



SERVIÇO PÚBLICO FEDERAL
UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE MATEMÁTICA

APRENDIZAGEM MÃO NA MASSA: INTEGRANDO A MATEMÁTICA E A ROBÓTICA EDUCACIONAL

Trabalho de Conclusão de Curso

MARIA EDUARDA PINHEIRO SIQUEIRA

Belém - Pará

2023

MARIA EDUARDA PINHEIRO SIQUEIRA

**APRENDIZAGEM MÃO NA MASSA: INTEGRANDO A
MATEMÁTICA E A ROBÓTICA EDUCACIONAL**

Trabalho de Conclusão de Curso, apresentado à Faculdade de Matemática do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará como requisito básico para obtenção do título de Licenciada em Matemática.

Orientador(a): Prof. Me. Edilson dos Passos Neri Júnior

Belém - Pará

2023


MARIA EDUARDA PINHEIRO SIQUEIRA

**APRENDIZAGEM MÃO NA MASSA: INTEGRANDO A
MATEMÁTICA E A ROBÓTICA EDUCACIONAL**


Trabalho de Conclusão de Curso, apresentado à Faculdade de Matemática do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará como requisito básico para obtenção do título de Licenciada em Matemática.

Trabalho aprovado com conceito Excelente. Belém - Pará , 06 de julho de 2023:

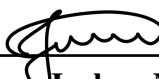
BANCA EXAMINADORA



Prof. Me. Edilson dos Passos Neri Júnior
Orientador



Profa. Dra. Cristina Lúcia Dias Vaz
Avaliadora



Profa. Dra. Joëlma Morbach
Avaliadora

Belém - Pará
2023

Às duas mulheres da minha vida, Clélia e Paula, por sonharem os meus sonhos e por abdicarem de seus planos para que os meus fossem realizados, sempre será por vocês.

Agradecimentos

Em primeiro lugar gostaria de agradecer a Deus, que na palavra de Josué 1:9 me manteve forte e corajosa, fazendo com que eu não desanimasse pois sempre esteve ao meu lado, me dando força e saúde para concluir este trabalho.

Agradeço à minha família que foi minha base e minha fortaleza durante esse tempo, sem eles nada seria possível, em especial minha mãe Paula e minha avó Clélia que foram minhas maiores incentivadoras desde o início da minha vida, me fazendo lembrar o quanto sou capaz e o quanto meus sonhos são possíveis.

Agradeço à Universidade Federal do Pará (UFPA) por ter me acolhido e me proporcionado os quatro anos mais desafiadores da minha vida, pelas oportunidades acadêmicas que eu recebi através dos programas ofertados pela Universidade, pelos profissionais que tive a honra de conhecer e pelos locais de estudos que pude frequentar que foram de suma importância para concluir meu curso.

Agradeço à Faculdade de Matemática (FACMAT) por todo suporte oferecido durante toda a minha graduação, em especial ao Almir por toda a paciência com as inúmeras dúvidas durante os anos em que aqui estive, por sempre se mostrar disponível para ajudar com os problemas possíveis e demonstrar apoio.

Agradeço de forma excepcional ao meu orientador Edilson dos Passos Neri Júnior, por me escolher e enxergar em mim um potencial que eu nunca fui capaz de ver antes. Obrigada pelos quatro anos de trabalho juntos em inúmeros projetos, por todos os conhecimentos repassados e por todo apoio recebido. Obrigada por ter acreditado no meu trabalho, tudo isso só foi possível pois trabalhamos em conjunto.

Gostaria de agradecer também de forma muito especial aos meus amigos de vida e os que ganhei com a graduação, vocês foram essenciais para que eu chegasse até aqui, obrigada pelas conversas, conselhos, palavras de afirmação e apoio, vocês tornaram essa jornada muito mais leve e bonita.

*"Somos aquilo que fazemos repetidamente.
Excelência, portanto, não é um ato mas sim um hábito"
(Aristóteles)*

Resumo

Com o avanço das tecnologias digitais de informação e comunicação (TDIC) e a busca por metodologias de ensino que atendessem as necessidades e dificuldades que surgiam no âmbito escolar, as TDIC se apresentam como uma ferramenta conveniente e eficiente no processo de ensino, sendo inseridas cada vez mais em sala de aula. Em meio a esse cenário, emerge a Robótica Educacional (RE) como uma metodologia de ensino que visa proporcionar um processo de aprendizagem inovador aos alunos. Este trabalho trata-se de uma pesquisa aplicada e exploratória que avalia a potencialidade dessas atividades como forma de promover ambientes de aprendizagem mão na massa onde os alunos são os protagonistas no processo de ensino e aprendem tudo na prática. Para isso, realizou-se uma revisão bibliográfica para definir conceitos importantes como cultura *maker*, metodologia *STEAM* e Pensamento Computacional, além do estudo de ferramentas como o kit de robótica Arduíno e o *software Autodesk Tinkercad* para auxiliar nas atividades propostas. Além desta pesquisa promover a obtenção de resultados sobre a utilização da RE na construção de uma aprendizagem onde os alunos possam “aprender fazendo”, foi desenvolvida uma oficina *maker* onde os alunos serão estimulados a aprender na prática a construir protótipos, aprendendo matemática trabalhando através da interdisciplinaridade.

Palavras-chave: Abordagem *STEAM*. Aprendizagem mão-na-massa. Arduíno. Matemática. Robótica Educacional. *Tinkercad*.

Abstract

With the advancement of digital information and communication technologies (TDIC) and the search for teaching methodologies that met the needs and difficulties that arose within the school, TDIC are presented as a convenient and efficient tool in the process of teaching, being inserted more and more in the classroom. In the midst of this scenario, emerges Educational Robotics (ER) as a teaching methodology that aims to provide a innovative learning process for students. This work is an applied research and exploratory that evaluates the potential of these activities as a way to promote environments of hands-on learning where students are the protagonists in the teaching process and they learn everything in practice. For this, a bibliographic review was carried out to define concepts such as maker culture, STEAM methodology and Computational Thinking, in addition to study of tools such as the Arduino robotics kit and the Autodesk Tinkercad software for assist in proposed activities. In addition to this research promoting the achievement of results on the use of ER in the construction of learning where students can “learn by doing”, a maker workshop was developed where students will be encouraged to learn in practice how to building prototypes, learning mathematics working through interdisciplinarity.

Keywords: STEAM approach. Hands-on learning. Arduino. Mathematics. Educational Robotics. tinkercad.

Lista de Figuras

Figura 1 – Placa Arduíno UNO R3	31
Figura 2 – <i>Software</i> Arduíno IDE	31
Figura 3 – Interface do Ambiente de Desenvolvimento do Arduíno	32
Figura 4 – Sensores e Módulos	33
Figura 5 – Resistores Ôhmicos	34
Figura 6 – Código de Cores	34
Figura 7 – Tabela de corrente e tensão de LEDs	36
Figura 8 – <i>Protoboard</i> com divisão de 4 matrizes	37
Figura 9 – Sentidos de transmissão de uma <i>protoboard</i>	37
Figura 10 – Interface Inicial do <i>Tinkercad</i>	38
Figura 11 – Interface do ambiente de projetos do <i>Tinkercad</i>	39
Figura 12 – Interface de trabalho do <i>Tinkercad</i>	39
Figura 13 – Placa Arduíno e <i>Protoboard</i>	44
Figura 14 – Placa Arduíno, <i>Protoboard</i> e LED	45
Figura 15 – Placa Arduíno, <i>Protoboard</i> , LED e Resistor	45
Figura 16 – Circuito do LED finalizado	45
Figura 17 – Circuito do LED programação em blocos	46
Figura 18 – Circuito do LED programação	46
Figura 19 – LED piscando programação em blocos	47
Figura 20 – LED piscando programação	48
Figura 21 – Circuito físico LED	48
Figura 22 – QR Code Vídeo do LED aceso	49
Figura 23 – QR Code Vídeo do LED piscando	49
Figura 24 – Sensor de Distância Ultrassônico	50
Figura 25 – Sensor HC-SR04, Arduíno e <i>protoboard</i>	51
Figura 26 – Circuito Sensor HC-SR04	51
Figura 27 – Circuito Sensor HC-SR04 programação em texto	52
Figura 28 – Circuito Sensor HC-SR04 finalizado	53
Figura 29 – Circuito Sensor HC-SR04 e Monitor Serial	54
Figura 30 – Circuito físico Sensor de distância	54
Figura 31 – QR Code vídeo do Sensor de distância	55
Figura 32 – Arduíno e <i>protoboard</i>	56
Figura 33 – Arduíno, <i>protoboard</i> e LEDs	56
Figura 34 – Arduíno, <i>protoboard</i> , LEDs e <i>jumpers</i>	57
Figura 35 – Simulação do circuito dos Sinais finalizado	58
Figura 36 – Programação em Blocos do Circuito dos Sinais	58

Figura 37 – Programação na Linguagem C\C++	59
Figura 38 – Programação na Linguagem C\C++	59
Figura 39 – Circuito físico sinal de uma via de mão única	60
Figura 40 – QR Code vídeos do Sina de uma via de mão única	61
Figura 41 – Sensor HC-SR04, Arduíno e <i>protoboard</i>	62
Figura 42 – Sensor HC-SR04, Arduíno, <i>protoboard</i> e LEDs	63
Figura 43 – Sensor HC-SR04, Arduíno, <i>protoboard</i> , LEDs e resistores	63
Figura 44 – Circuito Estacionamento Inteligente	64
Figura 45 – Programação Estacionamento Inteligente	64
Figura 46 – Programação Estacionamento Inteligente	65
Figura 47 – Circuito físico estacionamento inteligente	66
Figura 48 – QR Code vídeo do Estacionamento Inteligente	66
Figura 49 – Arduíno, <i>protoboard</i> , LED, Sensor HC-SR04 e <i>Buzzer Piezo</i>	68
Figura 50 – Arduíno, <i>protoboard</i> , LED, Sensor HC-SR04 e <i>Buzzer Piezo</i> e resistores	68
Figura 51 – Circuito Sensor de Estacionamento finalizado	69
Figura 52 – Programação Sensor de Estacionamento	69
Figura 53 – Programação Sensor de Estacionamento	70
Figura 54 – Circuito físico Sensor de estacionamento	71
Figura 55 – QR Code vídeo do Sensor de Estacionamento	71
Figura 56 – Protótipo Projeto 1	72
Figura 57 – QR Code vídeo do protótipo do Projeto 1	73

Lista de abreviaturas e siglas

BNCC	Base Nacional Comum Curricular
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
EAUFPA	Escola de Aplicação da Universidade Federal do Pará
IDE	Ambiente de Desenvolvimento Integrado
IPEC	Inteligência em Pesquisa e Consultoria
LABINFRA	Programa de Apoio à Infraestrutura de Laboratórios de Ensino de Graduação e da Educação Básica, Técnica e Tecnológica
PC	Pensamento Computacional
RE	Robótica Educacional
STEAM	Ciência, Tecnologia, Engenharia, Arte e Matemática
TDIC	Tecnologias Digitais de Informação e Comunicação
UFPA	Universidade Federal do Pará
UNICEF	Fundo das Nações Unidas para a Infância

Sumário

Introdução	17
1 Princípios Norteadores	23
1.1 A Cultura <i>MAKER</i>	23
1.2 Abordagem STEAM	25
1.3 Robótica Educacional	26
1.4 Pensamento Computacional	27
2 As Ferramentas para a Robótica Educacional	29
2.1 Arduíno	29
2.1.1 O que compõe a placa Arduíno?	30
2.1.2 A Linguagem de Programação e Ambiente de Desenvolvimento	31
2.1.3 Alguns Componentes	32
2.1.4 Protoboard	36
2.2 Tinkercad	38
3 Proposta de Oficina	41
3.1 Competências Trabalhadas	42
3.2 Recursos	42
3.3 Metodologia	43
3.4 Conteúdo Programático	43
3.5 Atividades Propostas	43
3.5.1 Projeto 1: Sinal de uma via de mão única	55
3.5.2 Projeto 2: estacionamento inteligente	61
3.5.3 Projeto 3: sensor de estacionamento	67
3.6 Protótipo Físico	72
Considerações Finais	75
Referências	77
APÊNDICE A A Linguagem de Programação do Arduíno	81
APÊNDICE B Comandos das Atividades da Oficina	85

Introdução

Este trabalho é fruto das atividades desenvolvidas no projeto "Laboratório de Ensino de Matemática", desenvolvido na Escola de Aplicação da Universidade Federal do Pará, cujo financiamento é oriundo do Programa de Apoio à Infraestrutura de Laboratórios de Ensino de Graduação e da Educação Básica, Técnica e Tecnológica – LABINFRA/2021, que busca investir em espaços de desempenho práticos, promovendo experiências, metodologias inovadoras e condições à flexibilização curricular e vivências inter(trans)disciplinares. O projeto tem como objetivo fortalecer as práticas de ensino no Laboratório de Matemática e ampliar as estratégias didático-metodológicas para o ensino de matemática na Escola de Aplicação da UFPA (EAUFPA), estabelecendo um diálogo contínuo entre a educação básica e a licenciatura em Matemática, com foco na formação inicial de professores e na melhoria do ensino e aprendizagem na educação básica.

Ao longo do projeto, desenvolvemos estudos dirigidos, produzimos materiais didáticos para as aulas de matemática, realizamos oficinas, palestras e cursos para os estudantes da educação básica e do ensino superior e outros produtos educacionais. Toda essa produção foi articulada a partir do planejamento dos professores de Matemática da EAUFPA, tendo em vista que o projeto busca dialogar diretamente com as práticas de ensino em sala de aula. Diversos temas foram abordados, tais como, a modelagem 3D, o uso de jogos educacionais, a produção de materiais concretos, a elaboração de simulações em GeoGebra, a resolução de problemas utilizando *softwares* de computação algébrica e, mais recentemente, a robótica educacional, tema central deste Trabalho de Conclusão de Curso. Nesse contexto, desejamos responder a seguinte pergunta: *é possível aliar a robótica educacional à Matemática para obter um processo de ensino e aprendizagem inovador e que incentive a autonomia?*

Este trabalho tem como objetivo apresentar algumas reflexões que surgiram a partir desta questão, apresentando também atividades de robótica educacional, com vista em uma metodologia inovadora, que incentive os alunos a serem os protagonistas na construção do próprio conhecimento e a buscar novos desafios. Uma das grandes dificuldades enfrentadas pela Educação é a desmotivação dos alunos, uma pesquisa realizada em setembro de 2022 pela Inteligência em Pesquisa e Consultoria (IPEC) para o Fundo das Nações Unidas para a Infância UNICEF ¹ revelou que mais de 2 milhões de meninos e meninas de 11 a 19 anos não terminaram o ensino básico e abandonaram a escola no Brasil, tendo como um dos principais fatores motivadores da evasão escolar a dificuldade de aprendizagem. Com base nisso, observa-se a necessidade de uma abordagem de ensino que proporcione ferramentas que auxiliem no processo de aprendizagem e que complementem às metodologias já utilizadas, atendendo às necessidades e dificuldades dos

¹ Link: <<https://www.unicef.org/brazil/comunicados-de-imprensa/dois-milhoes-de-criancas-e-adolescentes-de-11-a-19-anos-nao-estao-frequentando-a-escola-no-brasil>>

alunos, podendo assim contribuir para um desempenho mais satisfatório e eficaz. Destaca-se então a utilização das tecnologias em sala de aula, de maneira responsável e com a metodologia adequada, pois se não utilizada de forma correta, não gera os resultados esperados. É necessário, portanto, que as tecnologias sejam utilizadas de forma crítica, reflexiva e focada no objetivo de aprendizagem que se deseja e nas especificidades dos estudantes.

A inclusão das tecnologias na sala de aula foi muito defendida por [Papert \(1994\)](#), que acreditava no uso das tecnologias e no computador como ferramenta educacional, tudo isso através do construcionismo, teoria fundamentada por ele. Para [Santos, Costa e Alves \(2020\)](#) o construcionismo é visto como uma abordagem metodológica nova, no qual o professor não é o centro de sala de aula e nem o único detentor do conhecimento, sendo assim o aluno assume um papel importante no processo de aprendizagem.

[Santos, Costa e Alves \(2020\)](#) também afirmam que a teoria de Papert abordava a ideia do professor estimular seus alunos a serem os sujeitos responsáveis pela construção do seu próprio conhecimento, ao desenvolver as atividades e desafios propostos, aprendendo, inclusive, com o erro. Além disso, [Papert \(1994\)](#) acreditava que a inserção das tecnologias em sala de aula poderia auxiliar o aluno em muito mais características não previstas pelo professor, além de promover ainda mais autonomia sobre a construção do conhecimento, defendia a necessidade de repensar o processo de ensino com a inserção das tecnologias como ferramenta capaz de promover atividades atrativas, auxiliando no desempenho dos alunos.

O uso das tecnologias aliado à educação foi defendido por inúmeros pesquisadores ao decorrer dos anos, [Soffner \(2013 apud SOFFNER, 2005, p. 4\)](#) via esse recurso como uma opção relevante para potencializar o desenvolvimento e auxiliar na transformação dos alunos, tornando-os mais autônomos. Nesse contexto, o uso das tecnologias em sala pode acrescentar características aos recursos tradicionais, fazendo com que as aulas não sejam somente expositivas, gerando interesse nos alunos para que busquem querer aprender mais e não somente para obterem boas notas. E cabe ao professor planejar e adequar esses recursos às suas metodologias de ensino, [Moran \(2009\)](#) afirma:

Cada docente pode encontrar sua forma mais adequada de integrar as várias tecnologias e os muitos procedimentos metodológicos. Mas também é importante que amplie que aprenda a dominar as formas de comunicação interpessoal/grupal e as de comunicação audiovisual/telemática. ([MORAN, 2009, p. 32](#))

Ao utilizar tecnologia em sala, o professor não se exime da responsabilidade que possui, é necessário adequar às necessidades dos alunos, nivelando o conhecimento entre todos e fazendo com que consigam absorver e compreender os conteúdos de forma concreta. Para [Ferreira \(2014\)](#):

Essas novas tecnologias trouxeram grande impacto sobre a Educação, criando novas formas de aprendizado, disseminação do conhecimento e especialmente, novas relações entre professor e aluno. Existe hoje grande preocupação com a melhoria da escola, expressa, sobretudo, nos resultados de aprendizagem dos

seus alunos. Estar informado é um dos fatores primordiais nesse contexto. Assim sendo, as escolas não podem permanecer alheias ao processo de desenvolvimento tecnológico ou à pena de perder-se em meio a todo este processo de reestruturação educacional (FERREIRA, 2014, p. 15).

Com isso, a utilização de tecnologias em sala de aula se torna um grande diferencial para o processo de ensino, o professor precisa dominar esta ferramenta, sabendo usá-la ao seu favor, despertando a curiosidade do aluno, fazendo com que ele tenha interesse e consiga compreender os conteúdos de forma concreta. As tecnologias já fazem parte do âmbito escolar e da realidade dos alunos e professores, mas isso não garante que estão sendo utilizadas de forma correta e gerando os resultados esperados, que neste caso em questão, se trata da capacidade de promover uma aprendizagem mão na massa e auxiliar o aluno a ser o autor de seu conhecimento.

O uso das tecnologias em sala de aula também é defendido e sugerido pela Base Nacional Comum Curricular (BNCC), que traz dentre as competências gerais o desenvolvimento e promoção da cultura digital, que dentro do ambiente escolar, se refere a como os alunos se relacionam com as tecnologias e como eles se utilizam dela de maneira responsável, crítica e criativa, para fins de viabilizar o conhecimento e desenvolvimento de habilidades. De todas as competências gerais, há duas que fazem parte do contexto tecnológico de modo mais direto, a saber:

Competência 4: Comunicação - Utilizar diferentes linguagens - verbal (oral ou visual-motora, como Libras e escrita), corporal, visual, sonora e digital -, bem como conhecimentos das linguagens artística, matemática e científica para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo (BRASIL, 2018).

Competência 5: Cultura digital - Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva (BRASIL, 2018).

Essas competências se relacionam com a utilização das tecnologias em sala, trabalhando não somente a Matemática como também outras áreas do conhecimento. Nessa perspectiva, a BNCC propõe o uso da Robótica Educacional (RE) como uma ferramenta inovadora, em meio a esse contexto tecnológico, para auxiliar no campo educativo. Segundo Zilli et al. (2004):

A Robótica Educacional é um recurso tecnológico bastante interessante e rico no processo de ensino-aprendizagem, ela contempla o desenvolvimento pleno do aluno, pois propicia uma atividade dinâmica, permitindo a construção cultural e, enquanto cidadão tornando-o autônomo, independente e responsável (ZILLI et al., 2004, p. 77).

Para ensinar através da RE, é necessário trabalhar de maneira interdisciplinar, através do pensamento lógico, da proatividade, organização de ideias e solucionando problemas propostos.

Essa metodologia se relaciona com a abordagem *STEAM* (*Science, Technology, Engineering, Arts and Mathematics*), que segundo [Vaz, Júnior e Rocha \(2019\)](#) é uma tendência mundial de ensino que emerge em contraponto às metodologias tradicionais, propondo ambientes de aprendizagem mais criativos e inventivos e tornando os estudantes protagonistas na construção do conhecimento através da resolução de problemas que envolvam ações multidisciplinares.

Além disso, também está de acordo com o Movimento *Maker*, cujo princípio é “faça você mesmo”, em que os *makers*, que são pessoas que participam do movimento, tem uma identidade que prioriza o protagonismo, a busca por encontrar soluções para um problema, a busca pelo seu próprio conhecimento, criando, recriando, modificando, melhorando, fabricando, prototipando e projetando, ele não espera pelo outro, ele mesmo o faz, essas são as considerações de [Silva e Gil \(2019\)](#).

Nesse sentido, [Silva e Gil \(2019\)](#) também concluem que acima de tudo, o movimento *maker* propõe a diversão, construir o conhecimento, compartilhá-lo, fortalecendo as relações sociais e em equipe. Diante do exposto e, para responder à pergunta inicial deste trabalho, elencamos o seguinte objetivo geral:

- Investigar o potencial da Robótica Educacional para a promoção de uma aprendizagem mais autônoma e inovadora em Matemática.

Para que o objetivo geral seja atingido, elencamos os seguintes objetivos específicos:

- a) Compreender os princípios da cultura *MAKER*;
- b) Compreender os princípios da abordagem *STEAM*;
- c) Apresentar o conceito de Robótica Educacional;
- d) Apresentar o conceito de Pensamento Computacional;
- e) Elaborar proposta de oficina aliando a Matemática à Robótica Educacional.

Sendo assim, este trabalho trata-se de uma pesquisa aplicada, onde será desenvolvido um conhecimento sobre a capacidade de atividades de RE proporcionarem uma aprendizagem inovadora e autônoma, tendo como produto educacional, uma proposta de oficina, onde será colocado o objeto de estudo em prática. [Rodrigues et al. \(2007\)](#) afirma que a finalidade da pesquisa aplicada é fazer com que os conhecimentos adquiridos sejam utilizados para aplicação prática voltados para a solução de problemas concretos. Com isso, o objetivo de mostrar o potencial da RE na promoção de ambientes de aprendizagem inovadores podendo tornar o processo de ensino mais proveitoso será colocado em prática através da oficina *maker*, que será proposta com intuito de fazer com que os alunos aprendam através de situações-problemas reais.

A pesquisa é também de cunho exploratório, baseada em trabalhos acadêmicos e científicos, livros e artigos que abordem o objeto de estudo em questão. Para Severino (2017) a pesquisa exploratória visa apenas levantar informações sobre um determinado objeto, delimitando um campo de trabalho, mapeando as condições de manifestação desse objeto. Logo, será realizado um estudo sobre as características do Arduíno e do *Autodesk Tinkercad* como ferramentas para promover atividades de RE e de como ambos podem auxiliar os alunos na aprendizagem de matemática, sendo uma ferramenta inovadora capaz de estimulá-los a serem autônomos e criativos.

Tais pesquisas foram realizadas em plataformas como: Portal de Periódicos da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e *Google Scholar*, como forma de verificar o que já foi apresentado a respeito dos assuntos e a contribuição de alguns autores. A busca foi realizada utilizando palavras chaves como “Robótica Educacional”, “STEAM” e “Cultura *maker*”, priorizando trabalhos relacionados sempre a educação matemática. A partir desta busca foi possível perceber que os trabalhos que envolvem a robótica educacional e a matemática ainda são iniciais, com foco nos relatos de experiência em sala de aula, associando-o à BNCC e ao desenvolvimento do pensamento computacional.

O presente trabalho está organizado três capítulos, com introdução, considerações finais e anexos.

No primeiro capítulo dialoga com os itens a), b), c) e e) dos objetivos específicos. Através de uma revisão bibliográfica serão apresentados os princípios da cultura *maker*, a abordagem *STEAM*, a definição de Robótica Educacional e o conceito de Pensamento Computacional, mostrando como juntos eles podem promover um caminho para uma aprendizagem mão na massa baseada no pressuposto de aprenderem fazendo.

No segundo capítulo apresentaremos as ferramentas necessárias para que através da RE possamos desenvolver atividades para os alunos aprenderem matemática de uma maneira inovadora e prática. Serão apresentados o Arduíno e a plataforma *Autodesk Tinkercad*, mostrando o potencial e as principais características de ambas as ferramentas.

No terceiro capítulo será feita a proposta de oficina *maker* com o intuito de promover uma nova abordagem onde os alunos sejam os autores do conhecimento, aprendendo na prática e com a mão na massa, trabalhando a interdisciplinaridade e deixando o processo de aprendizagem mais leve e divertido, tudo isso através de problemas concretos do dia a dia dos alunos.

Finalizamos o trabalho nas Considerações Finais, retomando algumas ideias apresentadas e apresentando possibilidades de ampliação deste trabalho.

No que segue, falaremos sobre os princípios norteadores deste trabalho, ou seja, as bases teóricas que fundamentam nossa proposta.

1

Princípios Norteadores

Neste capítulo apresentaremos os princípios que conduzem esta pesquisa e são fundamentais para o entendimento do que está sendo estudado. Serão apresentados os princípios da cultura *maker*, a abordagem *STEAM* e o potencial de se trabalhar com a interdisciplinaridade, o conceito de Robótica educacional e a capacidade de criação de ambientes de aprendizagem inovadores, além de apresentar também o conceito de Pensamento Computacional e as habilidades desenvolvidas através dele.

1.1 A Cultura *MAKER*

Para além do ambiente escolar no qual sempre associamos o ensino, podemos perceber novos espaços emergindo no cenário atual. Espaços livres, autênticos e que promovem a autonomia dos alunos, esses espaços são conhecidos como espaços *makers*. Ao adentrar no Universo do Movimento *Maker* podemos nos deparar com o princípio “faça você mesmo”, que norteia toda a cultura que envolve os participantes do movimento. Para [Lopes et al. \(2019\)](#) a Cultura *Maker* é definida como:

Um conjunto de comportamentos, conhecimentos, valores, crenças e costumes que têm como base as ideias de que pessoas participativas podem se envolver em projetos que as façam se sentir capazes de construir coisas novas, ou seja, gerando seu empoderamento. Hoje a cultura maker se faz presente das garagens às universidades, passando por empresas, espaços colaborativos de criação, lugares de educação formal e informal. ([LOPES et al., 2019](#), p. 2)

Esse movimento se baseia no pressuposto de que qualquer pessoa pode criar, construir, consertar, fabricar objetos e realizar projetos, com materiais de baixo custo ou até mesmo reciclados, fazendo tudo com as próprias mãos. Para [SILVEIRA \(2016, p. 131\)](#), esses espaços “estimulam as pessoas comuns a construírem, modificarem, consertarem e fabricarem os próprios

objetos, com as próprias mãos”. Promovendo assim a ação de colocar a mão na massa, aliando ferramentas tecnológicas e materiais de baixo custo, proporcionando aos alunos através dessas atividades o protagonismo no processo de aprendizagem. [Oliveira, Santos e Souza \(2018\)](#) nos dizem que:

É perceptível que a cultura maker não influencia apenas nos aprendizados teóricos, não proporciona apenas uma ligação com os assuntos que são ou serão ministrados em sala, mas também provoca uma mudança de postura do aluno, tornando-o mais curioso e apto para questionar, inovar e produzir. Aprimora sua capacidade de percepção, investigação, raciocínio lógico e engenhosidade ([OLIVEIRA; SANTOS; SOUZA, 2018](#), p.283).

Ao inserir o aluno em um espaço onde ele é diretamente instigado a resolver problemas reais, a “fazer ele mesmo” com que projetos ganhem vida e saiam do papel, onde possa aprender com os erros e desenvolver novas habilidades, além de absorver conteúdos ensinados em sala de aula, esse aluno é capaz de mudar a postura fora do espaço *maker*, aprimorando mais suas engenhosidades no dia a dia.

As ideias defendidas pelos *makers* também se relacionam com as ideias defendidas por Paulo Freire, que reforçam um ensino onde o aluno possa ser autônomo e criativo, havendo uma troca mútua e muita colaboração. Uma aula *maker* é capaz de possibilitar “o desenvolvimento de habilidades de cooperação e comunicação entre os estudantes, que se envolvem em um projeto incomum” ([YOUNG DIGITAL PLANET S. A. et al., 2016](#), p. 273). Com base nisso, podemos identificar os espaços *makers* como espaços onde a aprendizagem potencializa a criatividade, autonomia e interdisciplinaridade.

Quando pensamos sobre a Cultura *Maker* inserida dentro do contexto escolar, [Paula, Martins e Oliveira \(2021 apud BRASIL, 2018, p. 2\)](#) afirmam:

Embora o termo não tenha sido proposto para ser explorado nas escolas, a abordagem *maker* promove o desenvolvimento do trabalho em grupo, habilidades sociais, autonomia, criatividade, uso da tecnologia, além de ampliar a comunicação, o saber se expressar, se apresentar e explorar conteúdos previstos no currículo acadêmico, ou seja, na Base Nacional Comum Curricular. ([PAULA; MARTINS; OLIVEIRA, 2021 apud BRASIL, 2018, p. 2](#)).

A BNCC traz essa proposta de metodologias de ensino que possam promover uma aprendizagem colaborativa e inovadora para os alunos. Sendo assim, a cultura *maker*, que dialoga fortemente com a educação, se aplicada em sala de aula, pode auxiliar no processo de ensino, desenvolvendo habilidades além das já previstas, tornando o aluno protagonista de seu conhecimento, valorizando a “mão na massa”.

1.2 Abordagem STEAM

Junto a Cultura *Maker* e os seus espaços, emerge também uma abordagem conhecida pelo acrônimo *STEAM* (Science, Technology, Engineering, Arts and Mathematics), o termo surgiu nos Estados Unidos e representa a junção das iniciais das áreas: ciências, tecnologias, engenharia, artes e matemática. A educação *STEAM* não é uma metodologia e sim, uma abordagem, [Riley \(2020\)](#) afirma:

[...] uma abordagem de aprendizagem que usa Ciência, Tecnologia, Engenharia, Artes e Matemática como pontos de acesso para orientar a investigação, o diálogo e o pensamento crítico discente. ([RILEY, 2020](#), p. 1)

A proposta da educação *STEAM* é justamente trabalhar a integração entre as áreas citadas e através da interdisciplinaridade proporcionar uma aprendizagem que instigue o pensamento criativo. [Lorenzin, Assumpção e Rabello \(2016\)](#) esclarecem que, nesta abordagem, os temas são explorados de forma investigativa, partindo da proposição de um problema, podendo se expandir para a realização de projetos, pautados na aprendizagem ativa.

Através desse diálogo interdisciplinar, essa educação visa suprir necessidades que podem acarretar problemas no processo de ensino, proporcionando uma conexão entre assuntos de várias áreas, gerando um melhor entendimento, tornando o processo de ensino mais expressivo do que se os conteúdos fossem fragmentados. Visualizando então a abordagem *STEAM* como alternativa para enfrentar desafios no processo de ensino e aprendizagem, nesse sentido [Bacich e Holanda \(2020\)](#) nos diz que:

[...] a educação *STEAM* pode contribuir para lidar com os desafios contemporâneos, ajudando a pensar uma educação que, sem abandonar a excelência acadêmica, também desenvolva competências importantes, como a criatividade, o pensamento crítico, a comunicação e a colaboração. ([BACICH; HOLANDA, 2020](#), p. 2)

Assim, com a capacidade de auxiliar em desafios no âmbito escolar, e com propósito pedagógico de unir as cinco áreas que congrega, essa abordagem se mostra importante para ajudar a chegar na aprendizagem criativa almejada no presente trabalho. Observamos assim, que a cultura *maker* aliada a metodologia *STEAM* favorecem um ensino multidisciplinar, promovendo a autonomia e estimulando o pensamento criativo, para [Maia, Carvalho e Appelt \(2021\)](#) apud [RESNICK, 2020](#), p. 71):

a Educação *STEAM* favorece a aprendizagem criativa e mão-na-massa (*maker*) que oportuniza aos alunos aprendizagem por meio do desenvolvimento de projetos, com seus pares, com engajamento e por experimentação. ([MAIA; CARVALHO; APPELT, 2021](#) apud [RESNICK, 2020](#), p. 71)

Além de proporcionar experiências onde o aluno é o autor na construção do seu conhecimento, através do “faça você mesmo”, potencializando as ideias defendidas por Resnick, que conversam com os princípios dos *makers* e com a proposta da abordagem *STEAM*.

1.3 Robótica Educacional

Com a demanda de utilização das TDICs na educação, ferramentas computacionais podem ser introduzidas no ambiente escolar de inúmeras formas fazendo com que haja estímulo para a aprendizagem do aluno. Em razão disso, uma das ferramentas utilizada é a Robótica Educacional, que vem ganhando destaque na área da educação por ser um método que relaciona a colaboração, troca de conhecimentos, proatividade, experimentação e prototipagem, entre outras características segundo [Noemi \(2020\)](#). Em suas considerações, [Ferreira \(1991\)](#) nos diz que:

A robótica é considerada hoje a mola mestra de uma nova mutação dos meios de produção, isto devido a sua versatilidade, em oposição à automação fixa ou “hard”, atualmente dominante na indústria. Os robôs, graças ao seu sistema lógico ou informático, podem ser reprogramados e utilizados em uma grande variedade de tarefas. Mas, não é a reprogramação o fator mais importante na versatilidade desejada e sim a adaptação às variações no seu ambiente de trabalho, mediante um sistema adequado de percepção e tratamento de informação. ([FERREIRA, 1991](#), p. 4)

A Robótica aliada a educação, tem como objetivo promover ao aluno o estudo de conceitos multidisciplinares, está constantemente associada a cultura *maker* e ao movimento *STEAM* pela sua capacidade de agregar conhecimentos de diferentes áreas, estimulando a criatividade e promovendo a interdisciplinaridade.

A prática da Robótica também é defendida por duas teorias muito importantes para a área da educação, o construtivismo de Piaget e o construcionismo de Papert. [Piaget \(1974\)](#) nos diz que manipular artefatos se torna a chave para que a criança construa seu conhecimento, [Papert \(1980\)](#) acrescentou que o conhecimento ocorre de maneira afetiva em um ambiente onde o aluno esteja sendo estimulado a construir um objeto, seja ele um castelo de areia na praia ou um artefato tecnológico.

O professor possui um papel fundamental nesse processo da inserção da RE na escola. Promovendo experiências de um ensino "mão na massa", oferecendo ferramentas que proporcionem a construção do conhecimento, criando um ambiente de aprendizagem no qual o aluno interaja e trabalhe problemas reais do seu dia a dia.

Para [Campos \(2017\)](#), a Robótica Educacional é vista sob três óticas no currículo escolar:

Currículo por tema: o currículo é desenhado a partir de um tema de saber específico, sendo organizado de forma disciplinar ou interdisciplinar. Nesta categoria, o currículo direciona-se para o aprendizado da robótica e sua tecnologia ou no uso da robótica para o aprendizado de conceitos de diferentes áreas do saber como, por exemplo, da física, matemática, ciências, etc.

Currículo por projeto: o currículo é desenhado para o desenvolvimento de projetos que envolvem vários temas/conteúdos, também se caracterizando pelo aprendizado da robótica e suas tecnologias ou para o aprendizado de conceitos de diferentes áreas dos saberes.

Currículo por Objetivo/competição: os alunos desenvolvem atividades que visam à participação em eventos e competições de robótica, trabalhando habilidades relacionadas à participação nos desafios propostos. (CAMPOS, 2017, p. 4)

Podemos então dizer, que RE se trabalhada dentro de um destes três tópicos, de maneira com que o professor lhe ofereça as condições necessárias anteriormente citadas, pode proporcionar um ambiente de aprendizagem criativa, onde os alunos ponham a “mão na massa” solucionando situações-problemas que devem ser resolvidas através de erros até que os objetivos propostos sejam alcançados.

1.4 Pensamento Computacional

O Pensamento Computacional (PC) refere-se a ideia de um conglomerado de habilidades relacionadas as formas nas quais as pessoas podem utilizar os computadores para solucionar situações-problemas. Podendo então criar e moldar soluções para tais problemas de forma conveniente utilizando a tecnologia. Segundo Nunes et al. (2021 apud WING, 2006, p. 3):

O conceito de PC foi publicado pela primeira vez em 2006 por Jeanette Wing, que defende que o conceito contempla muitas habilidades e abstrações, não apenas de fazer algo, mas sim de operar, fazer, realizar, ler, estudar, compreender, modificar, ajustar, e em diferentes níveis de compreensões. (NUNES et al., 2021 apud WING, 2006, p. 3)

O conceito de pensamento computacional vai além do conceito de letramento computacional que retrata apenas a utilização de forma intuitiva as interfaces tecnológicas e a habilidade de saber operar de forma correta o computador. O PC busca desenvolver essas habilidades retratadas por Wing nos alunos, faz com que eles aprimorem a capacidade de resolução de problemas, auxiliando e se relacionando com outras ciências. Além das características citadas acima, o PC pode desenvolver também habilidades analíticas e metacognitivas, além de interação social e criatividade Webber et al. (2022, p. 132).

Mesmo sendo relacionado a ciência da computação, os benefícios do PC incluem diversas áreas do conhecimento que buscam a utilização de modelos computacionais para resolução de problemas. No que se refere a matemática sabe-se que uma das competências esperadas dos alunos é que eles saibam identificar códigos e símbolos da linguagem matemática e espera-se que eles saibam traduzi-los, dada uma linguagem em outra e saibam traduzir uma representação em outra, segundo Barcelos e Silveira (2012). A utilização do pensamento computacional articulado à matemática pode proporcionar a capacidade de criar novos conhecimentos, melhorando a compreensão trabalhando através da interdisciplinaridade, para que assim consigam uma melhor conceitualização, análise e resolução de problemas mais complexos.

Representar a solução para uma determinada situação na linguagem algorítmica é uma das competências fundamentais do pensamento computacional Barcelos e Silveira (2012 apud

THE CSTA, 2011, p. 6) e através dessa linguagem descritiva do algoritmo pode proporcionar ao aluno a oportunidade de moldar suas hipóteses e testá-las, tornando-o autor da construção de seu conhecimento matemático a partir da construção do algoritmo. Desta forma, podemos observar que o PC busca uma perspectiva de um aluno que explore sua realidade e tudo que o cerca e consiga estabelecer relações entre o seu mundo e a matemática.

Barcelos e Silveira (2012 apud MOR; NOSS, 2008, p. 6) nos diz que:

A natureza sequencial do algoritmo, definido através de procedimentos a serem seguidos passo a passo, o aproxima da linguagem discursiva. Dessa forma, representar um problema na forma algorítmica pode se constituir como uma etapa intermediária entre a narração verbal e a linguagem algébrica, podendo promover uma transição mais “suave” para a compreensão da linguagem matemática. (BARCELOS; SILVEIRA, 2012 apud MOR; NOSS, 2008, p. 6)

Com isso, a utilização do PC proporciona o desenvolvimento de habilidades necessárias nos alunos. Ao representarem um problema de maneira algorítmica, eles acabam facilitando a compreensão da linguagem matemática através do reconhecimento de padrões, elaboração e interpretação de modelos de representações matemáticas para analisar situações. Desse modo, trabalharemos atividades de RE que desenvolvam o PC utilizando situações-problemas que possam estimular os alunos a solucionarem as atividades através da linguagem algorítmica utilizando ferramentas como Arduino e Autodesk Tinkercad que serão apresentados no próximo capítulo.

2

As Ferramentas para a Robótica Educacional

Neste capítulo serão apresentados o Arduíno e a plataforma *on-line Autodesk Tinkercad*, ferramentas escolhidas e necessárias para se trabalhar a Robótica Educacional em sala de aula e desenvolver atividades *makers* proporcionando aos alunos uma aprendizagem mão na massa.

2.1 Arduíno

Para trabalhar a Robótica Educacional como uma metodologia de ensino e desenvolver as atividades que serão propostas no capítulo seguinte, foi escolhida a placa de prototipagem eletrônica Arduíno, que é capaz de desenvolver projetos de automação, existem outras placas *makers* como *Raspberry Pi*, *Micro:bit* e *Makey makey*. Porém, a escolha se deu a partir da facilidade de acesso, pois o Arduíno possui tecnologia *open source*, o que significa que é um *software* de código aberto, onde o código fonte é disponibilizado e licenciado com uma licença de código aberto onde o direito autoral permite o estudo, modificação e distribuição gratuita do *software* para qualquer pessoa e finalidade.

O Arduíno surgiu na Itália em 2005, idealizado por Massimo Banzi e David Cuartielles e com auxílio de David Mellis aluno de Massimo, a proposta era que o dispositivo tivesse baixo custo para que estudantes e também pessoas interessadas pudessem desenvolver experiências com Robótica e produzir de forma simples e barata.

Construir ou programar um robô até pouco tempo atrás era algo que parecia exclusivo para grandes empresas de *hardwares* ¹, trazer isso para sala de aula faz com que os alunos sintam que são capazes de fazer grandes projetos, que na verdade programar robôs ou fazer projetos de automação é para qualquer pessoa, resolver problemas do cotidiano utilizando Robótica é

¹ O hardware é um termo técnico que foi traduzido para a língua portuguesa como equipamento, e pode ser definido como um termo geral da língua inglesa, que se refere à parte física de computadores e outros sistemas microeletrônicos.

algo possível. O Arduíno foi desenvolvido com esse intuito, no sentido de mudar essa condição e tornar mais acessível a realidade, abrindo possibilidades para muitas pessoas programadoras e alunos interessados.

2.1.1 O que compõe a placa Arduíno?

Existem diversos modelos de placas do Arduíno disponíveis no mercado, como: Arduíno Nano, Arduíno Micro, Arduíno Uno, Arduíno Mega, entre outros. Mas de maneira geral, [Robótica \(2012\)](#) nos diz que:

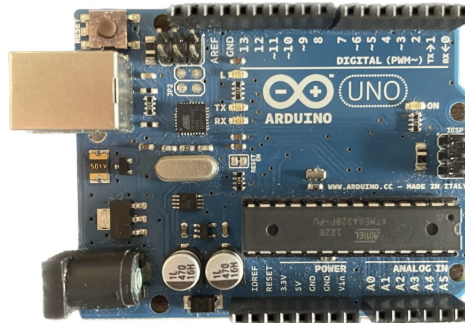
Arduíno é uma placa de controle de entrada de dados (IN), como sensores, e saída de dados (OUT), como motores e leds, com cristal oscilador de 16 Mhz, um regulador de tensão de 5 V, botão de reset, plugue de alimentação, pinos conectores, e alguns LEDs para facilitar a verificação do funcionamento. ([ROBÓTICA, 2012](#), p. 3)

Para fornecer energia para a placa basta conecta-la através de um cabo USB à um computador ou através de cabo de alimentação que varia de 7V a 12V. Ele também é composto por Ambiente de Desenvolvimento Integrado, para o desenvolvimento do *software* que vai ser introduzido na placa para realizar o que foi programado.

O arduíno então é composto por duas partes: o *hardware* e o Ambiente de desenvolvimento Integrado (IDE) que também é *open source*, para então colocar em prática o *software* desenvolvido. Esse ambiente pode ser baixado de forma gratuita no site oficial do Arduíno e possui versões para Windows, Linux e macOS.

O modelo de Arduíno utilizado será o Arduíno Uno R3, é o recomendado para projetos iniciais. Contendo diversos pinos, sendo eles 6 pinos de entradas analógicas, que podem ler sensores analógicos transformando em um sinal digital e 14 entradas e saídas digitais que entendem sinais digitais, podendo determinar quando um botão foi acionado e enviar comandos. Além de possuir botão de *reset* que reinicia a placa e o indicador de LED de energia indicando quando a placa está sendo energizada.

Figura 1 – Placa Arduino UNO R3



Fonte: autora.

2.1.2 A Linguagem de Programação e Ambiente de Desenvolvimento

A linguagem de programação utilizada no Arduino é baseada na linguagem Wiring ², desenvolvida nas linguagens C\C++, que é voltada para a programação de microcontroladores. Possuindo estruturas de controle, tipos de operações, funções e etc. Contendo também dois blocos principais que são necessários, são eles: *setup()* sendo responsável por definição de variáveis e utilização das portas, e *loop()* que executa os comandos de maneira infinita até que haja alguma condição de parada.

Figura 2 – Software Arduino IDE



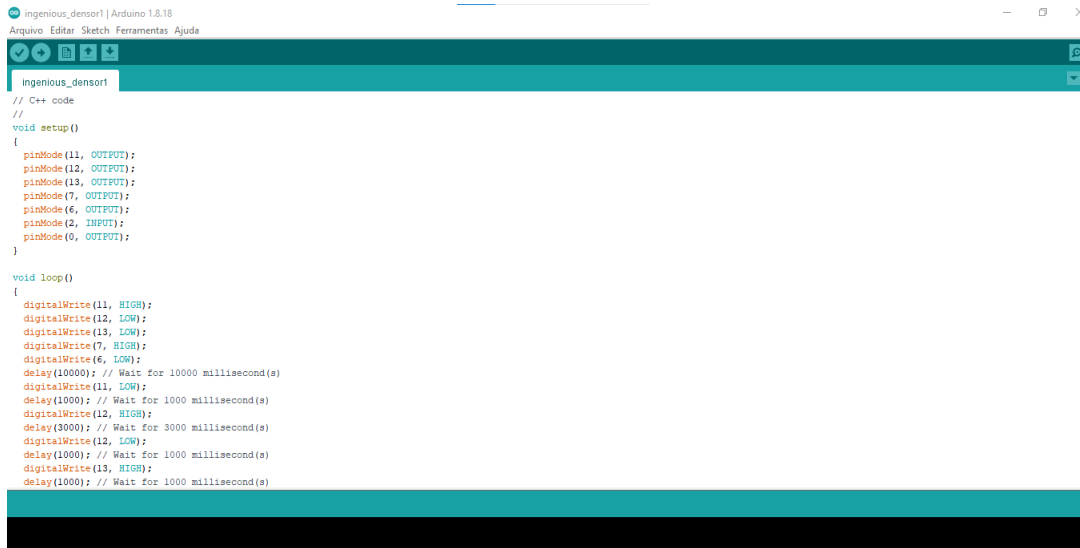
Fonte: autora.

A programação do código fonte é feita no ambiente de desenvolvimento e é transferida para a placa através da conexão USB, *Bluetooth*, *wireless*, ou infravermelho. Esse ambiente possui um editor de texto para a escrita dos códigos, área de mensagens, barra de ferramentas,

² Wiring é uma estrutura de programação de código aberto para microcontroladores.

console de texto e alguns menus. Esse ambiente se conecta com o *hardware* do Arduino e permite então o *upload* dos softwares programados e faz então a integração entre eles.

Figura 3 – Interface do Ambiente de Desenvolvimento do Arduino



```
ingenious_densort | Arduino 1.8.18
Arquivo  Editar  Sketch  Ferramentas  Ajuda
ingenious_densort
// C++ code
//
void setup()
{
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(2, INPUT);
  pinMode(0, OUTPUT);
}

void loop()
{
  digitalWrite(11, HIGH);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
  digitalWrite(7, HIGH);
  digitalWrite(6, LOW);
  delay(10000); // Wait for 10000 millisecond(s)
  digitalWrite(11, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(12, HIGH);
  delay(3000); // Wait for 3000 millisecond(s)
  digitalWrite(12, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
```

Fonte: autora

No Apêndice A podemos ver alguns dos principais comandos da Linguagem de Programação do Arduino.

2.1.3 Alguns Componentes

Além da placa Arduino, também existem inúmeros componentes que podem ser utilizados junto com a placa, agregando funcionalidades, auxiliando nos projetos e deixando eles ainda mais elaborados.

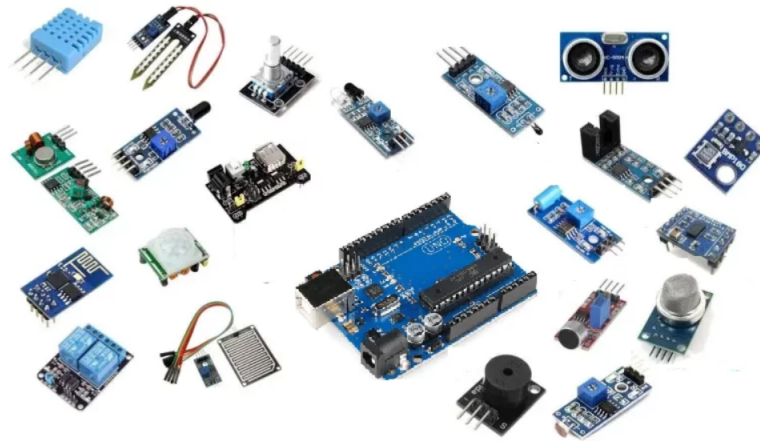
Sensores e Módulos

Existem muitos sensores e módulos que podemos conectar a placa do Arduino, como: sensor de distância ultrassônico, sensor fotoresistor, sensor de umidade e temperatura, sensor de luz, sensor de gás, sensor de fluxo de água, módulo de *bluetooth*, módulo de memória, módulo sensor de cor, entre muitos outros.

Módulos são circuitos inteiros que são condensados em uma placa, expandindo a capacidade do Arduino e fazendo com que realize funções que sozinho não seria capaz de realizar sozinho. Esses módulos podem tratar o sinal de de um sensor, controlar motores, conectar *bluetooth* e WI-FI e etc.

Já os sensores são dispositivos que respondem a estímulos físicos ou químicos, produzindo sinais que podem ser transformados em outras grandezas, para fins de medida e monitoramento. Estes podem ser considerados ferramentas de grande importância para os projetos devido suas funcionalidades e variedade. Assim, ao aliarmos os sensores à módulos, esse sensor transforma o estímulo captado pelo módulo em uma grandeza, convertendo o tipo de energia para fins de medição. Vejamos na imagem a seguir alguns exemplos:

Figura 4 – Sensores e Módulos



Fonte: [CAP SISTEMA](#).

Podemos observar na figura 4 alguns módulos e sensores como: módulo ultrassônico HC-SR04, sensor infravermelho, sensor de umidade do solo, sensor de microfone, sensor de pressão barométrica, sensor fotorresistor, sensor térmico digital, sensor de Detecção de Chuva e Umidade, módulo de campainha passiva e outros.

Resistores e Leds

Resistores e leitura de resistência

Resistores são componentes eletrônicos cuja principal função é limitar o fluxo de cargas elétricas por meio da conversão da energia elétrica em energia térmica. Os resistores são geralmente feitos a partir de materiais dielétricos, de grande resistência elétrica. A grande resistência elétrica dos resistores torna esses componentes capazes de reduzir a passagem da corrente elétrica.

Há, entre os resistores, aqueles que apresentam resistência elétrica variável, entre esses resistores, o mais simples é o varistor, que consiste em um longo resistor de resistência variável, já que essa grandeza diminui de acordo com o aumento de temperatura. Existem também resistores cuja resistência elétrica varia com a temperatura chamados de termorresistores, há os resistores que variam de resistência elétrica quando iluminados, que são chamados de fotorresistores.

Mesmo havendo tantos resistores diferentes, os resistores de maior interesse para este trabalho são aqueles que apresentam resistência fixa, conhecidos como resistores ôhmicos ³.

Figura 5 – Resistores Ôhmicos



Fonte: autora.

Podemos medir a resistência elétrica de um resistor a partir da leitura de seu corpo, onde as cores que ficam impressas em volta determinam o valor. Confira a tabela de código de cores:

Figura 6 – Código de Cores

COR	1ª FAIXA	2ª FAIXA	3ª FAIXA	4ª FAIXA	5ª FAIXA
	VALOR	VALOR	VALOR	MULTIPLICADOR	TOLERÂNCIA
PRETO		0	0	X 1 Ω	
MARROM	1	1	1	X 10 Ω	Mais ou menos 1 %
VERMELHO	2	2	2	X 100 Ω	Mais ou menos 2 %
LARANJA	3	3	3	X 1.000 Ω	
AMARELO	4	4	4	X 10.000 Ω	
VERDE	5	5	5	X 100.000 Ω	Mais ou menos 0,5%
AZUL	6	6	6	X 1.000.000 Ω	Mais ou menos 0,25%
VIOLETA	7	7	7	X 10.000.000 Ω	Mais ou menos 0,1%
CINZA	8	8	8		Mais ou menos 0,05%
BRANCO	9	9	9		
DOURADO				X 0.1 Ω	Mais ou menos 5 %
PRATA				X 0.01 Ω	Mais ou menos 10 %
SEM COR					Mais ou menos 20 %

Fonte: APRENDENDO ELÉTRICA.

A leitura das faixas (para um resistor de até 5 faixas) se dá da seguinte maneira:

³ A resistência não depende da tensão elétrica aplicada sobre eles, isso equivale dizer que esse tipo de resistor obedece à condição estabelecida pela Primeira Lei de Ohm. Link de acesso: <<https://shre.ink/resistores-mundo-educacao>>

- 1ª Faixa: Primeiro número do valor do resistor;
- 2ª Faixa: Segundo número do valor do resistor;
- 3ª Faixa: Terceiro número do valor do resistor;
- 4ª Faixa: Fator de multiplicação do valor do resistor;
- 5ª Faixa: Tolerância do valor da resistência (indica quanto a resistência pode variar para mais ou para menos do seu valor).

LEDs e cálculo de resistência

A estrutura básica de um LED é composta por dois terminais semicondutores chamados ânodo (lado menor) e cátodo (lado maior). Esses componentes ficam dentro de cápsulas com formatos e tamanhos variados. LEDs para lâmpadas têm entre 3 e 10 milímetros ⁴.

Para calcular a resistência, consideramos que a unidade de resistência elétrica é ohm, cujo símbolo é Ω (ômega) como manda o Sistema Internacional de Unidades (SI). Com isso, podemos definir a resistência (R) como:

$$R = \frac{U}{i}$$

Sendo:

U: a diferença de potencial (ddp);

i: intensidade da corrente elétrica;

R: a resistência elétrica.

Desta se faz então a dedução para a fórmula para calcular a resistência de um resistor:

$$R = \frac{(V_{cc} - V_{led})}{I_{led}}$$

Sendo:

R: a resistência;

V_{cc}: tensão da fonte;

V_{led}: Tensão de trabalho do LED;

I_{led}: Corrente (Amper)

Abaixo, mostramos a tabela de tensão e corrente de trabalho dos LEDs e suas cores:

⁴ Link de acesso: <<https://abrir.link/fF0wh>>

Figura 7 – Tabela de corrente e tensão de LEDs

Cor do led	Faixa de tensão	Corrente máxima
Vermelho	1,8 V - 2,0 V	20 mA
Amarelo	1,8 V - 2,0 V	20 mA
Laranja	1,8 V - 2,0 V	20 mA
Verde	2,0 V - 2,5 V	20 mA
Azul	2,5 V - 3,0 V	20 mA
Branco	2,5 V - 3,0 V	20 mA

Fonte: [APRENDENDO ELÉTRICA](#).

Como exemplo vamos calcular o valor de um resistor para ligar um LED vermelho a uma bateria com tensão de 12V, vejamos:

$$R = \frac{(V_{cc} - V_{led})}{I_{led}}$$

$$R = \frac{12 - 2}{0,02}$$

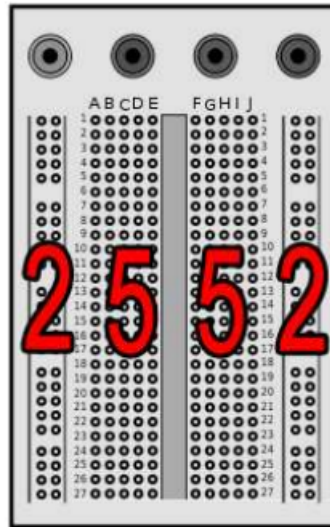
$$R = 500\Omega$$

2.1.4 Protoboard

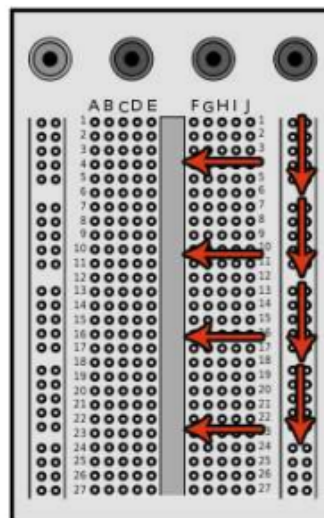
A *protoboard* é uma placa para prototipagem e facilita bastante a criação de projetos eletrônicos porque dispensa o uso de solda para testar circuitos. Isso faz com que possamos corrigir facilmente qualquer erro de projeto, a *protoboard* é reutilizável e justamente por isso, ela é indicada para projetos temporários, pois não precisa de solda.

Assim, com a *protoboard* podemos criar projetos eletrônicos de forma rápida e segura, todos os componentes vão encaixados e as interligações são feitas utilizando *jumpers* ⁵. No geral, uma *protoboard* possui 4 matrizes, podendo variar de acordo com a placa escolhida. Vejamos a imagem a seguir:

⁵ Um jumper é uma peça plástica que contém um pequeno filamento de metal responsável pela condução de eletricidade.

Figura 8 – *Protoboard* com divisão de 4 matrizesFonte: [ROBÓTICA](#).

A *protoboard* é composta por dois tipos de matrizes, a matriz 2 que indica duas colunas e a matriz 5 que indica 5 colunas, elas se diferenciam justamente pelo modo de transmissão de energia e dados. Enquanto a matriz 2 tem sua transmissão feita de coluna em coluna e é usada para fornecer energia a placa, a matriz 5 tem sua transmissão feita de linha em linha e é mais utilizada para os componentes do circuito.

Figura 9 – Sentidos de transmissão de uma *protoboard*Fonte: [ROBÓTICA](#).

2.2 Tinkercad

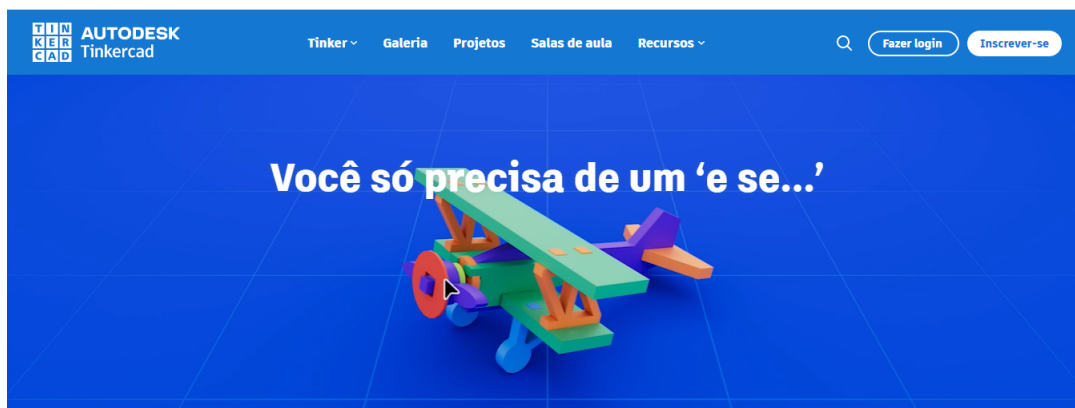
O *Tinkercad* é uma plataforma online, gratuita e de fácil acesso para projetos 3D, eletrônica e codificação. Permite a simulação, montagem e programação de protótipos de maneira virtual. Para [Oliveira et al. \(2022\)](#) os simuladores complementam a aprendizagem dos alunos pois permite que eles visualizem na prática o que foi ensinado teoricamente, minimizando o risco da perda de equipamentos e acidentes. Utilizando então o *tinkercad* pode-se realizar a simulação da construção de protótipos de maneira virtual, colocando em prática os princípios que foram ensinados na teoria.

[Oliveira et al. \(2022](#) apud [MEDEIROS; WÜNSCH, 2019](#), p. 96) inferem as seguintes considerações:

O simulador *Tinkercad* reproduz da forma mais fiel possível, o desempenho dos circuitos, constituindo-se numa ferramenta bastante útil para o aprendizado inicial de programação em Arduino, além de dar autonomia e motivação ao aluno no seu processo de aprendizagem, permitindo novas descobertas e possibilidades de uso. ([OLIVEIRA et al., 2022](#) apud [MEDEIROS; WÜNSCH, 2019](#), p. 96)

Por ser uma ferramenta *online*, para utilizar o *tinkercad* é necessário apenas o uso do navegador, como *Chrome*, *Edge* ou *Firefox*. A escolha deste *software* foi baseada justamente no fato de possuir livre acesso, estar disponível na *Web* e por apresentar fácil manipulação, sendo uma ferramenta que consegue reproduzir com fidelidade os protótipos que desejamos produzir fisicamente, fazendo com que aluno tenha mais confiança quando for construir o circuito eletrônico de forma física.

Figura 10 – Interface Inicial do *Tinkercad*



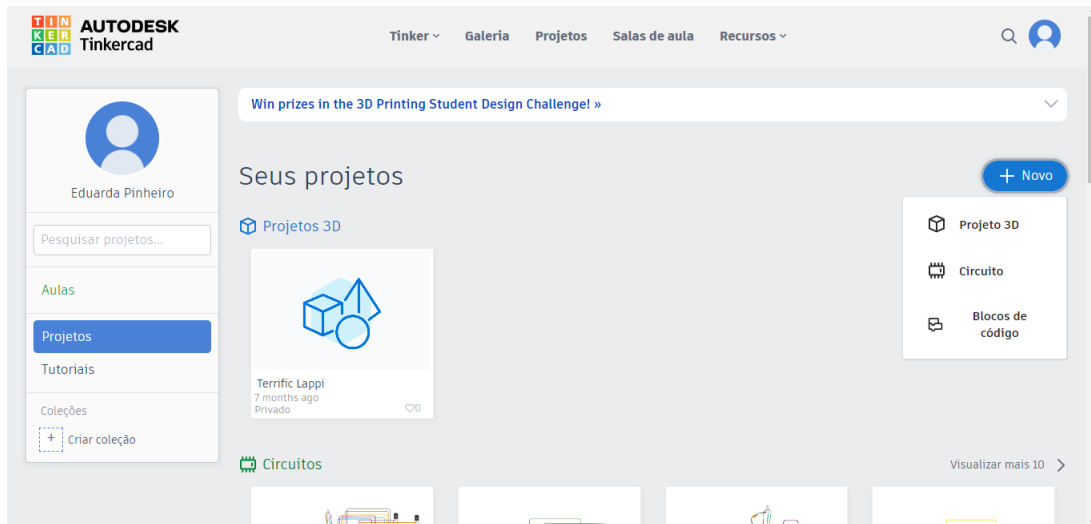
Fonte: autora.

Ao entrar no site do *Tinkercad* ⁶ é necessário criar uma conta para poder produzir projetos,

⁶ Link do site: <www.tinkercad.com>

assim que o *login*⁷ for feito, é aberta uma página inicial, mostrando o ambiente de projetos, onde podemos observar um ícone escrito “novo”, nele é possível escolher que tipo de projeto se quer criar, no caso deste trabalho, usaremos a opção “circuito”. Vejamos na imagem a seguir:

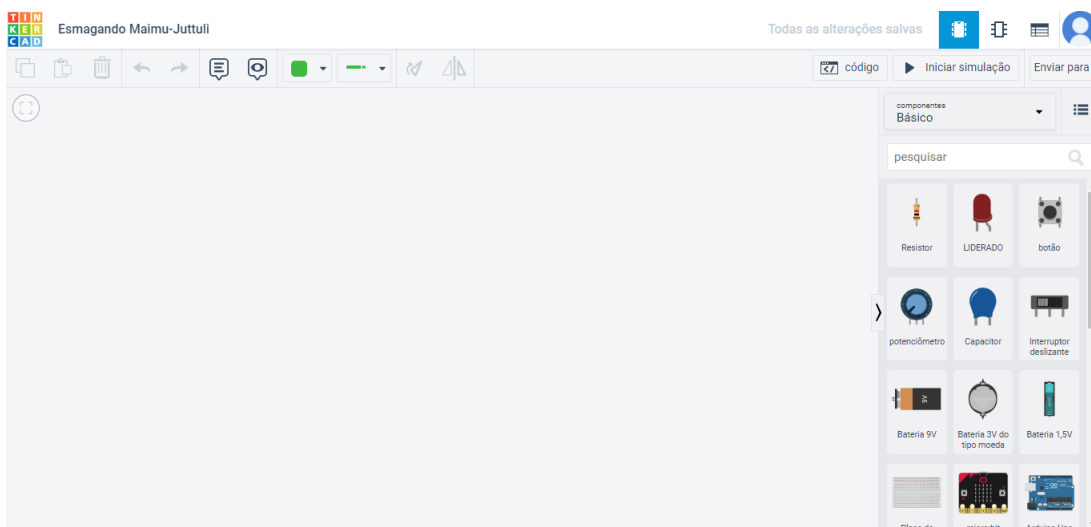
Figura 11 – Interface do ambiente de projetos do *Tinkercad*



Fonte: autora.

Ao escolher a opção *circuito*, o *tinkercad* abre a interface de trabalho, onde podemos observar a área de trabalho, a área onde estão disponíveis os sensores, módulos, LEDs, resistores, protoboard, as diferentes placas de arduino e muitas outras ferramentas.

Figura 12 – Interface de trabalho do *Tinkercad*



Fonte: autora.

⁷ Link passo a passo: <<https://3dfila.com.br/blog/tutorial-do-tinkercad-aprendendo-utilizar-o-seu-primeiro-software-de-modelagem-3d/>>

A manipulação é bem intuitiva, basta somente arrastar os componentes até a área de trabalho e montar o circuito elétrico e apertar o ícone que diz “iniciar simulação” para testar se seu circuito está correto, e se não estiver, o *software* emite comunicados e alertas de erros próximos aos componentes que não estão na disposição correta, além de possuir a opção “aprender” onde mostra todos os componentes e suas funcionalidades. Em relação a programação do circuito, o *tinkercad* oferece a possibilidade de programar tanto na linguagem C\C++ quanto também em blocos. Ao programar em blocos, o próprio *software* faz a conversão para a linguagem de programação do arduíno, onde é necessário baixar o projeto para poder abri-lo no ambiente de desenvolvimento.

3

Proposta de Oficina

Neste capítulo, será apresentada uma proposta de oficina voltada para estudantes e professores do Ensino Médio, com o objetivo de colocar em prática as potencialidades da Robótica Educacional como ferramenta que promove uma aprendizagem mão na massa em matemática.

A ideia principal desta oficina, é o estudante vivenciar diferentes etapas de pequenos projetos de RE. O processo utilizado para a resolução das atividades foi pensado da seguinte forma:

Teoria → Simulação → Prática

Inicialmente, os estudantes receberão um desafio/problema do cotidiano. Na primeira etapa – Teoria – o estudante deverá pensar em soluções diversas, realizar pesquisas sobre o tema, produzir ideias e algoritmos para tentar solucionar o desafio/problema. Na segunda etapa – Simulação – o estudante implementará computacionalmente a solução encontrada na etapa anterior, utilizando o *Tinkercad*. É neste momento que o estudante deverá programar o algoritmo, montar o circuito e testar computacionalmente o projeto. A partir disso, o estudante poderá refletir se a solução proposta é factível, se é necessário fazer adaptações ou encontrar uma nova solução. Finalmente, na última etapa – Prática – o estudante implementará fisicamente a solução, utilizando o Arduíno. Nesta etapa, é possível que novas questões surjam durante a execução, sendo necessário a inserção ou retirada de componentes no circuito, adaptação do código em função das variáveis da realidade (tempo, distância, etc), assim como, pensar em *upgrades* para o projeto.

É importante salientar que essa ordem proposta pode ser alterada, suprimindo a segunda etapa, em casos de dificuldades de acesso à internet. Contudo, a fase da “simulação” é de grande

relevância, pois é nela que os testes são realizados sem correr o risco de queimar ou prejudicar as placas e demais componentes eletrônicos. No que segue, apresentaremos a proposta de oficina.

3.1 Competências Trabalhadas

Serão trabalhadas duas Competências Gerais da BNCC (2018), a saber:

- Competência Geral nº 2: Pensamento Científico, Crítico e Criativo — Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas;
- Competência Geral nº 5: Cultura Digital — Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

Serão trabalhadas duas Competências Específicas de Matemática e suas Tecnologias para o Ensino Médio da BNCC, a saber:

- Competência Específica nº 1: Utilizar estratégias, conceitos e procedimentos matemáticos para interpretar situações em diversos contextos, sejam atividades cotidianas, sejam fatos das Ciências da Natureza e Humanas, das questões socioeconômicas ou tecnológicas, divulgados por diferentes meios, de modo a contribuir para uma formação geral.
- Competência Específica nº 2: Propor ou participar de ações para investigar desafios do mundo contemporâneo e tomar decisões éticas e socialmente responsáveis, com base na análise de problemas sociais, como os voltados a situações de saúde, sustentabilidade, das implicações da tecnologia no mundo do trabalho, entre outros, mobilizando e articulando conceitos, procedimentos e linguagens próprios da Matemática.

3.2 Recursos

Os recursos necessários para a realização da oficina seriam o laboratório ofertado pela Escola de Aplicação, data show, kits de placas Arduino juntamente com LEDs, *jumpers*, *buzzers piezo*, sensores de distância ultrassônicos, resistores e placas de ensaio. Além de computadores conectados à internet que sejam compatíveis aos *softwares* do Arduino e *Autodesk Tinkercad*.

3.3 Metodologia

Baseada nos princípios da cultura *maker*, a oficina pretende utilizar uma abordagem diferente onde os participantes explorarão as ferramentas dispostas e aprenderão as funcionalidades do Arduíno através de pequenos projetos e situações-problemas que serão apresentados, motivados pelo emblema do “faça você mesmo”, eles assimilarão os conceitos na prática, conforme as necessidades forem surgindo e buscando informações para solucionar os projetos, esse ambiente de aprendizagem será ofertado para que os alunos busquem o protagonismo no processo de aprendizagem. Todas essas atividades e problemas apresentados visam estabelecer conexões da matemática com diferentes áreas do conhecimento como física e arte.

3.4 Conteúdo Programático

O conteúdo programático possui os seguintes tópicos:

- Apresentação do Arduíno;
- Apresentação do *Autodesk Tinkercad*;
- Linguagem de programação do Arduíno;
- Apresentação de módulos, sensores e componentes;
- Resistores e leitura de resistência;
- LEDs e cálculo de resistência;
- Apresentação de duas situações-problemas para desenvolvimento de projetos.

3.5 Atividades Propostas

Antes de iniciar as atividades, serão apresentados aos alunos os componentes físicos que compõem o Arduíno de maneira breve (o *hardware*, o *software*, *protoboard* e alguns sensores, para que em seguida eles possam explorar sozinhos o *software Autodesk Tinkercad*, observando também de maneira virtual os componentes eletrônicos que podem ser utilizados na criação de projetos, tudo isso através da aba “aprenda” disponível no *Tinkercad*.

Exemplo 1: como acender um LED

Componentes

Tabela 1 – Recursos necessários para a simulação do Exemplo 1.

Componentes (versão virtual)	Componentes (versão física)
Computador conectado a internet para utilização do <i>Tinkercad</i> com o propósito de realizar a simulação do circuito	Computador com o <i>software</i> Arduino
	Placa Arduino UNO R3, <i>protoboard</i> , <i>jumppers</i> , 1 LED vermelho e 1 resistor de 150 Ω .

Fonte: produzido pela autora.

Nota: quantidade necessária para cada aluno.

Objetivo

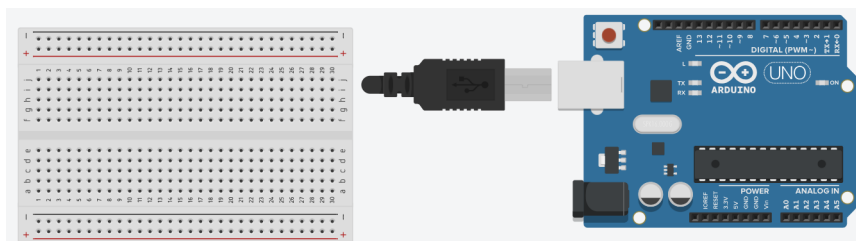
Fazer com que os alunos aprendam na prática a montar circuitos simples de maneira virtual e física, iniciando com a utilização de um LED. Fazendo assim com que sozinhos, eles conheçam melhor o Arduino e o *Tinkercad*.

Passo a passo

Será proposto aos alunos a montagem do circuito virtual no *Tinkercad*, onde eles precisarão posicionar o LED e o resistor na *protoboard* da maneira que acharem melhor, em seguida conectarão a *protoboard* a placa e o LED a uma porta digital do Arduino.

Para a montagem da simulação, será necessário primeiro selecionar a placa do Arduino UNO R3 e a *protoboard* e arrastá-las até a área de trabalho:

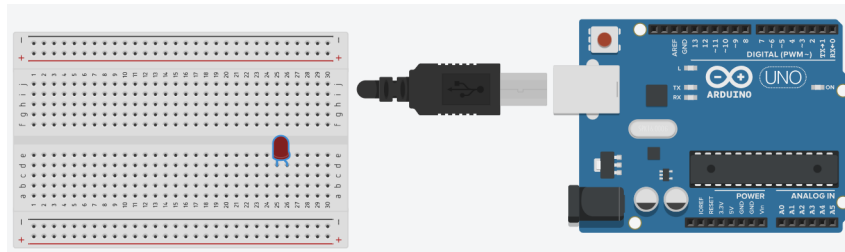
Figura 13 – Placa Arduino e *Protoboard*



Fonte: autora.

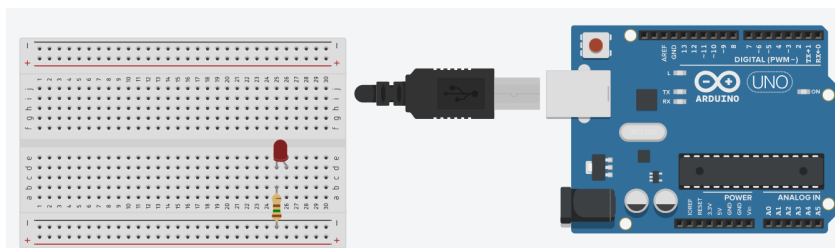
Em seguida será selecionado o LED e será necessário que os alunos calculem a resistência desse LED para saber qual resistor escolher, através da fórmula:

$$R = \frac{(V_{cc} - V_{led})}{I_{led}}$$

Figura 14 – Placa Arduino, *Protoboard* e LED

Fonte: autora.

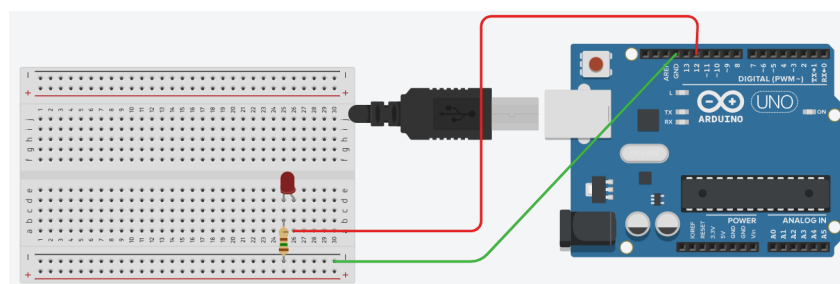
No caso desta simulação a cor escolhida do LED foi vermelho e a sua resistência é de 150Ω . Vejamos então a montagem quase finalizada com o resistor posicionado na perna catódica do LED e ligando a coluna da *protoboard* que será energizada através do GND do Arduino:

Figura 15 – Placa Arduino, *Protoboard*, LED e Resistor

Fonte: autora.

Para finalizar a montagem do circuito, resta conectar a *protoboard* à porta GND do Arduino e o LED à porta digital, neste caso foi escolhida a porta 12 da placa:

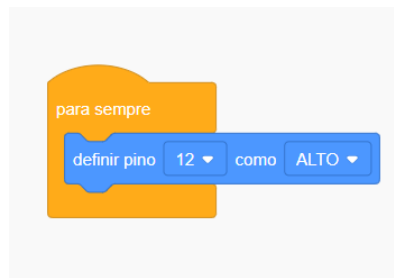
Figura 16 – Circuito do LED finalizado



Fonte: autora.

E essa será a visão final da construção eletrônica para acender um LED, fazendo assim com que os alunos aprendam na prática como montar um circuito simples, trabalhando conceitos de física e aprendendo a realizar o cálculo de resistência de um LED. Para avançar para a parte da programação, ao abrir a aba do *Tinkercad* onde ficam os blocos, o próprio *software* já terá feito sozinho uma programação base, que será a seguinte:

Figura 17 – Circuito do LED programação em blocos



Fonte: autora.

O *Tinkercad* entenderá que você quer que o LED fique aceso sem condição de parada, então definirá a porta no qual o LED está conectado como alta (*HIGH*) e isso implica que o LED permanecerá aceso. Ao baixar o arquivo para transformar em código fonte aceito pela IDE do Arduino, teremos a seguinte programação:

Figura 18 – Circuito do LED programação

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(12, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(12, HIGH);
11 }
12
```

Fonte: autora.

Dentro do comando *void setup()* definimos o pino 12 como pino de saída e dentro do comando *void loop()* que condiciona os comandos que estão dentro a se repetirem em *loop* até que haja condição de parada, a porta digital 12 foi definida como *HIGH*, fazendo com que o pino forneça ou drene corrente para o LED, acendendo ele.

Exemplo 2: como piscar um LED

Componentes

Tabela 2 – Recursos necessários para a simulação do Exemplo 2.

Componentes (versão virtual)	Componentes (versão física)
Computador conectado a internet para utilização do <i>Tinkercad</i> com o propósito de realizar a simulação do circuito	Computador com o <i>software</i> Arduino
	Placa Arduino UNO R3, <i>protoboard</i> , <i>jumpers</i> , 1 LED vermelho e 1 resistor de 150 Ω .

Fonte: produzido pela autora.

Nota: quantidade necessária para cada aluno.

Objetivo

Fazer com que os alunos aprendam na prática a montar circuitos simples de maneira virtual e física, iniciando com a utilização de um LED. Fazendo assim com que sozinhos, eles conheçam melhor o Arduino e o *Tinkercad*.

Passo a passo

Para fazer então com que um LED pisque, a montagem do circuito precisa dos mesmos componentes e a disposição deles continua a mesma da figura 16, o que diferencia são apenas algumas condições na programação, vejamos:

Figura 19 – LED piscando programação em blocos



Fonte: autora.

Os participantes precisam entender que o Arduino só realizará o que lhe for pedido, então, para fazer com que um LED pisque, precisamos acendê-lo e apagá-lo. Na atividade anterior eles

acenderão o LED, nesta, eles precisarão além de fazer o LED acender, fazer com que ele apague e mantenha o ciclo se repetindo até que haja uma condição de parada.

Ao baixar para o código fonte do Arduino, podemos analisar melhor que quando adicionamos os blocos de tempo, pedimos para que dentro do comando *void loop()* a porta digital 12 esteja definida através do *digitalWrite()* em estado *HIGH* (alta) fornecendo corrente para acender o LED, aguarde 1 segundo e seja definida novamente através do *digitalWrite()* ao estado *LOW* (baixa), apagando o LED, aguarde mais 1 segundo e tudo se repita novamente, por conta da condição de *loop* do comando principal, fazendo assim com que esse LED pisque.

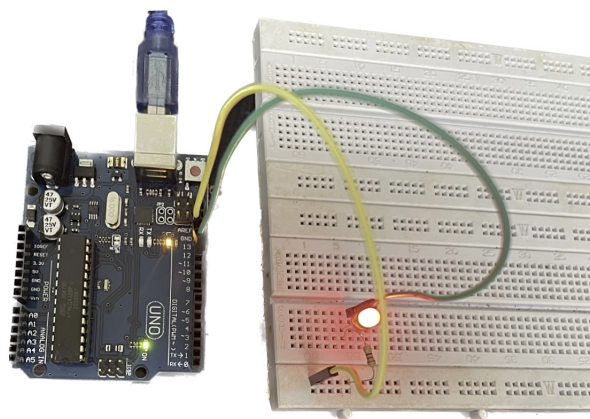
Figura 20 – LED piscando programação

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(12, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(12, HIGH);
11  delay(1000); // wait for 1000 millisecond(s)
12  digitalWrite(12, LOW);
13  delay(1000); // wait for 1000 millisecond(s)
14 }
15
```

Fonte: autora.

Vejamos a seguir o circuito eletrônico físico dos Exemplos 1 e 2:

Figura 21 – Circuito físico LED



Fonte: autora.

Figura 22 – QR Code Vídeo do LED aceso



Fonte: autora.

Link: <https://www.youtube.com/watch?v=_5oHFE6XEI>

Figura 23 – QR Code Vídeo do LED piscando



Fonte: autora.

Link: <<https://www.youtube.com/watch?v=SEMowiyl6sQ>>

Exemplo 3: medindo a distância com Sensor Ultrassônico

Componentes

Tabela 3 – Recursos necessários para a simulação do Exemplo 3.

Componentes (versão virtual)	Componentes (versão física)
Computador conectado a internet para utilização do <i>Tinkercad</i> com o propósito de realizar a simulação do circuito	Computador com o <i>software</i> Arduíno
	Placa Arduíno UNO R3, <i>protoboard</i> , <i>jum-pers</i> , Sensor de distância Ultrassônico HC-SR04

Fonte: produzido pela autora.

Nota: quantidade necessária para cada aluno.

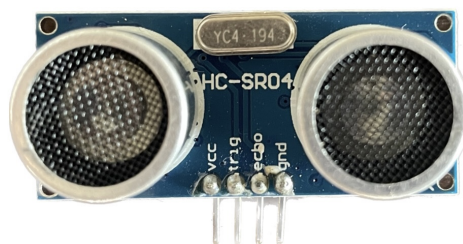
Objetivo

Fazer com que os alunos aprendam na prática a funcionalidade do Sensor de distância Ultrassônico, conhecendo o módulo fisicamente e montando o circuito eletrônico de maneira virtual e física.

Passo a passo

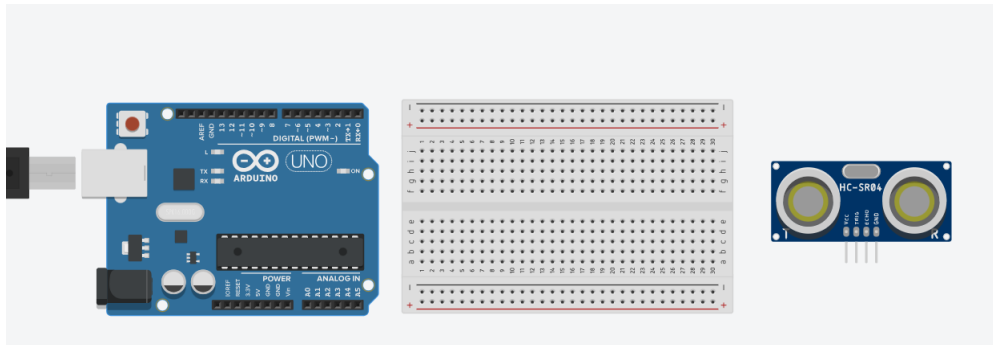
O módulo de Sensor de distância Ultrassônico HC-SR04, que permite medir distâncias através de ondas sonoras possui 4 pinos, no lado direito do *hardware* podemos encontrar o emissor, que é por onde a onda sonora é transmitida, ao chegar no objeto ela é refletida e recebida pelo receptor localizado do lado esquerdo, assim o Arduíno saberá o tempo percorrido dessa onda sonora e através de um cálculo, consegue medir a distância. Para a realização do cálculo e para receber alimentação, esse módulo possui 4 pinos: o VCC para a alimentação de 5V, o TRIG para emissão da onda sonora, o ECHO como receptor de onda sonora e o GND.

Figura 24 – Sensor de Distância Ultrassônico



Fonte: autora.

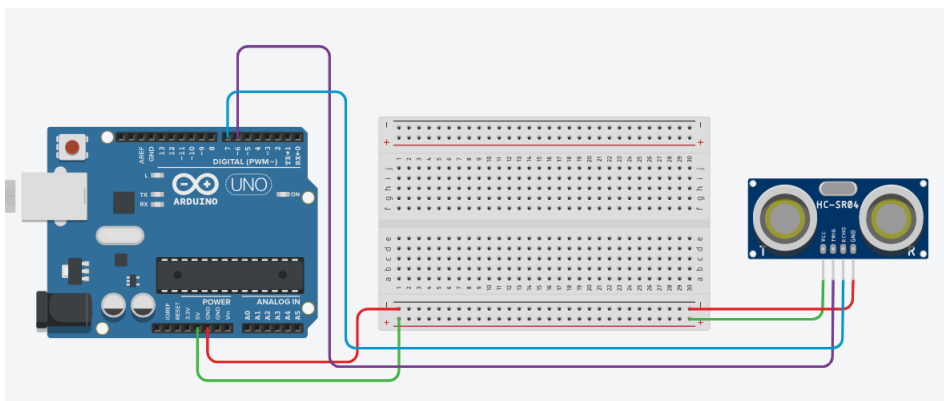
Para a montagem do circuito virtual será necessário primeiro arrastar até a área de trabalho do *Tinkercad* sensor HC-SR04, a placa Arduíno UNO R3 e a *protoboard*:

Figura 25 – Sensor HC-SR04, Arduino e *protoboard*

Fonte: autora.

Já tendo o conhecimento dos pinos do sensor, fica mais fácil realizar a ligação dele com a *protoboard* e com o Arduino. O pino GND do sensor é ligado à protoboard na fileira energizada pelo pino GND do Arduino e o pino VCC do sensor é ligado à fileira da *protoboard* energizada pelo pino 5V do Arduino. Já os pinos ECHO E TRIG podem ser ligados a portas digitais, as portas escolhidas foram 7 e 6 respectivamente. Vejamos:

Figura 26 – Circuito Sensor HC-SR04



Fonte: autora.

Para a programação dessa atividade, será proposto aos alunos que trabalhem direto com a programação em texto, por conta de ser uma atividade com um nível de dificuldade maior devido ao Sensor de distância, mas eles farão conforme acharem conveniente. Abaixo um exemplo de programação em texto:

Figura 27 – Circuito Sensor HC-SR04 programação em texto

```
1  #define echo 7 // definindo porta do echo6
2  #define trig 6 // definindo porta do trig
3
4  unsigned long tempo; // declarando a variável tempo que irá armazenar a duração do pulso do som
5  double distancia; // criando a variável distancia
6
7
8  void setup() {
9  pinMode(trig, OUTPUT); // configurando pino de saída
10 pinMode(echo, INPUT); // configurando pino de entrada
11
12
13  Serial.begin(9600);
14
15  digitalWrite(trig, LOW); // definindo trig como baixo
16  }
17
18  void loop(){
19  digitalWrite(trig, HIGH); // definindo trig como alto
20  delayMicroseconds(10); // espera dez milésimos
21  digitalWrite(trig, LOW); // definindo trig como baixo
22
23  tempo = pulseIn(echo, HIGH);
24  distancia = tempo/58;
25
26  Serial.println(distancia);
27  delay(500);
28  }
```

Fonte: autora.

Nas duas primeiras linhas definimos as portas 7 e 6 para serem respectivamente as portas do ECHO e TRIG. Na linha de programação 4 foi declarada a variável tempo através do comando *unsigned long*, que é responsável por armazenar a duração do pulso do som, o *unsigned* armazenará somente os valores positivos e não negativos, o *long* fará com que seja reservado um espaço maior na memória e permite que sejam armazenados números maiores. Já o *double* na linha 5 armazenará a distância.

Dentro do comando *void setup()* definimos o pino referente ao TRIG como pino de saída, já que é o pino que emite as ondas e o pino referente ao ECHO como entrada, já que funciona como receptor das ondas. Foi inicializado também na linha 13 o comando *Serial.begin()* para ler os dados passados pelo módulo através do monitor serial (monitor que mostra os valores da distância). Por último, através do *digitalWrite()* na linha 15, foi definido o pino TRIG em nível lógico baixo, garantindo que ao ligar o Arduino, fique em nível lógico baixo.

Dentro do comando *void loop()*, na linha de comando 19 o pino TRIG é definido com sinal lógico alto, aguarda 10 microssegundos e de novo é definido com sinal lógico baixo, então durante esse tempo, em que o pino TRIG fica em nível lógico alto, vai fazer com que o módulo emita pulsos de ondas sonoras. Em seguida é utilizado a função *pulseIn* que espera o pino ECHO ir para o estado HIGH e começa a temporizar, assim, os pulsos ocorrem no ar enquanto a temporização é realizada, atingindo o objeto e refletindo esse pulso, que viaja até o receptor e ao chegar, o pino vai para o estado LOW (baixo nível lógico), quando a contagem de tempo é encerrada. O valor em microssegundos é retornado e atribuído a variável “tempo”. Então esse

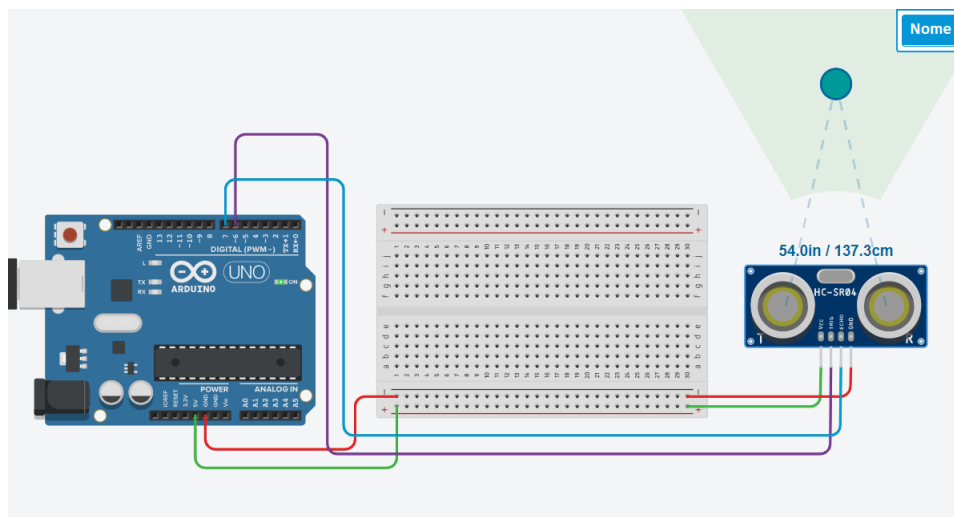
tempo é dividido por 58, sendo um fator de correção da placa para obtermos a distância em centímetros, esse valor é finalmente atribuído a variável “distancia”.

O cálculo que precisa ser realizado para que possamos ter a distância em centímetros será:

$$distancia = \frac{tempo}{58}$$

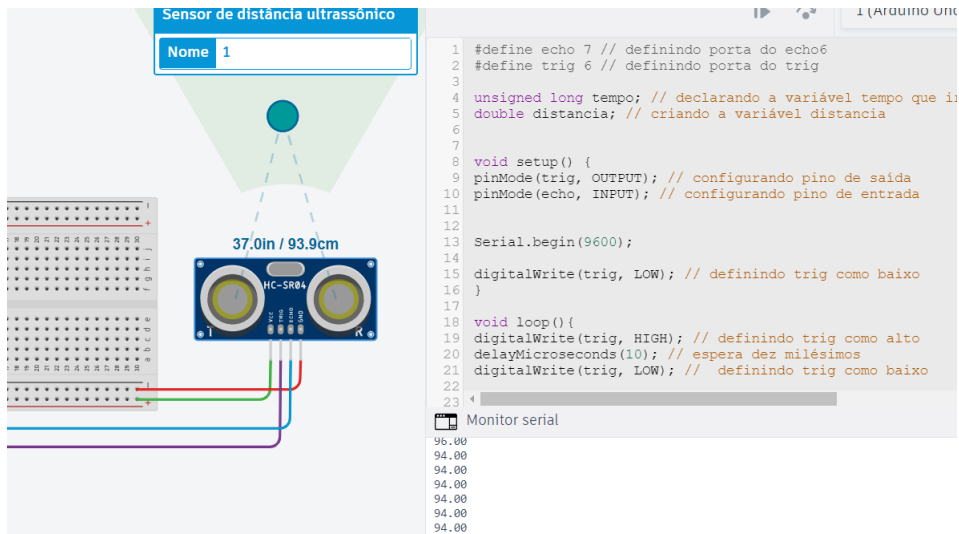
Nas últimas duas linhas de comando, 26 e 27, foram utilizados a *serial.println* por onde os valores da variável “distancia” passam para serem lidos e mostrados no monitor serial, em seguida foi utilizado o *delay()* fazendo com que os valores da “distancia” sejam enviados a cada 500 microssegundos. Vejamos a seguir as imagens do circuito finalizado e funcionando de forma correta:

Figura 28 – Circuito Sensor HC-SR04 finalizado



Fonte: autora.

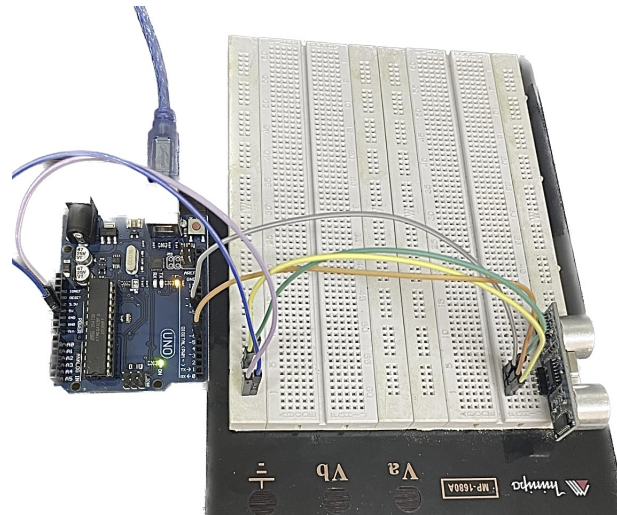
Figura 29 – Circuito Sensor HC-SR04 e Monitor Serial



Fonte: autora.

Vejamos o circuito eletrônico físico da atividade acima:

Figura 30 – Circuito físico Sensor de distância



Fonte: autora.

Figura 31 – QR Code vídeo do Sensor de distância



Fonte: autora.

Link: <<https://www.youtube.com/watch?v=Gur5CQYxIIQ>>

3.5.1 Projeto 1: Sinal de uma via de mão única

Situação-problema

“Em uma via de mão única, há um constante fluxo de carros, fazendo assim com que haja dificuldade dos pedestres atravessarem a rua, podendo até ocorrer acidentes. Faz-se necessária a criação de semáforos para obter controle do fluxo de pedestres e veículos, tornando então, seguro o tráfego dos carros e a travessia dos pedestres.”

Componentes

Tabela 4 – Recursos necessários para a simulação do Projeto 1.

Componentes (versão virtual)	Componentes (versão física)
Computador conectado a internet para utilização do <i>Tinkercad</i> com o propósito de realizar a simulação do circuito	Computador com o <i>software</i> Arduíno
	Placa Arduíno UNO R3, <i>protoboard</i> , <i>jumpers</i> , 2 LEDs vermelhos, 2 LEDs verdes, 1 LED amarelo e 5 resistores sendo 3 de 150Ω, 2 de 125Ω

Fonte: produzido pela autora.

Nota: quantidade necessária para cada equipe.

Objetivos

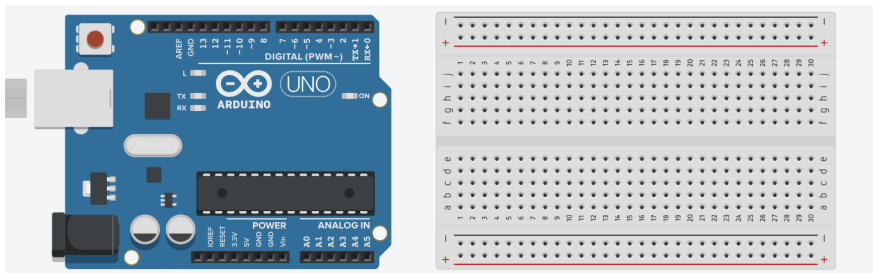
Essa atividade vai ser desenvolvida com o objetivo de construir dois semáforos que permitam solucionar o problema proposto, trabalhando de forma interdisciplinar conceitos matemáticos como funções periódicas e físicos como cálculo da resistência de LEDs, além de operarem através do pensamento computacional e trabalho em equipe.

Passo a passo

A priori será pedido aos alunos que montem o circuito de maneira virtual, através do *Tinkercad*. Será exposto um caminho possível sem dar respostas diretas, deixando com que eles possam tomar frente do que seria melhor para a resolução do problema em suas próprias visões, como eles já terão realizado atividades de acender e piscar LED, a montagem do circuito com LEDs não será novidade.

Eles terão que posicionar os LEDs na *protoboard*, calculando a resistência de cada um para poder escolher os resistores e por fim conectar tanto a *protoboard* quanto os LEDs à placa do Arduino. Para iniciar, terão que seleccionar os seguintes componentes e arrastá-los até a área de trabalho:

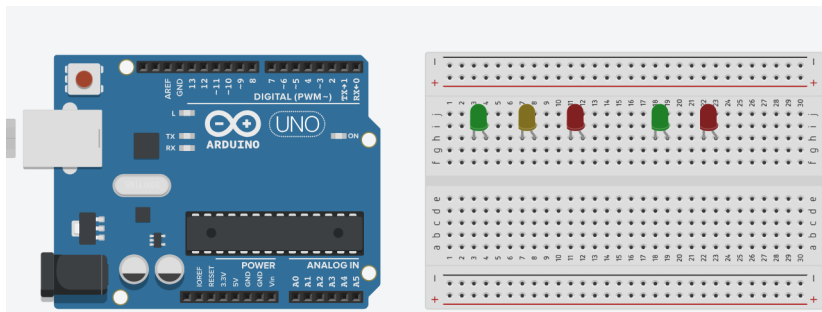
Figura 32 – Arduino e *protoboard*



Fonte: autora.

Em seguida, os alunos precisarão posicionar os 5 LEDs, 3 referentes ao sinal de carros e 2 referentes ao sinal de pedestres:

Figura 33 – Arduino, *protoboard* e LEDs



Fonte: autora.

Para utilizar os resistores será necessário calcular a resistência de cada LED através da fórmula dada na seção 2.1.3 com auxílio da tabela 7:

$$R = \frac{(V_{cc} - V_{led})}{I_{led}}$$

Como para os projetos que serão solicitados precisaremos dos LEDs nas cores amarelo, vermelho e verde, será pedido aos alunos o cálculo da resistência destes três. Que serão os seguintes de acordo com a tabela 7 e considerando a tensão de 5V:

Valor de um resistor para ligar um LED vermelho a pino com tensão de 5V:

$$R = \frac{(V_{cc} - V_{led})}{I_{led}}$$

$$R = \frac{5 - 2}{0,02}$$

$$R = 150\Omega$$

Valor de um resistor para ligar um LED verde a um pino com tensão de 5V:

$$R = \frac{(V_{cc} - V_{led})}{I_{led}}$$

$$R = \frac{5 - 2,5}{0,02}$$

$$R = 125\Omega$$

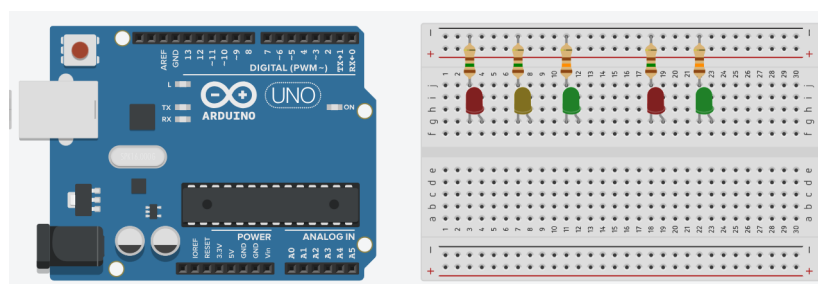
Valor de um resistor para ligar um LED amarelo a um pino com tensão de 5V:

$$R = \frac{(V_{cc} - V_{led})}{I_{led}}$$

$$R = \frac{5 - 2}{0,02}$$

$$R = 150\Omega$$

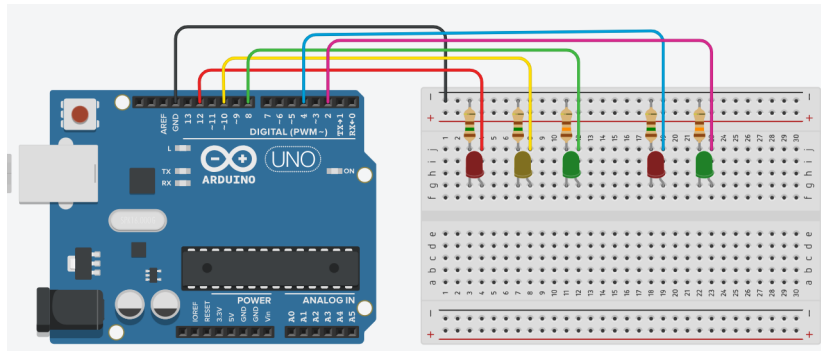
Figura 34 – Arduino, protoboard, LEDs e jumpers



Fonte: autora.

Após posicionar os resistores na perna catódica dos LEDs e na coluna que será energizada com a porta GND do Arduino, pode-se finalizar o circuito ligando o ânodo de cada LED a portas digitais da placa, neste caso foram escolhidas as portas digitais 2, 3, 8, 10 e 12:

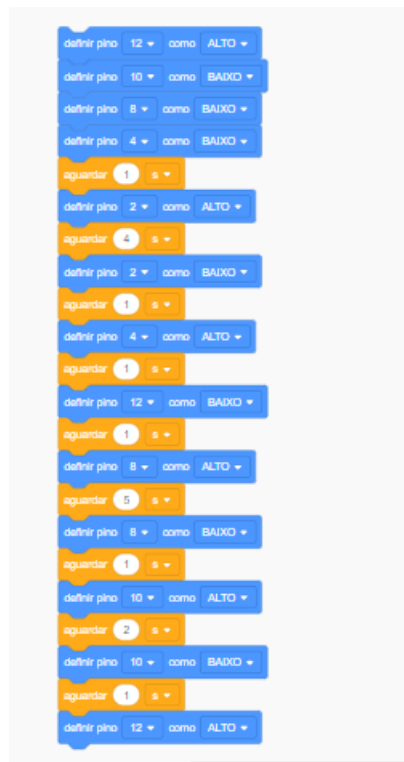
Figura 35 – Simulação do circuito dos Sinais finalizado



Fonte: autora.

Nesta atividade, será pedido aos alunos a programação em blocos, que é mais intuitiva do que na linguagem de programação C\C++, a figura 36 mostra o exemplo de uma programação em blocos que resolve a situação proposta aos alunos.

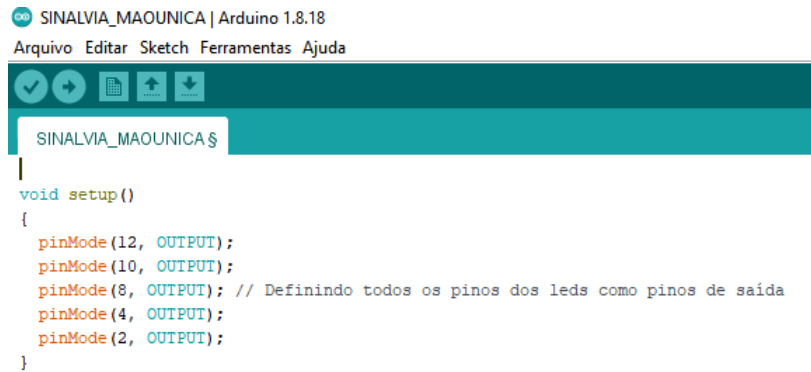
Figura 36 – Programação em Blocos do Circuito dos Sinais



Fonte: autora.

Ao baixar o arquivo na linguagem aceita pela IDE do Arduíno (o *Tinkercad* oferece essa opção), teremos a seguinte programação:

Figura 37 – Programação na Linguagem C\C++

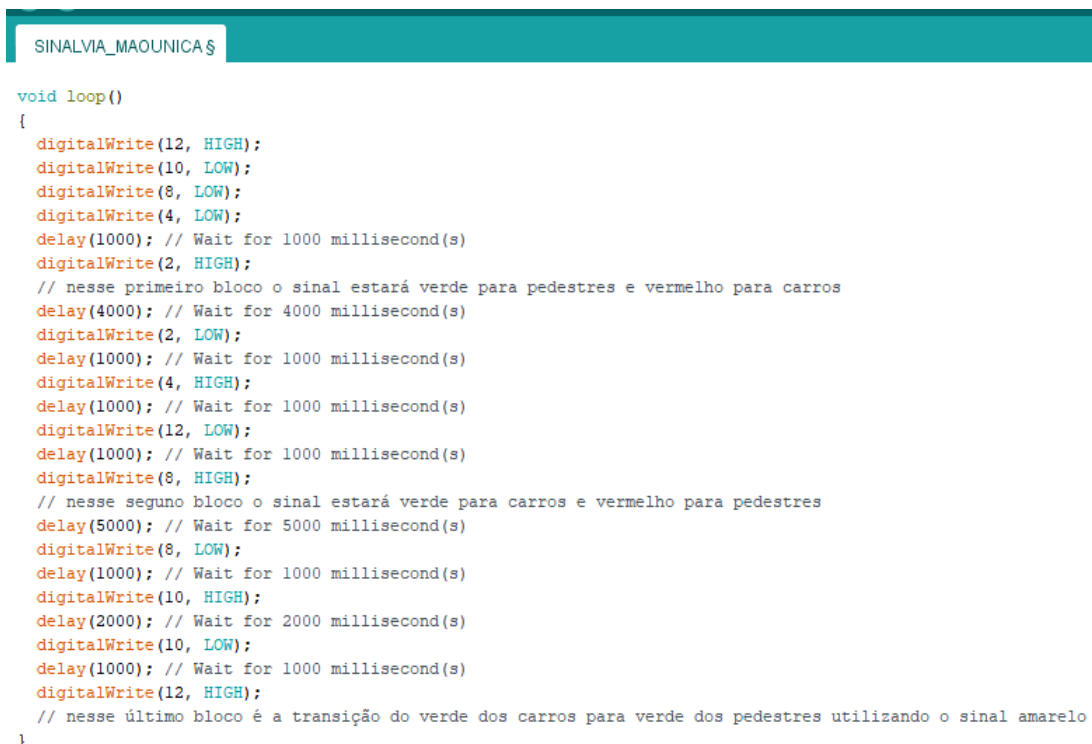


```
SINALVIA_MAOUNICA | Arduino 1.8.18
Arquivo Editar Sketch Ferramentas Ajuda
SINALVIA_MAOUNICA$
|
void setup()
{
  pinMode(12, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(8, OUTPUT); // Definindo todos os pinos dos leds como pinos de saída
  pinMode(4, OUTPUT);
  pinMode(2, OUTPUT);
}
```

Fonte: autora.

Na figura acima, no comando *Void setup()* estaremos definindo todos os pinos nos quais conectamos os LEDs como pinos de saída, fazendo assim com que os pinos forneçam ou drenem corrente para os LEDs.

Figura 38 – Programação na Linguagem C\C++



```
SINALVIA_MAOUNICA$
void loop()
{
  digitalWrite(12, HIGH);
  digitalWrite(10, LOW);
  digitalWrite(8, LOW);
  digitalWrite(4, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(2, HIGH);
  // nesse primeiro bloco o sinal estará verde para pedestres e vermelho para carros
  delay(4000); // Wait for 4000 millisecond(s)
  digitalWrite(2, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(4, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(12, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(8, HIGH);
  // nesse segundo bloco o sinal estará verde para carros e vermelho para pedestres
  delay(5000); // Wait for 5000 millisecond(s)
  digitalWrite(8, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(10, HIGH);
  delay(2000); // Wait for 2000 millisecond(s)
  digitalWrite(10, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(12, HIGH);
  // nesse último bloco é a transição do verde dos carros para verde dos pedestres utilizando o sinal amarelo
}
```

Fonte: autora.

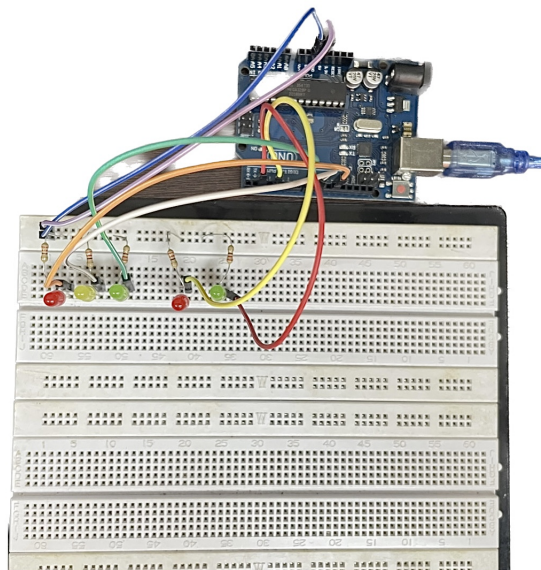
Já no comando *Void loop()* que é onde fica a parte principal da nossa programação, tudo que está dentro desse comando se repete em *loop* até que haja condição de parada (neste caso, não há). Como pode-se observar na figura 38 o primeiro bloco de comandos, define o pino digital 12 e 2 como altos e os pinos digitais 10, 8 e 4 como baixos, mostrando assim que o sinal está verde para pedestres e vermelho para os carros. O pino 12 se refere a um LED vermelho e o 2 a um LED verde, o pino 12 acende primeiro e o 2 com o *delay* de 1 segundo e o LED 2 fica aceso por 4 segundos.

No segundo bloco de comandos mostra a transição do sinal verde de pedestres, para o sinal verde dos veículos. A priori, após os 4 segundos de aceso do LED conectado ao pino digital 2, ele é definido como baixo, aguarda 1 segundo e o pino digital 4 que é o LED vermelho dos pedestres é definido como alto, aguarda 1 segundo, o LED conectado ao pino digital 12 que é o vermelho dos carros é definido como baixo, aguarda mais 1 segundo e o LED conectado ao pino digital 8 que é o verde dos carros é definido como alto, liberando a passagem dos veículos por 5 segundos.

Já no terceiro e último bloco de comandos, é mostrado a transição do sinal verde dos carros para o verde dos pedestres, o pino digital 8 é definido como baixo, aguarda 1 segundo e o pino digital 10 onde o LED amarelo está conectado é definido como alto para alertar os carros de que o sinal irá fechar logo, após 2 segundos, o pino 10 é definido como baixo, após 1 segundo o pino 12 que refere-se ao vermelho dos carros é definido como alto, finalizando assim as linhas de programação e voltando a realizar tudo o que está dentro do *Void loop()* de maneira infinita.

Vejamos o circuito eletrônico físico da atividade acima:

Figura 39 – Circuito físico sinal de uma via de mão única



Fonte: autora.

Figura 40 – QR Code vídeos do Sina de uma via de mão única



Fonte: autora.

Link: <<https://www.youtube.com/watch?v=Oan5EmVQA8c>>

Questionamentos adicionais

- Seria possível construir um sinal para uma via de mão dupla?
- Se sim, quais mudanças no algoritmo e no circuito precisaríamos fazer?

3.5.2 Projeto 2: estacionamento inteligente

Situação-problema

“Em um estacionamento de um Shopping no qual quase todas as vagas estão ocupadas, um determinado casal desejaria saber onde poderá encontrar uma vaga para estacionar seu carro com mais facilidade, com isso, vamos construir um estacionamento inteligente que mostre quando a vaga está ocupada e quando está vazia através de um sensor”

Componentes

Tabela 5 – Recursos necessários para a simulação do Projeto 2.

Componentes (versão virtual)	Componentes (versão física)
Computador conectado a internet para utilização do <i>Tinkercad</i> com o propósito de realizar a simulação do circuito	Computador com o <i>software</i> Arduíno
	Placa arduíno UNO R3, <i>protoboard</i> , <i>jumpers</i> , Sensor de distância Ultrassônico HC-SR04, 1 LED vermelho, 1 LED verde, e 2 resistores sendo 1 de 150 Ω , 1 de 125 Ω

Fonte: produzido pela autora.

Nota: quantidade necessária para cada equipe.

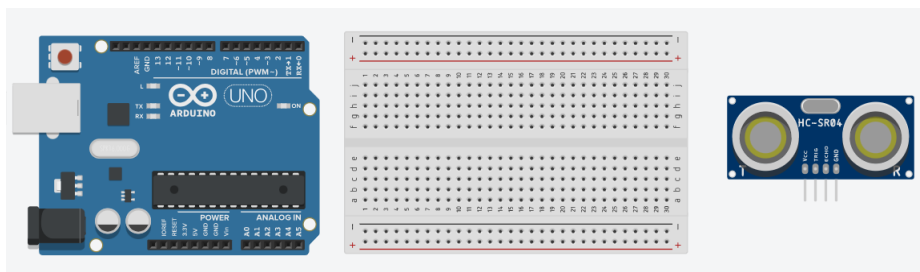
Objetivo

Essa atividade vai ser desenvolvida com o objetivo de construir um estacionamento inteligente que mostre aos passageiros dos veículos quais vagas estão ocupadas e quais vagas estão vazias, através de LEDs nas cores vermelho e verde, solucionando assim o problema proposto e tornando mais prático o estacionamento. Serão trabalhados conceitos matemáticos como mudanças de unidades, cálculo da distância e conceitos físicos como o estudo de ondas sonoras e cálculo de resistência. Novamente operando através do pensamento computacional e trabalho em equipe.

Passo a passo

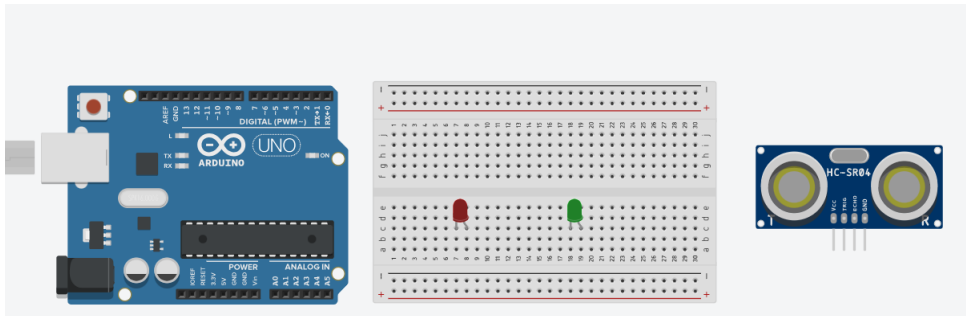
A segunda situação-problema que será trabalhada apresentará um pouco mais de dificuldade que a primeira, pois envolverá o módulo de sensor de distância ultrassônico. Como o módulo já foi mostrado anteriormente e já foi construído o circuito e programado o sensor, os alunos saberão como ele pode ser utilizado e para quê ele serve. Será proposto a ideia de ligá-lo direto à *protoboard* e em seguida eles precisarão utilizar os LEDs e conectar ambos a placa do Arduino. Começamos selecionando os componentes:

Figura 41 – Sensor HC-SR04, Arduino e *protoboard*



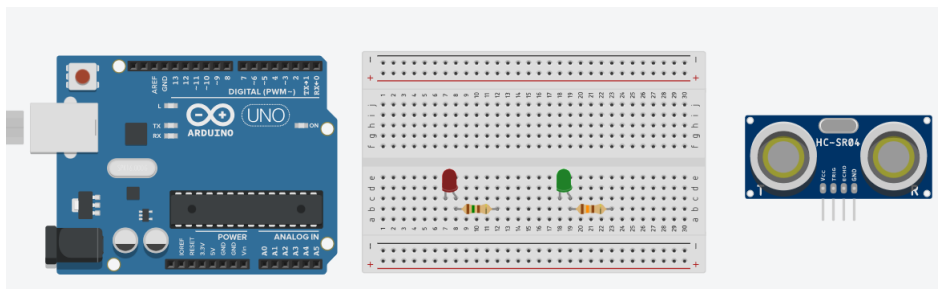
Fonte: autora.

Em seguida serão selecionados dois LEDs, um vermelho e um verde, que farão o papel de identificar quando a vaga do estacionamento está ocupada ou vazia. Vejamos:

Figura 42 – Sensor HC-SR04, Arduíno, *protoboard* e LEDs

Fonte: autora.

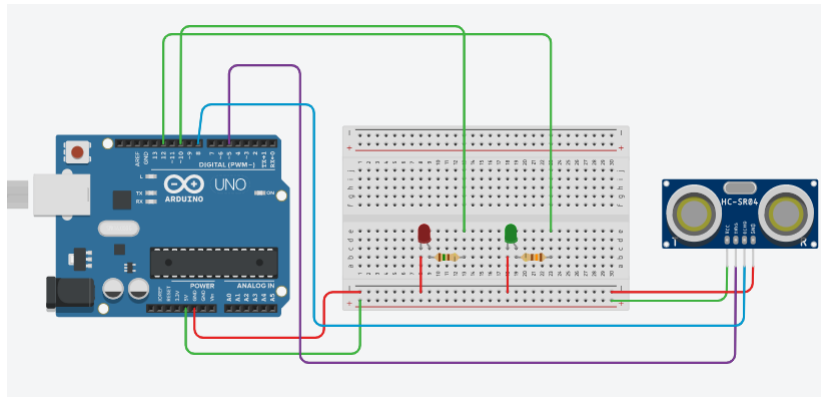
Na atividade anterior os alunos já calcularam a resistência dos LEDs vermelho e verde, que serão respectivamente $150\ \Omega$ e $125\ \Omega$, então basta selecioná-los e posicioná-los na *protoboard*.

Figura 43 – Sensor HC-SR04, Arduíno, *protoboard*, LEDs e resistores

Fonte: autora.

Para finalizar o circuito, foi realizada a ligação dos componentes, os pinos ECHO e TRIG foram ligados às portas digitais 8 e 5 respectivamente e os LEDs às portas digitais 10 e 12. O pino VCC do sensor foi ligado à coluna da *protoboard* energizada pela porta de 5V e o pino GND foi ligado à coluna energizada pela porta GND do Arduíno. Podemos observar a seguir:

Figura 44 – Circuito Estacionamento Inteligente



Fonte: autora.

Diferente da atividade anterior, neste projeto a programação foi feita diretamente na linguagem C\C++ e assim será proposto aos alunos, para eles também se familiarizarem programando em texto e não somente em blocos, trabalhando ainda mais o pensamento computacional. Quando abrimos a aba “código” no *Tinkercad* podemos escolher entre três opções: “blocos”, “blocos+texto” e “texto”, será dada a sugestão da programação em “texto” mas a escolha se dá entre os próprios alunos e suas equipes. Abaixo podemos observar um modelo de programação para o problema que foi colocado:

Figura 45 – Programação Estacionamento Inteligente

```

1 #define echo 8 // definindo porta do echo
2 #define trig 5 // definindo porta do trig
3
4 unsigned long tempo; // declarando a variável tempo que irá arma
5 double distancia; // criando a variável distancia
6
7
8 void setup() {
9   pinMode(trig, OUTPUT); // configurando pino de saída
10  pinMode(echo, INPUT); // configurando pino de entrada
11  pinMode(10, OUTPUT); // configurando pino de saída do led vermel
12  pinMode(12, OUTPUT); // configurando pino de saída do led verde
13
14  Serial.begin(9600);
15
16  digitalWrite(trig, LOW); // definindo trig como baixo
17 }

```

Fonte: autora.

Assim como na atividade da seção 3.5, a programação para o sensor será realizada da mesma maneira que no exemplo 3, incluindo então somente a programação dos dois LEDs. Nas duas primeiras linhas definimos as portas 8 e 5 para serem respectivamente as portas do ECHO e

TRIG. Na linha de programação 4 foi declarada a variável tempo através do comando *unsigned long*, que é responsável por armazenar a duração do pulso do som, o *unsigned* armazenará somente os valores positivos e não negativos, o *long* fará com que seja reservado um espaço maior na memória e permite que sejam armazenados números maiores. Já o *double* armazenará a distância.

Dentro do comando *void setup()* definimos o pino referente ao TRIG como pino de saída e o pino referente ao ECHO como entrada, além de também definir os pinos 10 e 12 referente aos LEDs como pinos de saída. Foi inicializado também na linha 14 o comando *Serial.begin()* para ler os dados passados pelo módulo através do monitor serial, por último através do *digitalWrite()*, foi definido o pino TRIG em nível lógico baixo.

Figura 46 – Programação Estacionamento Inteligente

```
19 void loop(){
20 digitalWrite(trig, HIGH); // definindo trig como alto
21 delayMicroseconds(10); // espera dez milésimos
22 digitalWrite(trig, LOW); // definindo trig como baixo
23
24 tempo = pulseIn(echo, HIGH);
25 distancia = tempo/58;
26
27 if (distancia <= 50) {
28     digitalWrite(10, HIGH); // acender led vermelho
29     digitalWrite(12, LOW); // apagar led verde
30 } else {
31     digitalWrite(10, LOW); // caso contrário, apagar led vermelho
32     digitalWrite(12, HIGH); // acender led verde
33 }
34
35 Serial.println(distancia);
36 delay(500);
37 }
```

Monitor serial

Fonte: autora.

Na figura 46, dentro do comando *void loop()*, na linha de comando 20 o pino TRIG é definido com sinal lógico alto, aguarda 10 microssegundos e de novo é definido com sinal lógico baixo, então durante esse tempo, em que o pino TRIG fica em nível lógico alto, vai fazer com que o módulo emita pulsos de ondas sonoras. Em seguida é utilizado a função *pulseIn* que espera o pino ECHO ir para o estado HIGH e começa a temporizar, assim, os pulsos ocorrem no ar enquanto a temporização é realizada, atingindo o objeto e refletindo esse pulso, que viaja até o receptor e ao chegar, o pino vai para o estado LOW (baixo nível lógico), quando a contagem de tempo é encerrada. O valor em microssegundos é retornado e atribuído a variável “tempo”. Então esse tempo é dividido por 58 para obtermos a distância em centímetros, esse valor é finalmente atribuído a variável “distancia”.

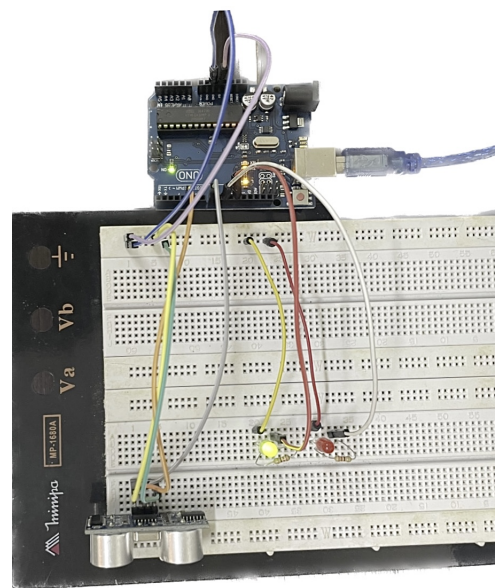
A partir da linha de comando 27 até a linha 32 será trabalhada a função *if(){}else{}* que significa que dado dois eventos *a* e *b* e uma determinada condição, se tal condição for satisfeita,

ocorrerá *a*, se essa condição não acontecer, resultará em *b*. Portanto, se a distância for menor ou igual a 50 centímetros, o pino 10 referente ao LED vermelho irá acender e o pino 12 referente ao LED verde permanecerá apagado. Caso a condição não ocorra e a distância seja maior do que 50 centímetros, então o pino 10 referente ao LED vermelho irá apagar e o pino 12 referente ao LED verde irá acender. Importante lembrar que como esses comandos estão dentro do *void loop()* serão executados até que haja uma condição de parada, que neste caso não tem, então serão executados em *loop*.

Nas últimas duas linhas de comando, 35 e 36, foram utilizados a *serial.println()* por onde os valores da variável “distancia” passam para serem lidos e mostrados no monitor serial, em seguida foi utilizado o *delay()* fazendo com que os valores da “distancia” sejam enviados a cada 500 microssegundos.

Vejamos a seguir o circuito eletrônico físico da atividade acima:

Figura 47 – Circuito físico estacionamento inteligente



Fonte: autora.

Figura 48 – QR Code vídeo do Estacionamento Inteligente



Fonte: autora.

Link: <<https://www.youtube.com/watch?v=JUvHK21fxFM>>

Questionamentos adicionais

- Há alguma ideia de como melhorar o estacionamento inteligente?
- Podemos criar algum protótipo que indique na entrada do estacionamento se ainda há vagas disponíveis e a quantidade disponível? Como?

3.5.3 Projeto 3: sensor de estacionamento

Situação-problema

“Ao localizar uma vaga no estacionamento de um shopping, o motorista possui um pouco de dificuldade para estacionar pois não tem uma boa noção de espaço, para ajudá-lo a estacionar sem bater seu carro, se faz necessário a criação de um sensor que indique através de sinal sonoro e luminoso quando o carro se aproxime demais de outro objeto (outros carros, paredes, pessoas e etc)”.

Componentes

Tabela 6 – Recursos necessários para a simulação do Projeto 3.

Componentes (versão virtual)	Componentes (versão física)
Computador conectado a internet para utilização do <i>Tinkercad</i> com o propósito de realizar a simulação do circuito	Computador com o <i>software</i> Arduíno
	Placa Arduíno UNO R3, <i>proto</i> board, <i>jumper</i> s, 1 LED vermelho, 1 buzzer piezo, 1 Sensor de distância Ultrassônico HC-SR04, 2 resistores sendo 1 de 150 Ω e 1 de 140 Ω (para cada equipe).

Fonte: produzido pela autora.

Nota: quantidade necessária para cada equipe.

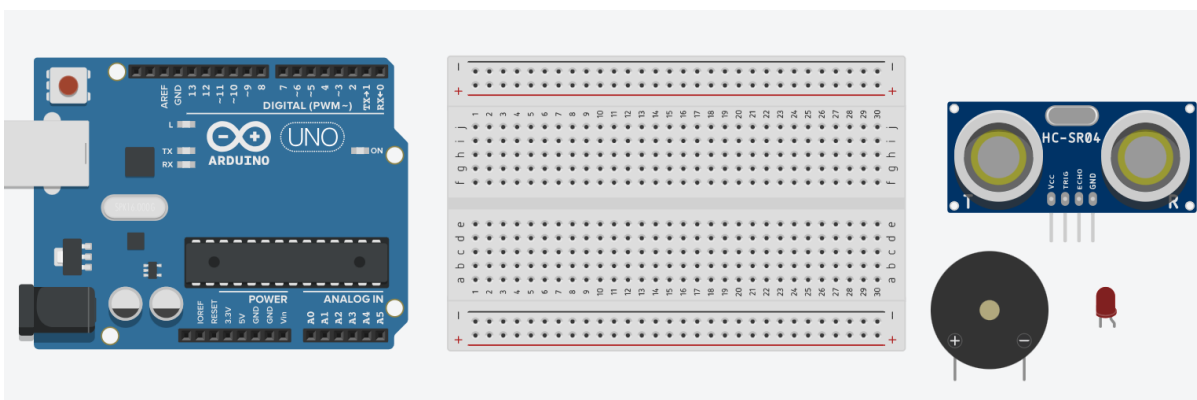
Objetivo

Essa atividade vai ser desenvolvida com o objetivo de construir um sensor de estacionamento que ajude o motorista a estacionar seu carro de maneira segura, avisando através de sinais sonoros e luminosos quando algum objeto está muito próximo do carro. Serão trabalhados conceitos matemáticos como mudanças de unidades, cálculo da distância e conceitos físicos como o estudo de ondas sonoras e cálculo de resistência, além de trabalhar com um novo módulo proporciona efeitos sonoros. Mais uma vez operando através do pensamento computacional e trabalho em equipe.

Passo a Passo

Para a construção deste último projeto, os alunos utilizarão os conhecimentos que já possuem em relação a LEDs e ao Sensor de distância, mas também utilizarão um módulo novo chamado de *Buzzer Piezo*, que é um componente indicado para quem precisa adicionar efeitos sonoros nos circuitos, é uma pastilha de cerâmica especial com dois eletrodos em sua face, que sofre deformações mecânicas quando submetida a um campo elétrico e possui impedância de 140Ω . Para iniciar a montagem do circuito primeiro selecionamos os componentes necessários e trazemos até a área de trabalho:

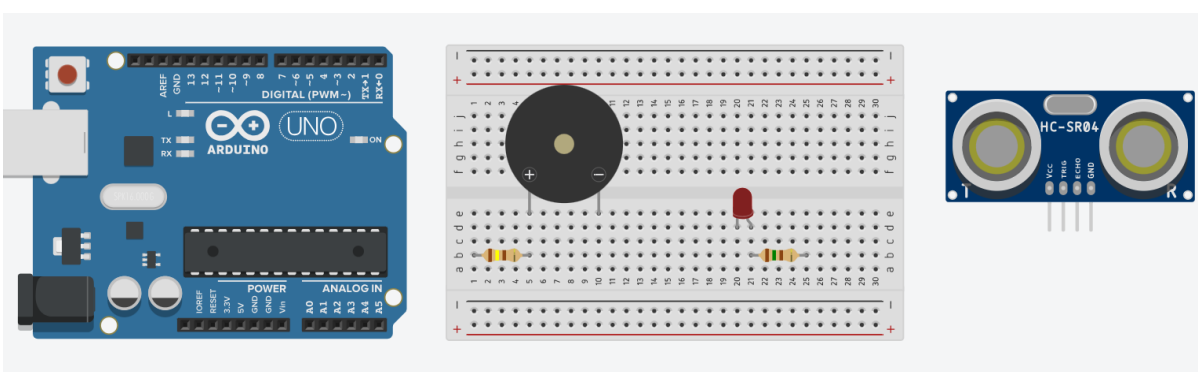
Figura 49 – Arduíno, *protoboard*, LED, Sensor HC-SR04 e *Buzzer Piezo*



Fonte: autora.

A informação do resistor para o *piezo* será fornecida aos alunos, será necessário utilizar um resistor de 140Ω e para o LED vermelho, eles já possuem o valor do resistor que é de 150Ω , então selecionaremos e vamos dispor na *protoboard*. No *piezo* será posicionado o resistor no eletrodo positivo e no LED será posicionado no ânodo, vejamos:

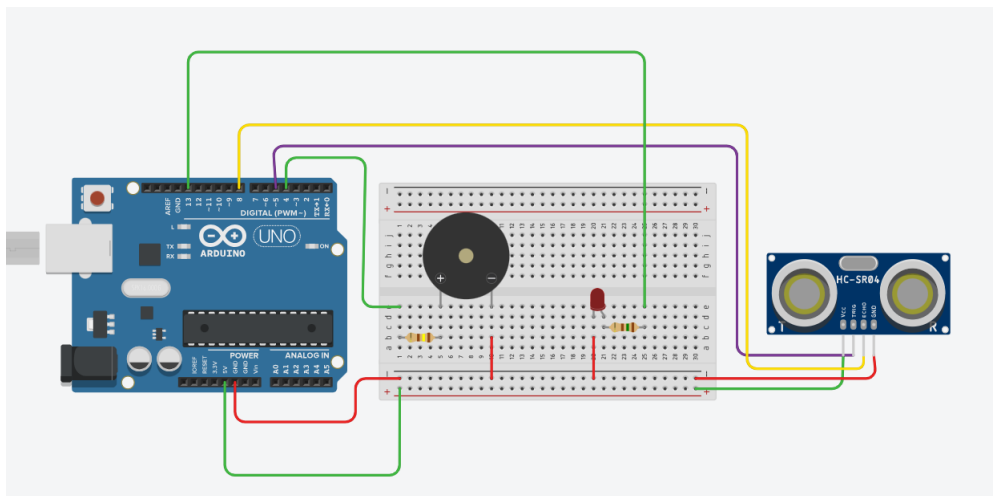
Figura 50 – Arduíno, *protoboard*, LED, Sensor HC-SR04 e *Buzzer Piezo* e resistores



Fonte: autora.

E para finalizar o circuito, vamos realizar as ligações necessárias. O *piezo* terá o eletrodo positivo ligado a porta digital 4 e o eletrodo negativo à coluna da *protoboard* que será energizada pela porta GND do Arduino. O LED terá a perna catódica ligada a coluna energizada pelo GND e o ânodo ligado a porta digital 13 da placa. os pinos ECHO e TRIG do sensor foram ligados às porta digitais 8 e 5 respectivamente, o pino VCC do foi ligado à coluna da *protoboard* energizada pela porta de 5V e o pino GND foi ligado à coluna energizada pela porta GND do Arduino. Podemos observar a seguir:

Figura 51 – Circuito Sensor de Estacionamento finalizado



Fonte: autora.

Para realizar a programação desse circuito, trabalharemos novamente com a programação em texto, muito semelhante ao projeto da seção 3.5.2. A programação para o sensor será exatamente a mesma. Podemos observar:

Figura 52 – Programação Sensor de Estacionamento

```

1  #define echo 8 // definindo porta do echo6
2  #define trig 5 // definindo porta do trig
3
4  unsigned long tempo; // declarando a variável tempo que irá armazenar a duração do pulso do som
5  double distancia; // criando a variável distancia
6
7
8  void setup() {
9  pinMode(trig, OUTPUT); // configurando pino de saída
10 pinMode(echo, INPUT); // configurando pino de entrada
11 pinMode(13, OUTPUT); // configurando pino de saída do led vermelho
12 pinMode(4, OUTPUT); // configurando pino de saída piezzo
13
14 Serial.begin(9600);
15
16 digitalWrite(trig, LOW); // definindo trig como baixo
17 }
--

```

Fonte: autora.

Foram utilizados os mesmos comandos para o funcionamento do sensor e dentro do comando `void setup()` só modificou a definição do LED, que foi declarado através do `pinMode()` na porta 13 como pino de saída e o `piezo` que pelo mesmo comando foi declarado na porta 4 como pino de saída também. Vejamos o segundo bloco de comandos:

Figura 53 – Programação Sensor de Estacionamento

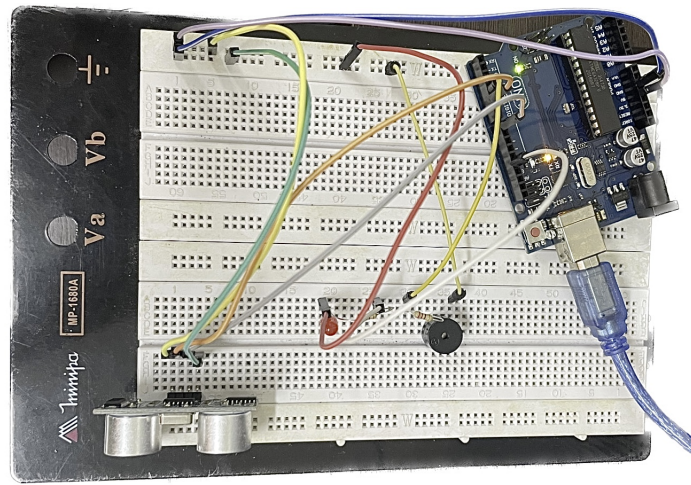
```
19 void loop(){
20   digitalWrite(trig, HIGH); // definindo trig como alto
21   delayMicroseconds(10); // espera dez milésimos
22   digitalWrite(trig, LOW); // definindo trig como baixo
23
24   tempo = pulseIn(echo, HIGH);
25   distancia = tempo/58;
26
27   if (distancia <= 20) {
28     digitalWrite(13, HIGH);
29     digitalWrite(4, HIGH);
30     delay(1000); // Wait for 1000 millisecond(s)
31     digitalWrite(4, LOW);
32     delay(1000); // Wait for 1000 millisecond(s)
33   } else {
34     digitalWrite(13, LOW);
35     digitalWrite(4, LOW);
36   }
37
38   Serial.println(distancia);
39   delay(500);
40 }
```

Fonte: autora.

Dentro do bloco `void loop()` foram utilizados os mesmos comandos para funcionamento do sensor. A partir da linha de comando 27 até a linha 36 será trabalhada a função `if(){}else{}` que significa que dado dois eventos *a* e *b* e uma determinada condição, se tal condição for satisfeita, ocorrerá *a*, se essa condição não acontecer, resultará em *b*. Portanto, se a distância for menor ou igual a 20 centímetros, o pino 13 referente ao LED vermelho é definido como *HIGH* e irá acender, o pino 4 referente ao *piezo* é definido como *HIGH* também, fornecendo corrente para que sejam emitidos sons através do módulo. Caso a condição não ocorra e a distância seja maior do que 20 centímetros, então o pino 13 referente ao LED vermelho será definido como *LOW* e apagar e o pino 4 referente ao *piezo* também será definido como *LOW* fazendo com que não emita efeitos sonoros. Importante lembrar que como esses comandos estão dentro do `void loop()` serão executados até que haja uma condição de parada, que neste caso não tem, então serão executados em *loop*.

Vejamos a seguir o circuito eletrônico físico da atividade acima:

Figura 54 – Circuito físico Sensor de estacionamento



Fonte: autora.

Figura 55 – QR Code vídeo do Sensor de Estacionamento



Fonte: autora.

Link: <https://www.youtube.com/watch?v=5eCeQY_4YqY>

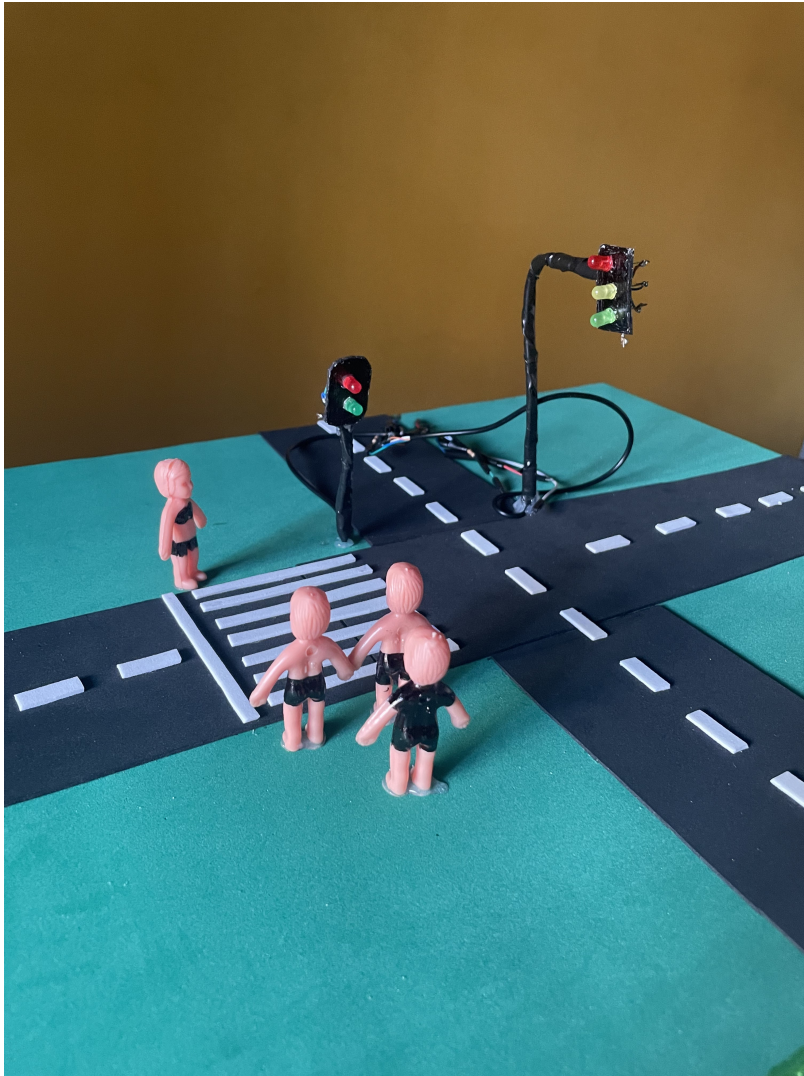
Questionamentos adicionais

- Há algum módulo que podemos utilizar para mostrar para o motorista a distância exata que o carro esta do objeto?
- Podemos criar um protótipo que ao tentar abrir o carro sem desligar o alarme o *buzzer piezzo* dispare?

3.6 Protótipo Físico

Foi escolhido um dos projetos para que fosse construído o protótipo físico. Vejamos a seguir a imagem do protótipo do Projeto 1, descrito na seção 3.5.1:

Figura 56 – Protótipo Projeto 1



Fonte: autora.

Figura 57 – QR Code vídeo do protótipo do Projeto 1



Fonte: autora.

Link: <<https://www.youtube.com/watch?v=CYmQk1zHCQM>>

Considerações Finais

Este trabalho se iniciou com a proposta de promover uma aprendizagem mão na massa em matemática através de atividades de Robótica que tornassem os alunos autores de seu conhecimento e protagonistas em seus processos de aprendizagem. Com o passar dos anos e com a evolução das tecnologias digitais, as metodologias utilizadas para o ensino da matemática acabaram não conseguindo manter os alunos interessados, tornando a ideia de aprender matemática algo fora do comum e exclusivo para pessoas que possuíssem “dom” para a disciplina. O uso das TDIC’s no ensino foi proposto pela BNCC e defendido por inúmeros pesquisadores, em meio a esse cenário de busca por metodologias e abordagens pedagógicas que suprissem as necessidades dos alunos, a Robótica Educacional pode ser vista como uma ferramenta capaz de promover uma aprendizagem baseada nos princípios *makers*, tornando os alunos mais motivados e dispostos a entrar no universo da matemática.

Através da pesquisa realizada pode-se observar o conceito de aprendizagem mão na massa como sendo o processo onde o aluno “aprende fazendo” através de atividades propostas, tudo na prática, um processo onde o aluno criativo é aquele que não só cria como recria ideias, vê o mundo sob sua própria ótica, sendo criativo dentro da sua realidade. E para proporcionar aos alunos essa aprendizagem, foi estudado a possibilidade de utilidade da Robótica Educacional aliada aos princípios da cultura *maker* que incentiva os alunos a construir com suas próprias mãos e da abordagem *STEAM* que proporciona a conexão de diversas áreas do conhecimento, trabalhando com a interdisciplinaridade.

E como alternativa para favorecer a criação de novos ambientes de aprendizagem, este trabalho escolheu o Arduíno, uma placa de prototipagem que permite o desenvolvimento de atividades de RE, proporcionando ao aluno a oportunidade de aprender matemática aliada a outras disciplinas através da construção de robôs, criação de projetos de automação e protótipos. Fazendo então com que esse aluno por meio da montagem de circuitos físicos, ao utilizar a placa, seus sensores e módulos, coloque em prática sua imaginação, aprendendo conceitos de física, eletrônica, matemática, arte, além de desenvolver seu pensamento computacional programando. Também foi estudada a plataforma *Autodesk Tinkercad* que oferece um painel para simular a criação de circuitos de maneira virtual, sendo então uma alternativa para quem não possui o Arduíno físico, mas apresenta interesse para programar e criar protótipos.

Porém, a implementação de atividades da RE não resolve o problema sozinho, é necessária uma abordagem correta, onde o professor saiba organizar situações-problemas e saiba instigar seus alunos a assumirem o papel de protagonistas. Para [Borba e Penteado \(2019\)](#) se faz necessário um espaço para refletir sobre as mudanças que o uso da informática traz, caso contrário, os professores podem acabar não utilizando tais mídias ou utilizem de maneira rasa e pouco

proveitosa, domesticando-as. Logo, ao utilizar a RE para promover uma aprendizagem mão na massa o professor precisa refletir sobre qual a melhor forma de inserir as atividades em sala de aula para não usufruir de maneira errônea, não despertando o interesse dos alunos e consequentemente, não atingindo os objetivos desejados.

Nesse sentido, foi proposto o desenvolvimento de atividades de Robótica através de uma oficina *maker* que possui o objetivo de promover aos alunos o contato com a RE através do Arduíno, por meio de situações-problemas, possibilitando a criação de protótipos e a compreensão de conceitos matemáticos, através da abordagem *STEAM*, trabalhando de forma interdisciplinar e apoiada nos princípios da cultura *maker*. Será um ambiente de aprendizagem onde os alunos assumirão os papéis de protagonistas e autores do conhecimento e o professor será apenas o meio que proporciona condições para que eles consigam alcançar uma aprendizagem onde desenvolvam habilidades na prática.

A partir deste trabalho, afirmamos que é possível aliar a Robótica Educacional à Matemática de modo a obter um processo de ensino e aprendizagem inovador e autônomo, a partir das atividades investigativas, baseadas na cultura *maker* e na abordagem *STEAM*. Percebemos também que a Robótica Educacional pode promover a construção de um ambiente aprendizagem onde os alunos consigam assumir papéis de liderança e que podem desenvolver a criatividade, o espírito de colaboração e sobretudo, sentirem-se mais motivados a aprender matemática e mais inspirados a desenvolver novas soluções para situações e desafios do cotidiano, ou seja, desenvolveram habilidades que a tornarão pessoas mais criativas dentro de suas realidades.

Finalmente, acreditamos que este trabalho pode contribuir para a difusão do movimento *maker*, da abordagem *STEAM* e da Robótica Educacional no âmbito da matemática, assim como, pode ser ampliado em estudos futuros de pós-graduação, nas áreas de ensino e educação matemática.

Referências

APRENDENDO ELÉTRICA. *Tabela de Código de cores para resistores*. 2021. Disponível em: <<https://aprendendoeletrica.com/codigo-de-cores-para-resistores/>>. Acesso em: 20 de maio de 2023. Citado 2 vezes nas páginas 34 e 36.

ARDUÍNO. *Referência de Idioma*. Disponível em: <<https://www.arduino.cc/reference/en/>>. Acesso em: 15 de junho de 2023. Citado na página 81.

BACICH, L.; HOLANDA, L. Steam: integrando as áreas para desenvolver competências. *STEAM em sala de aula: a aprendizagem baseada em projetos integrando conhecimentos na educação básica*, p. 12–22, 2020. Citado na página 25.

BARCELOS, T. S.; SILVEIRA, I. F. Pensamento computacional e educação matemática: Relações para o ensino de computação na educação básica. In: SN. *XX Workshop sobre Educação em Computação, Curitiba. Anais do XXXII CSBC*. [S.l.], 2012. v. 2, p. 23. Citado 2 vezes nas páginas 27 e 28.

BORBA, M. D. C.; PENTEADO, M. G. *Informática e educação matemática*. [S.l.]: Autêntica Editora, 2019. Citado na página 75.

BRASIL. Ministério da educação, base nacional comum curricular. 2018. Citado 3 vezes nas páginas 19, 24 e 42.

CAMPOS, F. R. Robótica educacional no brasil: questões em aberto, desafios e perspectivas futuras. *Revista ibero-americana de estudos em educação*, Universidade Estadual Paulista Júlio de Mesquita Filho, v. 12, n. 4, p. 2108–2121, 2017. Citado 2 vezes nas páginas 26 e 27.

CAP SISTEMA. *Lista dos 21 melhores módulos para o Arduino*. 29 de junho de 2021. Disponível em: <<https://capsistema.com.br/index.php/2021/06/29/lista-dos-21-melhores-modulos-para-arduino/>>. Acesso em: 13 de junho de 2023. Citado na página 33.

FERREIRA, E. d. P. *Robótica Básica, Modelagem de Robôs*. [S.l.], 1991. Citado na página 26.

FERREIRA, M. Novas tecnologias na sala de aula. *Monografia do Curso de Especialização em Fundamentos da Educação: Práticas Pedagógicas Interdisciplinares. Universidade Estadual da Paraíba, Pró-Reitoria de Ensino Médio, Técnico e Educação à Distância, Departamento da PROEAD, Sousa, PB*, 2014. Citado 2 vezes nas páginas 18 e 19.

LOPES, L. O. et al. O “maker” na escola: uma reflexão sobre tecnologia, criatividade e responsabilidade social. In: SBC. *Anais do IV Congresso sobre Tecnologias na Educação*. [S.l.], 2019. p. 367–376. Citado na página 23.

LORENZIN, M. P.; ASSUMPCÃO, C. M.; RABELLO, M. Metáforas mecânicas: Uma proposta steam para o ensino de ciências. In: *6 Congresso Pesquisa do Ensino educação e tecnologia: revisitando a sala de aula*. [S.l.: s.n.], 2016. p. 1–14. Citado na página 25.

MAIA, D. L.; CARVALHO, R. A. de; APPELT, V. K. Abordagem steam na educação básica brasileira: uma revisão de literatura. *Revista Tecnologia e Sociedade*, v. 17, n. 49, p. 68–88, 2021. Citado na página 25.

- MEDEIROS, L. F. de; WÜNSCH, L. P. Ensino de programação em robótica com arduino para alunos do ensino fundamental: relato de experiência. *Revista Espaço Pedagógico*, v. 26, n. 2, p. 456–480, 2019. Citado na página 38.
- MOR, Y.; NOSS, R. Programming as mathematical narrative. *International Journal of Continuing Engineering Education and Life Long Learning*, Inderscience Publishers, v. 18, n. 2, p. 214–233, 2008. Citado na página 28.
- MORAN, J. M. *Novas tecnologias e mediação pedagógica*, Coleção Papyrus Educação. 16. ed. [S.l.]: Campinas, Papyrus Editora, 2009. Citado na página 18.
- NOEMI, D. *Como a aprendizagem maker pode ser inserida na escola*. [S.l.], 2020. Citado na página 26.
- NUNES, N. B. et al. (des) pluga: O pensamento computacional aplicado em atividades inovadoras:(dis) plug: Computational thinking applied to innovative activities. *Revista Contexto & Educação*, v. 36, n. 114, p. 72–88, 2021. Citado na página 27.
- OLIVEIRA, K. L. R. de et al. Formação online de professores em robótica educacional com práticas no simulador tinkercad. *Revista Brasileira de Ensino de Ciências e Matemática*, v. 5, n. especial, 2022. Citado na página 38.
- OLIVEIRA, R. E.; SANTOS, C. A. M. dos; SOUZA, E. E. de. Aplicação de conceitos e práticas de atividades do movimento maker na educação infantil-um relato de experiência para o ensino fundamental 1. In: SBC. *Anais do XXIV Workshop de Informática Na Escola*. [S.l.], 2018. p. 275–284. Citado na página 24.
- PAPERT, S. *Mindstorms: computers, children and powerful ideas*. [S.l.], 1980. Citado na página 26.
- PAPERT, S. *A máquina das crianças: repensando a escola na era da informática*. [S.l.: s.n.], 1994. Citado na página 18.
- PAULA, B. B. de; MARTINS, C. B.; OLIVEIRA, T. de. Análise da crescente influência da cultura maker na educação: revisão sistemática da literatura no brasil. *Educitec-Revista de Estudos e Pesquisas sobre Ensino Tecnológico*, v. 7, p. e134921–e134921, 2021. Citado na página 24.
- PIAGET, J. *To understand is to invent*. [S.l.], 1974. Citado na página 26.
- RESNICK, M. *Jardim de infância para a vida toda: por uma aprendizagem criativa, mão na massa e relevante para todos*. [S.l.]: Penso Editora, 2020. Citado na página 25.
- RILEY, S. Arts integration and steam: quick resource pack. The Institute for Arts Integration and STEAM: Westminster, MD, 2020. Citado na página 25.
- ROBÓTICA, G. de. Introdução ao arduino. *Notas de aula, Universidade Federal do Mato Grosso do Sul*, p. 10, 2012. Citado 2 vezes nas páginas 30 e 37.
- RODRIGUES, W. C. et al. Metodologia científica. *Faetec/IST. Paracambi*, p. 2–20, 2007. Citado na página 20.
- SANTOS, R. S.; COSTA, D. E.; ALVES, D. B. O papel do professor e do estudante em ambiente construcionista de aprendizagem. *REMATEC*, v. 15, p. 252–265, 2020. Citado na página 18.

SEVERINO, A. J. *Metodologia do trabalho científico*. [S.l.]: Cortez editora, 2017. Citado na página 21.

SILVA, C. S. Gomes da; GIL, M. del C. S. Criando material educacional: invoção, arduino e movimento maker. *Cuadernos de Documentación Multimedia*, v. 30, 2019. Citado na página 20.

SILVEIRA, F. Design & educação: novas abordagens. *A revolução do design: conexões para o século XXI*. São Paulo: Editora Gente, p. 116–131, 2016. Citado na página 23.

SOFFNER, R. *As tecnologias da inteligência e a educação como desenvolvimento humano*. [S.l.], 2005. Citado na página 18.

SOFFNER, R. Tecnologia e educação: um diálogo freire–papert. *Tópicos Educacionais*, v. 19, n. 1, 2013. Citado na página 18.

THE CSTA, S. T. F. *CSTA K-12 Computer Science Standards*. 2011. Citado na página 28.

UNICEF. *Dois milhões de crianças e adolescentes de 11 a 19 anos não estão frequentando a escola no Brasil*. 15 de setembro de 2022.

Disponível em: <<https://www.unicef.org/brazil/comunicados-de-imprensa/dois-milhoes-de-criancas-e-adolescentes-de-11-a-19-anos-nao-estao-frequentando-a-escola-no-brasil>>. Acesso em: 10 de maio de 2023. Citado na página 17.

VAZ, C. L. D.; JÚNIOR, E. d. P. N.; ROCHA, H. d. S. C. D. *Cartas para Marear: Percursos para uma Aprendizagem Criativa em Matemática e Arte*. [S.l.]: Belém, EditAedi/UFPA, 2019. Citado na página 20.

WEBBER, C. G. et al. Experiências do pensamento computacional no ensino de ciências e matemática. *Revista Brasileira de Ensino de Ciências e Matemática*, v. 5, n. especial, p. 120–130, 2022. Citado na página 27.

WING, J. *Computational thinking*. *Communications of the ACM*. v. 49, n. 3, p. 33-35. 2006. Disponível em: <<https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>>. Acesso em: 20 de junho de 2023. Citado na página 27.

YOUNG DIGITAL PLANET S. A. et al. *Educação no século 21: tendências, ferramentas e projetos para inspirar*. [S.l.]: Trad. Danielle Mendes Sales. São Paulo: Fundação Santillana, 2016. Citado na página 24.

ZILLI, S. d. R. et al. *A robótica educacional no ensino fundamental: perspectivas e prática*. Florianópolis, SC, 2004. Citado na página 19.

APÊNDICE A – A Linguagem de Programação do Arduíno

Todos os comandos a seguir e suas sintaxes estão disponíveis no *site* oficial do [ARDUÍNO](#).

Input e Output

Os pinos digitais configurados como *OUTPUT* com a função *pinMode()* são considerados em um estado de baixa impedância. Isso significa que eles podem fornecer uma quantidade substancial de corrente para outros circuitos, podendo assim acender um led, mas se tornando inapto à leitura de sensores.

Os pinos digitais configurados como *INPUT* com a função *pinMode()* são considerados em um estado de alta impedância. Isso os torna úteis para a leitura de um sensor, mas sem capacidade para energizar um led.

PinMode

Configura o pino especificado para se comportar como entrada ou saída, “pin” refere-se ao número do pino que esta sendo utilizado e “mode” ao modo de *INPUT* ou *OUTPUT*. Possui a seguinte sintaxe:

$$\text{pinMode}(\text{pin}, \text{mode})$$

DigitalRead

Lê o valor de um pino digital especificado, como *HIGHT* ou *LOW*. Possui a sintaxe:

$$\text{digitalRead}(\text{pin})$$

Sendo “pin” o número do pino do Arduino que você deseja ler.

DigitalWrite

Se o pino foi configurado como um *OUTPUT* com a função *pinMode()*, sua tensão será ajustada para o valor correspondente: 5V (ou 3,3V em placas de 3,3V) para *HIGH* e 0V (terra) para *LOW*.

Se o pino estiver configurado como *INPUT*, *digitalWrite()* habilitará *HIGH* ou desabilitará *LOW* o *pull-up* interno no pino de entrada.

Se acaso não for definido a função *pinMode()* para *OUTPUT*, e conectar um LED a um pino, ao chamar *digitalWrite(HIGH)*, o LED pode parecer escuro. Se *pinMode()* não for definido de maneira explícita, *digitalWrite()* ativará o resistor *pull-up* interno, que age como um grande resistor limitador de corrente.

O *digitalWrite* possui a seguinte sintaxe:

digitalWrite(pin, value)

Sendo “pin” o número do pino que irá ser lido e “value” refere-se aos valores *HIGH* ou *LOW*.

AnalogRead

Lê o valor do pino analógico especificado. O arduíno possui um conversor analógico para digital multicanal de 10 bits. Com isso ele mapeará as tensões de entrada entre 0 e a tensão de operação (5V ou 3,3V) em valores inteiros entre 0 e 1023. Proporcionando uma resolução entre as leituras de: 5 volts/1024 unidades ou, 0,0049 volts (4,9 mV) por unidade. Possui a sintaxe:

analogRead(pin)

Sendo “pin” o nome do pino de entrada analógica a ser lido, os pinos analógicos correspondem as portas de A0 a A5 na maioria das placas, inclusive no Arduino Uno R3.

AnalogWrite

Grava um valor analógico, uma onda PWM (Pulse Width Modulation) ou Modulação por Largura de Pulso (MLP) em um pino. Podendo ser utilizado para acender um LED em vários brilhos ou acionar um motor em várias velocidades.

Após chamar *analogWrite()*, o pino gerará uma onda retangular constante do ciclo de trabalho especificado até a próxima chamada para *analogWrite()* (ou uma chamada para *digitalRead()* ou *digitalWrite()*) no mesmo pino. Possui a sintaxe:

analogWrite(pin, value)

Sendo “pin” o pino do Arduino para escrever e “value” o ciclo de trabalho entre 0 (sempre desligado) e 255 (sempre ligado).

Setup e Loop

A *setup()* é utilizada para inicializar variáveis e definir utilização de portas, será executada apenas uma vez, após cada energização ou reinicialização da placa Arduino. Temos a seguinte sintaxe:

```
void setup()
{
.
}
```

Depois de criar uma *setup()*, que inicializa e define os valores iniciais, a *loop()* faz exatamente o que seu nome sugere e faz *loops* consecutivamente a não ser que seja programado uma condição de parada. Temos a sintaxe:

```
void loop()
{
.
}
```

High e Low

Quando estamos trabalhando com pinos digitais, eles podem possuir dois valores, são eles: *HIGH* (alto) e *LOW* (baixo).

O significado de *HIGH* (em referência a um pino) é um pouco diferente, dependendo se um pino está definido como um *INPUT* ou *OUTPUT*. Quando um pino é configurado como um *INPUT* com a função *pinMode()*, e lido com *digitalRead()*, o Arduino reportará *HIGH* se:

- uma tensão maior que 3,0V está presente no pino (placas de 5V);
- uma tensão maior que 2,0 V está presente no pino (placas de 3,3 V).

Um pino também pode ser configurado como *INPUT* com a função *pinMode()*, e em seguida lido *HIGHT* com *digitalWrite()*. Isso habilitará os resistores *pull-up* internos de 20K, que puxarão o pino de entrada para uma leitura *HIGH*, a menos que seja puxado *LOW* por um circuito externo.

Quando um pino é configurado para *OUTPUT* com a função *pinMode()*, e definido como *HIGH* com *digitalWrite()*, o pino está em:

- 5 volts (placas de 5V);
- 3,3 volts (placas de 3,3 V)

Podendo então fornecer corrente e acender à um LED conectado através de um resistor em série ao terra.

Para *LOW* o significado também depende se o pino está definido como *INPUT* ou *OUTPUT*. Quando um pino é configurado como um *INPUT* com a função *pinMode()*, e lido com *digitalRead()*, o Arduino reportará *LOW* se:

- uma tensão inferior a 1,5V está presente no pino (placas de 5V);
- uma tensão inferior a 1,0 V (aproximadamente) está presente no pino (placas de 3,3 V).

Quando um pino é configurado como *OUTPUT* com a função *pinMode()*, e definido como *LOW* com *digitalWrite()*, o pino está em 0 volts (placas de 5V e 3,3V). Nesse estado, ele pode drenar corrente, por exemplo, acender um LED conectado por meio de um resistor em série a +5 volts (ou +3,3 volts).

APÊNDICE B – Comandos das Atividades da Oficina

Acender um LED

```
void setup(){  
  
    pinMode(12, OUTPUT);  
  
}  
  
void loop(){  
  
    digitalWrite(12, HIGH);  
  
}
```

Piscar um LED

```
void setup(){  
  
    pinMode(12, OUTPUT);  
  
}  
  
void loop() {  
  
    digitalWrite(12, HIGH);  
    delay(1000); // Wait for 1000 millisecond(s)  
    digitalWrite(12, LOW);  
    delay(1000); // Wait for 1000 millisecond(s)  
  
}
```

Sensor de Distância

```
#define echo 7 // definindo porta do echo6
#define trig 6 // definindo porta do trig

unsigned long tempo; // declarando a variável tempo que irá armazenar a
duração do pulso do som
double distancia; // criando a variável distância

void setup(){

    pinMode(trig, OUTPUT); // configurando pino de saída
    pinMode(echo, INPUT); // configurando pino de entrada

    Serial.begin(9600);

    digitalWrite(trig, LOW); // definindo trig como baixo
}

void loop(){

    digitalWrite(trig, HIGH); // definindo trig como alto
    delayMicroseconds(10); // espera dez milésimos
    digitalWrite(trig, LOW); // definindo trig como baixo

    tempo = pulseIn(echo, HIGH);
    distancia = tempo/58;

    Serial.println(distancia);
    delay(500);
}
```

Sinal via de mão única

```
void setup(){

    pinMode(12, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(2, OUTPUT);

}

void loop(){

    digitalWrite(12, HIGH);
    digitalWrite(10, LOW);
    digitalWrite(8, LOW);
    digitalWrite(4, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(2, HIGH);
    delay(4000); // Wait for 4000 millisecond(s)
    digitalWrite(2, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(4, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(12, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(8, HIGH);
    delay(5000); // Wait for 5000 millisecond(s)
    digitalWrite(8, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(10, HIGH);
    delay(2000); // Wait for 2000 millisecond(s)
    digitalWrite(10, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(12, HIGH);

}
```

Estacionamento Inteligente

```
#define echo 8 // definindo porta do echo6
#define trig 5 // definindo porta do trig

unsigned long tempo; // declarando a variável tempo que irá armazenar
a duração do pulso do som
double distancia; // criando a variável distância

void setup(){

    pinMode(trig, OUTPUT); // configurando pino de saída
    pinMode(echo, INPUT); // configurando pino de entrada
    pinMode(10, OUTPUT); // configurando pino de saída do led vermelho
    pinMode(12, OUTPUT); // configurando pino de saída do led verde
    Serial.begin(9600);
    digitalWrite(trig, LOW); // definindo trig como baixo

}

void loop(){

    digitalWrite(trig, HIGH); // definindo trig como alto
    delayMicroseconds(10); // espera dez milésimos
    digitalWrite(trig, LOW); // definindo trig como baixo

    tempo = pulseIn(echo, HIGH);
    distancia = tempo/58;

    if (distancia <= 50){
        digitalWrite(10, HIGH); // acender led vermelho
        digitalWrite(12, LOW); // apagar led verde
    }else{
        digitalWrite(10, LOW); // caso contrário, apagar led vermelho
        digitalWrite(12, HIGH); // acender led verde
    }
}
```

```
    Serial.println(distancia);
    delay(500);
}
```

Sensor de estacionamento

```
#define echo 8 // definindo porta do echo6
#define trig 5 // definindo porta do trig

unsigned long tempo; // declarando a variável tempo que irá armazenar
a duração do pulso do som
double distancia; // criando a variável distância

void setup(){

    pinMode(trig, OUTPUT); // configurando pino de saída
    pinMode(echo, INPUT); // configurando pino de entrada
    pinMode(13, OUTPUT); // configurando pino de saída do led vermelho
    pinMode(4, OUTPUT); // configurando pino de saída piezzo
    Serial.begin(9600);
    digitalWrite(trig, LOW); // definindo trig como baixo

}

void loop(){

    digitalWrite(trig, HIGH); // definindo trig como alto
    delayMicroseconds(10); // espera dez milésimos
    digitalWrite(trig, LOW); // definindo trig como baixo

    tempo = pulseIn(echo, HIGH);
    distancia = tempo/58;

    if (distancia <=$ 20){
        digitalWrite(13, HIGH);
        digitalWrite(4, HIGH);
    }
}
```

```
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(4, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
  } else{
    digitalWrite(13, LOW);
    digitalWrite(4, LOW);
  }

  Serial.println(distancia);
  delay(500);

}
```