 10 - Esquema de montagem projeto braço servo motor com potenciômetro - Motor



Fonte: Arduino (2020).

Figura 11 - Código do projeto braço servo motor com potenciômetro

```
#include <Servo.h> // chama a biblioteca para o motor servo
Servo myservo; // cria um objeto servo para controlar um motor servo
int potpin = 0; // pino analógico conectado ao potenciômetro
int val; // variável que guarda o valor do pino analógico conectado ao potenciômetro (potpin).
void setup() { // a função setup é executada quando liga a placa ou preciona o reset.
  myservo.attach(9); // relaciona o motor servo ligado no pino 9 do arduino ao objeto servo
}
void loop() { // a função loop é repetida indefinidamente.
  val = analogRead(potpin); // lê o valor do potenciômetro (valor entre 0 e 1023)
  val = map(val, 0, 1023, 0, 180); // escala o valor do potenciômetro para o servo (valor ente 0 e 180)
  myservo.write(val); // seta a posição do servo conforme valor escalonado.
  delay(15); // aguarda o servo ir para a posição
}
```

Fonte: Arduino (2020), com adaptações.

Os três projetos práticos acima foram testados e aplicados durante oficina de robótica educativa. No trabalho junto a alunos com Altas Habilidades ou Superdotação (BRASIL, 2001 e 2013; RENZULLI, 2004 e 2014) da pré-escola, a ênfase principal ficou no processo de

montagem eletrônica, tratada de forma concreta e lúdica. Junto aos licenciandos em matemática deu-se ênfase a montagem eletrônica e a programação, trabalhando conceitos de forma abstrata.

A total aceitação dos participantes a essa prática pedagógica por meio da robótica educativa, metodologias ativas e Educação 4.0 superou as expectativas. Vale mencionar que os licenciandos em matemática que participaram da oficina de robótica educativa ficaram encantados com a proposta e demonstraram interesse em um guia para auxiliá-los em suas futuras aulas por meio da robótica educativa.

## Considerações finais

Em tempo de inovação tecnológica, a Educação 4.0 vem provocando novas diretrizes educacionais e forçando naturalmente modificações no processo ensino-aprendizagem e na relação professor-aluno.

Em paralelo as metodologias ativas colocam o professor no papel de facilitador do aprendizado, onde o aluno aprende através do uso de diferentes atividades de treinamento ativo.

Cientes que o avanço tecnológico precisa ser acompanhado pelas escolas, a robótica educativa surge como forte aliada, aproximando estudantes das tendências tecnológicas do cotidiano, da filosofia do aprender fazendo da Educação 4.0 e das demandas do mundo atual.

No Brasil já se observam esforços tímidos no sentido de inserção da Educação 4.0 por meio da robótica educativa nas escolas, entretanto, não representa regra geral principalmente quando se fala em escolas públicas. Alguns motivos são apontados para esse fato e entre eles está a falta de professores capacitados para implementar projetos que envolvam ciência e tecnologia.

Sensíveis ao discurso de que a escola precisa se adaptar as novas demandas educacionais, bem como a falta de professores capacitados a implementar projetos de natureza tecnológica, as autoras, leigas em robótica e em eletrônica, iniciam, por meios próprios, um trabalho com robótica educativa na perspectiva da Educação 4.0, com placa Arduino. Ao longo de seis meses, uma oficina foi implementada com informações obtidas na *internet* e aplicada a dois grupos de estudantes (pré-escola e licenciandos em matemática). A total aceitação dos mesmos a prática pedagógica proposta, motiva o presente artigo que desenvolve um guia básico sobre robótica educativa destinado a professores que desejam ingressar nesse universo tecnológico.

Todo conteúdo descrito nesse artigo foi testado e aplicado. O professor seguindo as orientações descritas, os esquemas de montagens, obtendo os poucos materiais necessários para cada atividade proposta e reproduzindo os códigos indicados conseguirá reproduzir os projetos práticos sem maiores dificuldades. Após a familiarização inicial alcançar autonomia na busca de mais conhecimentos.



## Referências

- ANDRADE JUNIOR, J. de M.; SOUZA, L. P. de; SILVA, N. L. C. da. Metodologias ativas: práticas pedagógicas na contemporaneidade. Campo Grande - MS: Editora Inovar, 203p, e-book, 2019.
- ANDRADE, K. **O desafio da Educação 4.0 nas escolas**. 2018. Disponível em: <https://canaltech.com.br/mercado/o-desafio-da-educacao-40-nas-escolas-109734/>. Acesso em: 06 jan. 2020.
- ARDUINO. **Arduino**. 2020. Disponível em: <https://www.arduino.cc>. Acesso em: 01 set. 2020.
- ARRUDA, J. S.; NASCIMENTO, K. A. S. do; CASTRO NETO, D. N. de O. **Metodologias Ativas: pense, mude, planeje e compartilhe**. Fortaleza: Editora Unichristus, 2020.
- BRASIL. Conselho Nacional de Educação (CNE). **Resolução CNE/CEB Nº 02** de 11 de setembro de 2001. Brasília, DF, 2001.
- BRASIL. Lei de Diretrizes e Bases da Educação (LDB) - **Lei Nº 12.176** de 4 de abril de 2013. Brasília, DF, 2013.
- CAMPOS, Flavio Rodrigues. Robótica educacional no Brasil: questões em aberto, desafios e perspectivas futuras. **RIAEE – Revista Ibero-Americana de Estudos em Educação**, v. 12, n. 4, p. 2108-2121, 2017. Disponível em: <http://dx.doi.org/10.21723/riaee.v12.n4.out./dez.2017.8788>. Acesso em: 10 out. 2021.
- CIRCUITAR. **Circuitar Eletrônicos**. 2018. Disponível em: <https://www.circuitar.com.br/projetos/monitor-serial/index.html>. Acesso em: 01 dez. 2020.
- FABRÍCIO, P. R. de A. M.; COSTA NETO, O. E. da e ANDRADE, E. L. de S. Utilização da robótica na educação: Uma realidade no município de Solânea – PB. **Nuevas Ideas en Informática Educativa TISE 2014**, p 857 – 860, 2014.
- FISK, P. **Education 4.0 ... the future of learning will be dramatically different, in school and throughout life**. 2017. Disponível em: <https://www.peterfisk.com/2017/01/future-education-young-everyone-taught-together/>. Acesso em: 06 jan. 2020.
- FOCCOERP. **Indústria 4.0 - Guia completo da indústria do futuro**. 2017. Disponível em: <https://www.foccoerp.com.br/wp-content/uploads/2017/09/ebook-industria-4>. Acesso em: 01 nov. 2020.
- GAROFALO, D. **Que habilidades deve ter o professor da Educação 4.0**, 2018. Disponível em: <https://novaescola.org.br/conteudo/11677/que-habilidades-deve-ter-o-professor-da-educacao-40>. Acesso em: 06 jan. 2018.

GER. Grupo de Estudos em Robótica - Equipe de robótica autônoma da Faculdade de Engenharia Mecânica da Unicamp. **Introdução ao Arduino**, São Paulo, 2016. Disponível em: <http://www.gerunicamp.com.br/wp-content/uploads/2016/08/Apostila-GER.pdf>. Acesso em: 01 dez. 2020.

GOMES, C. G.; SILVA, F. O. da; BOTELHO, J. da C.; SOUZA, A. R. de. A robótica como facilitadora do processo ensino-aprendizagem de Matemática no ensino fundamental. *In*: PIROLA, N. A. (Org.). **Ensino de ciências e matemática, IV**: temas de investigação [online]. São Paulo: Editora UNESP; Cultura Acadêmica, 2010.

HUSSIN, A. A. Education 4.0 made simple: Ideas for teaching. **International Journal of Education & Literacy Studies**. Publicado por Australian International Academic Centre PTY. LTD. v. 6, n. 3, p. 92-98, 2018. Disponível em: <http://dx.doi.org/10.7575/aiac.ijels.v.6n.3p.92>. Acesso em: 10 jul. 2020.

MORAN, J. Metodologias ativas e modelos híbridos na educação. *In*: YAEGASHI, S. *et al.* (Orgs). **Novas Tecnologias Digitais**: Reflexões sobre mediação, aprendizagem e desenvolvimento. Curitiba: CRV, 2017.

MOTA, A. **Apostila Arduino Básico**, v. 1 e 2, 2018. Disponível em: <https://pt.scribd.com/document/347472450/Arduino-Basico-Vol-1> e <http://commitlinux.com.br/wp-content/uploads/2016/06/Apostila-Arduino-Vol-2.pdf>. Acesso em: 01 dez. 2020.

PANAGIOTOPOLOS, G. A.; KARANIKOLA, Z. A.; Education 4.0 and teachers: Challenges, risks and benefits. **European Scientific Journal**, v. 16, n. 34, p. 114-128, 2020.

RENZULLI, J. A concepção de superdotação no modelo dos três anéis: Um modelo de desenvolvimento para a promoção da produtividade criativa. *In*: VIRGOLIM, A. M. R. e KONKIEWITZ, E. C. (Org.). **Altas habilidades/superdotação, inteligência e criatividade**: uma visão multidisciplinar – Campinas, SP: Papyrus, 2014.

RENZULLI, J. O que é esta coisa chamada superdotação, e como a desenvolvemos? Uma retrospectiva de vinte e cinco anos. **Educação**, ano XXVII, n. 1, v. 52, p. 75-131, 2004.

ROBOCORE. **ROBOCORE tecnologia**. Tutoriais Pocket Kit Arduino - 5. A Protoboard. 2020. Disponível em: <https://www.robocore.net/tutoriais/arduino-pocket-kit-a-protoboard>. Acesso em: 01 dez. 2020.

SILVA, A. J. B. da. **Um modelo de baixo custo para aulas de robótica educativa usando a interface Arduino**. 2014. 109 f. Dissertação de Mestrado – Universidade Federal de Alagoas, Programa de Pós-Graduação em Modelagem Computacional do Conhecimento, Alagoas, 2014. Disponível em: <http://www.repositorio.ufal.br/handle/riufal/1608>. Acesso em: 01 dez. 2020.

SPONTE. **Software de Gestão Educacional**. Entenda o que é STEAM e como aplicar esse conceito em sua escola. 2018. Disponível em: <https://site.sponte.com.br/gestao-escolar/entenda-o-que-e-steam-e-como-aplicar-esse-conceito-na-escola/>. Acesso em: 06 jan. 2020.

TINKERCAD. **Tinkercad**. 2021. Disponível em: <https://www.tinkercad.com/>. Acesso em: 10 jun. 2021.

VIDA DE SILÍCIO. **Portal Vida de Silício**. 2021. Disponível em: <https://portal.vidadesilicio.com.br/protoboard/>. Acesso em: 10 jan. 2021.

ZILLI, S. do R. **A robótica educacional no ensino fundamental: perspectivas e práticas**. 2004. 89 f. Dissertação de Mestrado – Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção, Santa Catarina, 2004. Disponível em: <http://repositorio.ufsc.br/xmlui/handle/123456789/86930>. Acesso em: 20 out. 2021.