



UNIVERSIDADE ESTADUAL DO MARANHÃO – UEMA
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO E
SISTEMAS - PECS

MESTRADO PROFISSIONAL EM ENGENHARIA DE COMPUTAÇÃO E SISTEMAS

**ROBÓTICA EDUCACIONAL LIVRE NO PROCESSO DE ENSINO E
APRENDIZAGEM DE PROGRAMAÇÃO NO ENSINO BÁSICO**

NELIO ALVES GUILHON

São Luís
2023

UNIVERSIDADE ESTADUAL DO MARANHÃO – UEMA
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO E
SISTEMAS - PECS

MESTRADO PROFISSIONAL EM ENGENHARIA DE COMPUTAÇÃO E SISTEMAS

NELIO ALVES GUILHON

**ROBÓTICA EDUCACIONAL LIVRE NO PROCESSO DE ENSINO E
APRENDIZAGEM DE PROGRAMAÇÃO NO ENSINO BÁSICO**

Dissertação apresentada ao programa do Mestrado Profissional em Engenharia de Computação e Sistemas da Universidade Estadual do Maranhão como parte dos requisitos para obtenção do título de Mestre em Engenharia da Computação.

Orientador: Prof. Dr. Luís Carlos Costa Fonseca.

São Luís
2023

Guilhon, Nélio Alves.

A Robótica educacional livre no processo de ensino e aprendizagem de programação no ensino básico./ Nélio Alves Guilhon. – São Luís, 2023.
155f

Dissertação (Mestrado em Engenharia de Computação e Sistemas) -
Universidade Estadual do Maranhão, 2023.

Orientador: Prof. Dr. Luís Carlos Costa Fonseca.

1.Aprendizagem .2.Arduino. 3.Robótica Educacional. 4.Ensino de Computação.
5. Ambiente de Programação I. Título.

CDU:681.5:373

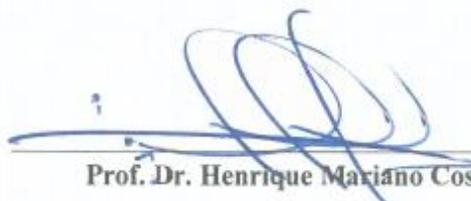
NELIO ALVES GUILHON

**ROBÓTICA EDUCACIONAL LIVRE NO PROCESSO DE ENSINO E
APRENDIZAGEM DE PROGRAMAÇÃO NO ENSINO BÁSICO**

Dissertação apresentada ao programa do Mestrado Profissional em Engenharia de Computação e Sistemas da Universidade Estadual do Maranhão como parte dos requisitos para obtenção do título de Mestre em Engenharia da Computação, sob a orientação do Prof. Dr. Luís Carlos Costa Fonseca.

Aprovado em 25/8/23


Prof. Dr. Luís Carlos Costa Fonseca


Prof. Dr. Henrique Mariano Costa do Amaral


Prof. Dr. Sofiane Labidi

*A minha família pelo incentivo e apoio
incondicional em toda minha trajetória,*

Agradecimentos

Em primeiro lugar, agradeço a Deus.

Agradeço a minha esposa Rubenice, por sempre está ao meu lado.

A minha querida filha Nayanne, pelo amor e companhia.

Aos meus colegas de mestrado que caminharam comigo durante o desafio da pandemia.

Agradeço a todos os professores que me orientaram ao longo da minha trajetória no mestrado, pelo saber e pelas vivências compartilhadas.

Ao professor e orientador Luiz Carlos, expresso minha gratidão pela direção, empatia nos momentos mais desafiadores e pelo constante apoio ao longo da realização deste trabalho.

“A vida é combate, que os fracos abate,
que os fortes, os bravos só pode exaltar!”

(Gonçalves Dias)

RESUMO

A presente pesquisa aborda a utilização da Robótica Educacional (RE) como estratégia didática para o ensino de computação no ensino básico. O problema investigado foi o de compreender como a Robótica pode potencializar o processo de ensino dos fundamentos da programação para os alunos do Ensino Médio, sendo utilizada como ferramenta didática ao docente. A robótica promove no aluno, além da motivação, outros aspectos positivos para a sua aprendizagem como, raciocínio lógico, trabalho em equipa, interdisciplinaridade, autonomia. O campo de pesquisa deste trabalho foi o Colégio de aplicação da Universidade Federal do Maranhão, tendo como participantes alunos do Ensino Médio. Os objetivos específicos elencados nesta pesquisa foram: elaborar uma oficina de programação e robótica, desenvolver atividades que utilizando RE para o ensino aprendizagem da programação, integração da oficina a um Ambiente de Programação Robótica educacional, descrever o ponto de vista dos estudantes sobre a experiência vivenciada e avaliar os resultados obtidos na oficina. Para que estes objetivos fossem alcançados, utilizou-se a Plataforma Arduino, um kit licenciado no conceito de hardware livre. As estratégias da sua implementação assentaram na utilização da aprendizagem baseada em problemas e do trabalho colaborativo. Os resultados demonstram que a utilização da RE contribuiu para o cumprimento do objetivo proposto.

Palavras-chave: aprendizagem; Arduino; Robótica Educacional; Ensino de Computação; Ambiente de Programação.

ABSTRACT

This research addresses the use of Educational Robotics (ER) as a didactic strategy for teaching computer science in basic education. The problem explored was understanding how Robotics can enhance the teaching process of programming fundamentals for high school students, serving as a didactic tool for teachers. Robotics fosters in students not only motivation but also other positive aspects for their learning such as logical reasoning, teamwork, interdisciplinarity, and autonomy. The research setting was the Application School of the Federal University of Maranhão, with participating students from high school. The specific objectives outlined in this study were: to create a programming and robotics workshop, develop activities using ER for teaching and learning programming, integrate the workshop into a Robótica educacional Programming Environment, describe the students' perspectives on the lived experience, and evaluate the results obtained in the workshop. To achieve these objectives, the Arduino Platform was used, a kit licensed under the open hardware concept. Its implementation strategies were based on problem-based learning and collaborative work. The results show that the use of ER contributed to achieving the proposed goal.

Keywords: *learning; Arduino; Educational Robotics; Computer Science Education; Programming Environment.*

Lista de Figuras

Figura 1:	Fundamentos da ABP.	33
Figura 2:	Habilidades computacionais para a Educação Básica.	36
Figura 3:	Arduino Uno.	46
Figura 4:	Tela inicial do Tinkercad.	49
Figura 5:	Projeto de um semáforo.	50
Figura 6:	Representação robótica educacional e textual de um código.	51
Figura 7:	Modelo de Aceitação de Tecnologia	53
Figura 8:	Sensores e atuadores.	57
Figura 9:	Questionário 1 – Capacitação e Programação.	64
Figura 10:	Questionário – Gráfico 1.	65
Figura 11:	Questionário – Gráfico 2.	66
Figura 12:	Questionário – Gráfico 3.	67
Figura 13:	Questionário – Gráfico 4.	68
Figura 15:	Questionário – Gráfico 6.	70
Figura 16:	Questionário – Gráfico 7.	71
Figura 17:	Gráfico descritivo.	73

Lista de Tabelas

Tabela 1: Estatística descritiva dos constructos (N = 20).

72

Lista de Siglas

ABP	Aprendizagem Baseada em Projetos
BBPEs	Block-Based Programming Environments
BDTD	Biblioteca Digital Brasileira de Teses e Dissertações
BNCC	Base Nacional Comum Curricular
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNE	Conselho Nacional de Educação
COLUN	Colégio Universitário
MEC	Ministério da Educação
NTIC	Novas Tecnologias da Informação e Comunicação
PBL	Aprendizagem Baseada em Projetos (do inglês "Project-Based Learning")
RE	Robótica Educacional
REL	Robótica Educacional Livre
SBC	Sociedade Brasileira de Computação
TAM	Modelo de Aceitação de Tecnologia
TIC	Tecnologias da informação e comunicação
TRA	Teoria da Ação Racional
UFMA	Universidade Federal do Maranhão
UTAUT	Teoria Unificada da Aceitação e Uso de Tecnologia
VPL	Linguagens de Programação Visual (do inglês "Visual Programming Languages")

Lista de Quadros

Quadro 1:	Conexões entre o construtivismo e o ensino de programação.	26
Quadro 2:	Benefícios decorrente do uso do construcionismo.	31
Quadro 3:	Competências da computação no ensino fundamental.	36
Quadro 4:	Competências da computação no ensino médio.	38
Quadro 5:	Definições de Robótica Educacional.	39
Quadro 6:	Modulo da Oficina.	56
Quadro 7:	Instrumento de Pesquisa.	59

Sumário

1. INTRODUÇÃO	16
1.1. Motivação	18
1.2. Questão de investigação e Objetivos do trabalho	20
1.3. Objetivos	20
1.3.1. Objetivos gerais.....	20
1.3.2. Objetivos específicos.....	21
1.4. Abordagem metodológica	21
1.4.1. Organização da pesquisa	22
1.4.2. Organização do Trabalho	23
2. FUNDAMENTAÇÃO TEÓRICA	24
2.1. Teorias de Aprendizagem	25
2.1.1. Abordagem Construtivismo	25
2.1.2. Abordagem Construcionista	27
2.2. Metodologias Ativas	32
2.2.2. Aprendizagem baseada em projetos	33
2.3. A BNCC e a Programação	34
2.4. Robótica Educacional no Ensino de Programação	38
2.5. Software e Hardware Livres	44
2.6. Plataforma Arduino	45
2.7. Ambientes Visuais de Programação	47
2.8. Plataforma Tinkercad	48
2.9. Modelo de Aceitação da Tecnologia	52
3. ESTUDO EXPERIMENTAL	55
3.1. Oficina em Programação e Robótica	56
3.2. Descrição da pesquisa	59
4. ESTUDO DE CASO E DISCUSSÕES.....	61
4.1. Estudo de caso	62
4.2. Discursão	72

5. CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS	75
5.1. Conclusões	76
5.2. Contribuições da Pesquisa	77
5.3. Trabalhos Futuros e Limitações	78
6. BIBLIOGRAFIA.....	81
APENDICE A	91
APENDICE B.....	108
APENDICE C.....	122
APENDICE D	130
APENDICE E.....	137
APENDICE F	144
APENDICE G	151
APENDICE H	153

1. INTRODUÇÃO

O uso das tecnologias da informação e comunicação (TIC) tem impactado significativamente a sociedade nas últimas décadas. A tecnologia se tornou uma parte importante da vida cotidiana, afetando a forma como as pessoas se comunicam, consomem informações, realizam transações financeiras, trabalham, aprendem e se divertem. A evolução tecnológica tem se tornado cada vez mais presente em nossa vida diária, desde o uso de smartphones, tablets e computadores até a automação de processos em indústrias e serviços. Além disso, a informática tem contribuído para a melhoria da qualidade de vida das pessoas, com a criação de aplicativos e dispositivos eletrônicos que monitoram a saúde, a prática de atividades físicas e o bem-estar.

Hoje em dia, é cada vez mais comum ver crianças já nascendo em um mundo totalmente imerso na cultura digital, desde muito cedo, elas são expostas a dispositivos eletrônicos, essa imersão precoce no mundo digital tem gerado discussões sobre os impactos na saúde e no desenvolvimento das crianças. É preciso visualizar essa situação social que estamos vivendo.

Porém, o fato das TIC's já fazerem parte do nosso cotidiano, não significa que estamos fluentes no seu uso. Para o pesquisador Mitchel Resnick (2013), a verdadeira fluência digital vai além do uso natural das TIC, ela se baseia na habilidade de projetar, criar e se expressar através dessas tecnologias, sendo a programação uma competência essencial para alcançar esse objetivo, é imperativo que se saiba “projetar, criar e se expressar através dessas tecnologias” e, para tanto, é imprescindível saber programar. A aprendizagem de programação não se limita apenas a desenvolver habilidades técnicas. Ela é também uma importante ferramenta para o desenvolvimento do Pensamento Computacional, como afirmam os pesquisadores Grover e Pea (2013). O Pensamento Computacional é uma forma de raciocínio lógico que utiliza conceitos de programação para solucionar problemas complexos. Ele envolve habilidades como decomposição de problemas, reconhecimento de padrões, abstração e algoritmos, e é cada vez mais valorizado em diversos campos profissionais.

Resnick (2014) destaca que a fluência digital vai além da simples habilidade de navegar e interagir em ambientes digitais. Ela requer também a capacidade de projetar, criar e inventar novas mídias. Para alcançar esse nível de fluência, é essencial aprender alguma forma de linguagem de programação. Nas palavras de Flannery:

Vejo a programação de computadores como uma extensão da escrita. A capacidade de programar permite "escrever" novos tipos de coisas - histórias interativas, jogos, animações e simulações. E, como acontece com a escrita tradicional, há fortes razões para que todos possam aprender a escrever em código, mas eu vejo razões muito mais profundas e mais amplas para aprender. No processo de aprender a programar, as pessoas aprendem muitas outras coisas. Eles não estão apenas aprendendo a programar, elas estão programando para aprender. Além de aprender ideias matemáticas e computacionais (como variáveis e condicionais), eles também estão aprendendo estratégias para a resolução de problemas, elaboração de projetos, e comunicação de ideias. Estas habilidades são úteis não apenas para os profissionais de TI, mas para todos, independentemente da idade, origem, interesses ou ocupação. (Flannery *et al.*, 2013)

No Brasil, o Ministério da Educação (MEC) reconhece a importância do letramento digital como um direito a ser assegurado e um instrumento fundamental para a formação de cidadãos críticos e atuantes na vida social. Por essa razão, o MEC definiu, na Base Nacional Comum Curricular (BNCC), um tema integrador denominado “Culturas Digitais e Computação”. Esse tema está relacionado ao uso pedagógico das Novas Tecnologias da Informação e Comunicação (NTIC) e à exploração dessas tecnologias para a compreensão do mundo e para a atuação nele. Segundo o MEC (2016), em uma sociedade cada vez mais tecnologicamente organizada, torna-se imprescindível que as escolas passem a considerar as potencialidades do uso dos recursos tecnológicos para o alcance de suas metas educacionais.

A Sociedade Brasileira de Computação (SBC) também tem se mostrado preocupada com a temática da introdução da tecnologia na educação, em especial no ensino de computação, e tem desenvolvido ações para incentivar e fomentar o debate sobre o assunto. No ano de 2016, a SBC realizou reuniões com o (MEC), com o objetivo de discutir a inclusão da disciplina de Ciência da Computação na grade curricular do ensino fundamental e médio no país (SBC, 2016a; SBC, 2016b). E em 2022 A SBC defende que o ensino de Ciência da Computação é fundamental para a formação de cidadãos críticos e preparados para lidar com o mundo cada vez mais tecnológico. Em fevereiro de 2022 a Câmara de Educação Básica do Conselho Nacional de Educação (CNE), homologou a resolução que define normas sobre a Computação na Educação Básica, em complemento à Base Nacional Comum Curricular (BNCC).

No entanto, o ensino e a aprendizagem da programação representam um desafio significativo tanto para os professores quanto para os alunos (KELLEHER e PAUSCH, 2005). A ocorrência frequente de altas taxas de insucesso em disciplinas introdutórias de programação, independentemente do nível de ensino, é uma preocupação constante, motivando a busca por propostas que possam aprimorar os resultados nessa área. Uma das

hipóteses levantadas para abordar essa questão foi a introdução da Robótica Educacional (RE) como instrumento metodológico. Essa abordagem permite que os alunos observem o comportamento dos experimentos robóticos que eles próprios desenvolvem. Benitti (2012), reforça que a presença da RE durante as aulas pode enriquecer as experiências educacionais e proporcionar uma assimilação mais rápida dos conceitos apresentados. Além disso, o interesse pela robótica estimula uma aprendizagem lúdica e divertida de programação, contribuindo para um processo educativo mais envolvente e dinâmico.

Neste contexto, desde 2009, o MEC vem divulgando os benefícios da Robótica Educacional (RE) por meio do Guia das Tecnologias Educacionais (GTE)¹. O propósito desse documento é selecionar e pré-qualificar produtos e propostas que visam promover o aprimoramento da qualidade da educação básica (GTE, 2013). A RE se configura como ambientes de aprendizagem que proporcionam ao aluno produzir e programar um robô ou sistema robotizado.

1.1. Motivação

Aprender programação de computadores pode ser um processo desafiador, exigindo um alto nível de abstração e compreensão de linguagens com sintaxe complexa. De acordo com Gomes et al. (2008), muitos alunos iniciantes enfrentam dificuldades na resolução de problemas, pois lhes faltam competências necessárias para essa tarefa. Os métodos tradicionais de ensino de programação, que utilizam fluxogramas e pseudocódigos para ilustrar conceitos abstratos, nem sempre levam em consideração o processo de construção da solução.

Além disso, há outro desafio no ensino de programação, conforme apontado por Gomes et al. (2008): a motivação dos alunos em um ambiente de aprendizagem. Muitas vezes, o que é ensinado segue abordagens baseadas em leituras sucessivas, memorização e mecanização de procedimentos. No entanto, essa abordagem mecânica da aprendizagem não fornece as razões e fundamentos por trás do uso de determinados conceitos de programação. Aprender programação requer abordagens que facilitem a compreensão dos alunos, incentivem a resolução de problemas de forma mais prática e motivadora, e que demonstrem a relevância e os propósitos por trás dos conceitos ensinados. Superar esses desafios contribuirá para uma aprendizagem mais significativa e eficaz na área de programação de computadores.

¹ <http://portal.mec.gov.br/guia-de-tecnologias>

No contexto teórico da Educação, Papert (1993) ressalta que o uso de robôs como ferramentas educacionais tem um grande potencial para criar um ambiente propício ao aprendizado dentro da sala de aula. Por sua vez, Rusk et al. (2008) destacam que a RE pode promover uma interdisciplinaridade interessante, explorando uma ampla variedade de aplicações possíveis para envolver os jovens com diferentes interesses. Essas perspectivas teóricas destacam a relevância e o potencial da utilização de robôs como artefatos educacionais, enriquecendo significativamente o processo de aprendizagem em sala de aula.

Com a incorporação da RE, além das questões de motivação e aquisição de conhecimento, torna-se possível criar um ambiente de trabalho colaborativo e promover atividades interpessoais, algo que muitas vezes é difícil de alcançar com instrumentos tradicionais. A RE abrange competências para além daquelas propostas pelos currículos escolares, proporcionando aos alunos experiências reais nas áreas da informática, eletrônica, mecânica e design, por meio da resolução de problemas ou desafios específicos (PAPERT, 1993). Dessa forma, a utilização da RE no ensino de programação permite uma abordagem mais prática e interdisciplinar, ampliando as oportunidades de aprendizado e preparando os alunos para enfrentar os desafios do mundo contemporâneo.

A Aplicação da RE contribui para o desenvolvimento de competências fundamentais para alunos iniciantes em programação, tais como raciocínio lógico, habilidades de representação e comunicação, resolução de problemas por meio de tentativa e erro, aplicação prática das teorias formuladas e capacidade crítica (Zilli 2004). Através de desafios e atividades com robôs, os estudantes têm a oportunidade de desenvolver uma compreensão mais profunda dos conceitos de programação, adquirindo competências essenciais para o domínio dessa área e para o desenvolvimento de uma abordagem crítica e analítica diante de questões complexas.

Quando os robôs estão montados, é necessário fornecer-lhes inteligência por meio da programação. Existem várias formas de programação para esse fim, uma delas são as plataformas de programação em blocos, que são chamadas de “block-based programming environments” (BBPEs) (MILNE e LADNER, 2019). Essas plataformas permitem que os alunos montem sequências de blocos para instruir o robô sobre o que fazer. Cada bloco tem uma função determinada, como mover-se para a frente, girar, aguardar ou detectar obstáculos. Ao organizar esses blocos em uma sequência lógica, os alunos constroem programas visuais que controlam o comportamento do robô.

Neste contexto, a RE tem se destacado como uma ferramenta extremamente vantajosa, especialmente no contexto da educação científica e tecnológica (RIBEIRO, COUTINHO & COSTA, 2009). Embora ainda seja uma abordagem emergente, ela tem mostrado um grande potencial para enriquecer o processo de aprendizagem, proporcionando uma experiência prática e interativa no campo da ciência e tecnologia.

Assim, a RE associada à estratégia de aprendizagem baseada em problemas, à aprendizagem colaborativa e a abordagem construcionista são usados como ferramentas para a aprendizagem dos fundamentos de programação no ensino básico, associada a programação visual como plataforma de programação. Segundo vários pesquisadores e Kumar (2001) os alunos participam do processo de ensino e aprendizagem de forma lúdica e divertida.

1.2. Questão de investigação e Objetivos do trabalho

Ao longo deste desafio, serão apresentadas diversas atividades contextualizadas, com foco na programação, e sempre que possível, apoiadas nos módulos de programação. O objetivo é verificar a adequação dessas atividades às orientações e recomendações curriculares, coletar dados, analisar e avaliar o desempenho e as aprendizagens dos estudantes. Através desse processo, busca-se proporcionar uma experiência de aprendizagem mais abrangente e eficaz, onde os alunos podem aplicar seus conhecimentos de programação em situações práticas e relevantes, e os resultados obtidos são utilizados para aprimorar e adaptar as estratégias educacionais, visando sempre melhorar o processo de ensino-aprendizagem.

Definiu-se a seguinte questão de investigação:

- **A Robótica Educacional auxilia no processo de aprendizagem da programação?**

1.3. Objetivos

Para responder à questão de investigação foram definidos os seguintes objetivos:

1.3.1. Objetivos gerais

Explorar, por meio da percepção dos estudantes, o desenvolvimento de atividades de ensino-aprendizagem da programação, utilizando-se como ferramenta a Robótica Educacional.

1.3.2. Objetivos específicos

1. Elaborar uma oficina de programação e robótica;
2. Desenvolver atividades que utilizem RE para o ensino aprendizagem da programação, com base na literatura disponível atinente à aplicação da RE para o ensino da programação e tendo como referencial a pedagogia de projetos, visando a escolha das metodologias ativas para aplicação mais adequada aos participantes;
3. Integração da oficina a um Ambiente de Programação Visual;
4. Descrever a participação e o ponto de vista dos estudantes sobre a experiência vivenciada.
5. Avaliação dos resultados obtidos na oficina.

1.4. Abordagem metodológica

Quanto à metodologia de pesquisa utilizada, este trabalho é classificado como um estudo de caso, exploratório e de abordagem qualitativa. A escolha por conduzir uma pesquisa por meio de um estudo de caso torna-se relevante quando o objetivo é investigar fenômenos contemporâneos inseridos em um contexto real, o que se aplica diretamente ao escopo deste estudo (YIN, 2001, p.19).

Os estudos de caso apresentam características distintas e relevantes que os tornam uma excelente ferramenta para a observação e interpretação de contextos educacionais (MERRIAM, 1988 apud SOUZA, 2007). Eles se preocupam em oferecer uma descrição rica e vívida de eventos relevantes para o caso e, o pesquisador se envolve integralmente retratando com mais riqueza, quando da escrita do relato.

Neste tipo de pesquisa, é comum trabalhar com amostras reduzidas, e a análise dos dados é conduzida de forma qualitativa (MALHOTRA, 2001 apud OLIVEIRA, 2011). Segundo Creswell (2007), na abordagem qualitativa, o ambiente natural é a principal fonte de dados, e o pesquisador desempenha um papel fundamental como instrumento de coleta e análise. Os dados coletados são predominantemente descritivos, visando compreender os aspectos do fenômeno estudado em sua totalidade. Nesse contexto, a ênfase recai no processo mais do que no produto, ou seja, o interesse do pesquisador está em compreender "como" o problema se manifesta nas atividades, procedimentos e interações cotidianas.

Uma característica relevante é que a análise dos dados tende a seguir um processo indutivo, no qual os padrões e temas emergem dos dados coletados, em vez de serem pré-definidos ou estritamente pré-configurados. A pesquisa qualitativa é flexível e adaptativa,

permitindo que o pesquisador explore novos insights e direções à medida que o estudo progride, possibilitando uma compreensão mais profunda e holística do fenômeno em análise.

Na observação participante, o pesquisador tem a possibilidade de assumir diversas atribuições e participar ativamente dos eventos sob estudo (YIN, 2001), como no caso desta pesquisa em que o investigador desempenhou o papel de professor durante a oficina realizada para a coleta de dados. Dessa forma, ele participou diretamente do fenômeno investigado, contribuindo de maneira ativa para o contexto em análise.

1.4.1. Organização da pesquisa

Para criar o cenário adequado para a realização do estudo de caso apresentado e disponibilizar as ferramentas necessárias para as investigações propostas, este trabalho foi estruturado em etapas. Cada etapa teve como objetivo fornecer o suporte necessário para a execução adequada da próxima fase, culminando, por fim, na realização do próprio estudo de caso.

1. Revisão Bibliográfica,
 - a. Plataformas de Programação,
 - b. Plataformas de Programação Visual,
 - c. Estudo Robótica Educacional,
2. Seleção dos materiais de robótica, tendo como orientação a Robótica Educacional Livre (REL),
3. Escolha das plataformas de desenvolvimento,
4. Definição das estratégias pedagógicas a serem adotadas,
5. Desenho de um conjunto de atividades didáticas.
6. Desenvolvimento de materiais auxiliares de robótica,
7. Desenvolvimento de avaliação.
8. Realização de uma oficina com alunos do ensino médio, do Colégio Universitário da UFMA.
9. Análise dos dados

O estudo bibliográfico foi realizado com o intuito de encontrar trabalhos relacionados com RE e o estudo de programação no ensino básico e outros temas relacionados com o tema principal da pesquisa. Com base nas informações apresentadas por Prodanov e Freitas (2013), foram estabelecidas as fontes teóricas para embasar o estudo. A partir dessas referências, foram selecionadas as pesquisas mais relevantes relacionadas ao tema em questão. Em seguida, os resultados dessas pesquisas foram analisados e as informações

encontradas foram discutidas detalhadamente. Esse método permite aprofundar a compreensão do tema em análise, proporcionando uma base consistente para o desenvolvimento da pesquisa e contribuindo para a produção de conhecimento mais embasado e confiável. As bases de dados utilizadas foram: IEEE Xplore¹, Google Acadêmico², Portal de Periódicos da CAPES³, Scielo Brasil⁴ e Biblioteca Digital Brasileira de Teses e Dissertações (BDTD)⁵. Finalmente, foram mencionados os estudos identificados que apresentavam maior afinidade com o tema da pesquisa, bem como a exposição de suas principais ideias e conceitos.

A escolha da plataforma de desenvolvimento de programação é uma decisão crítica para bom andamento do estudo. As plataformas de desenvolvimento podem incluir ambientes de programação, frameworks, bibliotecas, linguagens de programação, sistemas operacionais, entre outros recursos que facilitam o processo de criação de algoritmos. Foi escolhida a programação Visual, principalmente por questões intuitiva e por apresentarem menor ênfase em codificação manual. As plataformas de desenvolvimento de programação visual são projetadas para permitir que os usuários construam aplicativos, sites, jogos e outras aplicações por meio de interfaces gráficas e arrastar e soltar elementos, reduzindo a necessidade de escrever código tradicional.

1.4.2. Organização do Trabalho

O trabalho está organizado em seis capítulos, cada um abordando aspectos específicos da pesquisa realizada sobre robótica educacional livre no processo de ensino e aprendizagem de programação especialmente no ensino básico. O Capítulo 1, são apresentadas informações essenciais sobre o contexto do estudo, incluindo o tema em questão e o cenário em que a pesquisa foi conduzida. A delimitação do escopo do estudo é definida, bem como a justificativa, os objetivos delineados no que se pretende alcançar, a metodologia é descrita, destacando como os dados serão coletados e analisados. O Capítulo 2 apresenta a fundamentação teórica, abordando as perspectivas de outros autores e pesquisadores por meio do referencial teórico. No capítulo 3 é discutido o detalhado dos materiais e métodos utilizados. Isso inclui a descrição dos recursos e ferramentas empregados nas aulas. Apresentamos a análise e discussão dos dados no capítulo 4. Por fim, o Capítulo 5 apresenta os resultados da pesquisa, contribuições, limitações e sugestões para trabalhos futuros.

¹ <https://ieeexplore.ieee.org/>

² <https://scholar.google.com.br/>

³ <https://www-periodicos-capes-gov-br.ez1.periodicos.capes.gov.br/>

⁴ <https://www.scielo.br/>

⁵ <http://bdtb.ibict.br/vufind/>

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo teve o intuito de organizar o embasamento teórico fundamental para a compreensão e desenvolvimento do presente estudo. Através da revisão da literatura e análise crítica das principais teorias e conceitos relacionados ao nosso tema de pesquisa, buscamos estabelecer um sólido alicerce para as etapas subsequentes deste trabalho.

Foram explorados os principais tópicos que abrangem o escopo deste estudo, incluindo:

- 2.1 Teorias de aprendizagem
- 2.2 Metodologias ativas
- 2.3 A BNCC e a programação
- 2.4 Robótica Educacional no ensino de programação
- 2.5 Software e Hardware Livres
- 2.6 Plataforma Arduino
- 2.7 Ambientes visuais de programação
- 2.8 Plataforma Tinkercad
- 2.9 Modelo de aceitação da tecnologia

2.1. Teorias de Aprendizagem

2.1.1. Abordagem Construtivismo

Surgindo nos anos 80, a abordagem construtivista apresentou-se como uma alternativa ao ensino tradicional. Neste modelo antigo, o aluno era relegado a um papel de mero observador, enquanto o professor dominava ativamente o processo de ensino, focando na memorização. Esse enfoque na memorização de conteúdos impedia o aluno de desenvolver a adaptabilidade cognitiva necessária para aplicar o conhecimento em diferentes contextos.

Piaget (1979) foi uma figura central para o construtivismo. Ele observou que, embora uma criança nasça com uma predisposição genética, ela não é inicialmente capaz de executar simples funções de pensamento. Isso exige que a criança passe por um contínuo desenvolvimento e reorganização mental. Esse processo é particularmente evidente quando a criança se confronta com situações desconhecidas, que geram desequilíbrio cognitivo. Para resolver isso, a criança precisa adaptar ou refinar seu modelo mental para entender a situação.

Campos (2019) relata que Piaget identificou dois mecanismos mentais, assimilação e acomodação, como fundamentais para esse ajuste cognitivo. Ele descreveu a assimilação como a maneira pela qual novas informações são integradas com base em conhecimentos anteriores para tornar novas experiências compreensíveis. Nesse processo, a estrutura mental da criança permanece inalterada; apenas a nova informação é incorporada. Ele começou seus estudos observando bebês em seu processo de desenvolvimento e emancipação, desde momentos iniciais em que não reconhecem sua própria individualidade em relação ao mundo exterior até a adolescência, período no qual começam a desenvolver raciocínios mais complexos.

Silveira Junior e seus colaboradores (2015), expressaram que a partir dessas observações, Piaget formulou uma metodologia de análise conhecida como método clínico, estabelecendo assim os fundamentos de sua teoria, a epistemologia genética. Essa base teórica está articulada na seguinte fala: “as relações entre o sujeito e o seu meio consistem numa interação radical, de modo tal que a consciência não começa pelo conhecimento dos objetos nem pela atividade do sujeito, mas por um estado diferenciado” (PIAGET, 1987, p. 78).

Na perspectiva construtivista, o aluno é percebido como o arquiteto de seu próprio aprendizado, porém, está imerso em um contexto social e cultural que influencia e molda seu

conhecimento (VYGOTSKY, 1989). O processo de construção do conhecimento, fundamental para a aprendizagem, capacita os estudantes a integrarem novas informações com base em conceitos previamente adquiridos. Essa construção é um amálgama de interação, análise, vivência e falhas. Dessa forma, o ensino e a aprendizagem não devem ser reduzidos a simples atividades repetitivas (SILVEIRA *et al.*, 2019).

O construtivismo, originado nas teorias de Jean Piaget, propõe que os aprendizes construam ativamente seu conhecimento através de interações com o ambiente, ao invés de apenas receber informações passivamente. O ensino de programação, com sua abordagem prática e orientada a problemas, encaixa-se bem dentro deste paradigma construtivista.

Segundo vários autores, existe conexões entre o construtivismo e o ensino de programação, segundo informações contidas no (Quadro 1)

Quadro 1: Conexões entre o construtivismo e o ensino de programação

Aprendizado Ativo
<p><u>Programação</u>: O processo de escrever código e depurá-lo é inerentemente ativo e participativo (PAPERT, 1993).</p> <p><u>Construtivismo</u>: Aprendizes são vistos como ativos em seu processo de aprendizado (SILVEIRA <i>et al.</i>, 2018).</p>
Construção do Conhecimento
<p><u>Programação</u>: Os aprendizes constroem programas peça por peça, conceito por conceito, similar a construir o conhecimento (KELLEHER e PAUSCH, 2005).</p> <p><u>Construtivismo</u>: O aluno é visto como construtor do seu conhecimento (SILVEIRA <i>et al.</i>, 2018).</p>
Resolução de Problemas
<p><u>Programação</u>: A natureza da programação envolve resolver problemas de maneira estruturada (PAPERT, 1993).</p> <p><u>Construtivismo</u>: A resolução de problemas permite aos alunos transportarem as competências adquiridas num contexto para outro</p>

(AFONSO, 2013).
Colaboração
<u>Programação</u> : A programação promove a colaboração e aprendizado em grupo (WERNER <i>et al.</i> , 2012). <u>Construtivismo</u> : A aprendizagem é social e colaborativa (VYGOTSKY, 1980).
Autonomia do Aluno
<u>Programação</u> : Ambientes de programação modernos oferecem aos aprendizes uma grande autonomia em como abordar problemas (RESNICK, 2013). <u>Construtivismo</u> : Encoraja a independência e a exploração pelo aluno (CAMPOS, 2019).

Fonte: Elaborado pelo autor

Implementar uma abordagem construtivista no ensino de programação requer que educadores projetem ambientes de aprendizado que permitam a exploração, experimentação e reflexão. Ferramentas e plataformas de programação, principalmente as com apelo visual, já adotam essa filosofia, oferecendo aos alunos um ambiente rico e interativo onde eles podem "brincar" com conceitos de programação e aprender através da descoberta. Além disso, o construtivismo oferece uma lente útil através da qual o ensino de programação pode ser visto, orientando a prática pedagógica para uma abordagem mais centrada no aluno e baseada em projetos. Referenciar e alinhar o ensino de programação com teorias construtivistas pode enriquecer e melhorar a experiência de aprendizado.

O trabalho apresentado nesta pesquisa propõe uma abordagem construtivista no ensino de programação através do uso da robótica educacional, pois trata o aluno como sujeito ativo, criador da sua própria lógica, incluindo o uso de ferramentas de ensino de programação visual, deixando de utilizar apenas os modelos repetitivos explorados nas metodologias tradicionais de ensino.

2.1.2. Abordagem Construcionista

Segundo Arendt (2003), o construcionismo é baseado na hipótese mestra piagetiana de que não existem estruturas cognitivas inatas, sendo estas construídas pelo sujeito, no decorrer de suas interações com o meio. Nessa perspectiva, a ênfase está na priorização dos objetivos do aluno, suas experiências pessoais e o desenvolvimento de estratégias metacognitivas. A teoria construcionista enfatiza a construção ativa do conhecimento pelo próprio aprendiz, ao criar coisas tangíveis ou digitais. Através dessa criação, os alunos interagem com conceitos e ideias de forma concreta, o que facilita a compreensão e retenção do conhecimento. O processo de construção de artefatos também encoraja os alunos a refletirem sobre suas ações e estratégias, desenvolvendo habilidades metacognitivas, como a autorregulação e o monitoramento do próprio aprendizado.

A teoria Construcionista de Papert foi influenciada por, além de Piaget, Lev Vigotsky, Montessori, John Dewey e Paulo Freire (SILVEIRA, 2012). Papert tinha como alvo reduzir a dependência dos aprendizes em relação aos adultos que tradicionalmente eram considerados os provedores de informações na educação. Para isto, ele propôs a incorporação de computadores no processo de ensino-aprendizagem, com a intenção de substituir os livros e cadernos por essa tecnologia de forma igualmente eficaz. Dessa maneira, ele buscava empoderar os estudantes, permitindo que eles explorassem o conhecimento de forma mais autônoma e interativa, com o auxílio das ferramentas computacionais.

Enquanto o construtivismo delimita a construção de estruturas de conhecimento por intermédio da internalização progressiva de ações, o construcionismo acrescenta que isso ocorre de maneira mais eficaz quando o aprendiz está em um contexto consciente e quando pode construir suas ideias e representá-las no mundo real. Com o advento do computador, passamos a enfrentar não só novos paradigmas no processo educativo, mas também podemos confrontar as teorias de aprendizagem que permeiam o ensino e a aprendizagem. Surgem, assim, novas possibilidades para que os alunos possam materializar seus pensamentos e suas ideias por intermédio da máquina e das novas tecnologias que nos cercam. (CAMPOS, 2019).

Apesar de ter sido pioneiro no uso de computadores na educação, o enfoque principal de Papert não residia na máquina em si, mas sim no aprendiz. Ao utilizar as máquinas, o aprendiz se inseri em um ambiente de aprendizagem aprimorado. O uso do computador como instrumento de instrução não implicava, necessariamente, que a máquina detivesse o controle sobre o aprendiz, mas sim que ela proporcionasse ao aprendiz a capacidade de desenvolver seu próprio pensamento, promovendo uma transição do aprendizado indutivo para a

construção ativa do conhecimento. Dessa forma, o aprendiz exercia controle sobre as máquinas, passando para a construção do conhecimento de forma mais autônoma e significativa (ANTONACHI, 2020).

A partir dessa concepção, o termo "construcionismo" de Papert foi cunhado, derivando da ideia da construção ativa de conhecimento nos moldes propostos por Piaget, mas com um resultado tangível dessa construção. O construcionismo enfatiza não apenas o processo de construção do conhecimento pelo aprendiz, mas também a materialização ou concretização desse conhecimento por meio de projetos e criações tangíveis. Dessa forma, os estudantes não apenas adquirem conhecimentos de forma ativa, mas também aplicam esse conhecimento na criação de algo real, o que reforça ainda mais o entendimento e a conexão com o conteúdo aprendido.

Durante sua investigação dos processos tradicionais de aprendizagem, Papert observou que, geralmente, os aprendizes seguiam primeiro as instruções dos professores e, somente depois disso, partiam para a criação. Essa abordagem gerava inquietação em Papert, uma vez que ele acreditava que o mais importante era exatamente o oposto: os aprendizes deveriam primeiro criar abstrações, utilizando a tecnologia como uma ferramenta, para, então, assimilar o ensino formal de disciplinas como a Física.

Nesse contexto, a proposta era que os estudantes utilizassem a robótica educacional para concretizar conceitos abstratos, como a velocidade média e angular, por meio da aplicação prática da tecnologia, antes de tentarem entendê-los apenas através das explicações no quadro negro da sala de aula. Dessa forma, a construção de conhecimento seria mais significativa e concreta, permitindo aos aprendizes uma compreensão mais profunda dos conceitos e a possibilidade de relacioná-los com suas experiências e criações.

Segundo Papert (1991):

O construcionismo compartilha com o construtivismo a visão de que aprender é "construir estruturas de conhecimento" através da internalização progressiva de ações. [...] Em seguida, adiciona a ideia de que isso acontece de forma especialmente feliz em um contexto onde o aluno está conscientemente empenhado na construção de uma entidade, seja ela um castelo de areia na praia ou uma teoria do universo. (PAPERT, 1991).

A linha de pensamento do construcionismo, proposta por Papert, representa uma abordagem totalmente oposta ao formato tradicional de ensino conhecido como instrucionismo. Valente (1993) já argumentava que o uso do computador no processo

educacional como uma máquina de ensino deveria atuar como suporte, reforço ou complemento ao que ocorre na sala de aula. Enquanto o instrucionismo defende uma abordagem centrada na instrução direta e sequencial, na qual os alunos são receptores passivos de informações, o construcionismo enfatiza a construção ativa do conhecimento pelo próprio aluno.

Nessa perspectiva, o computador é visto como uma ferramenta poderosa que pode auxiliar os estudantes a desenvolverem seu próprio pensamento e criarem projetos significativos. O foco do está na exploração, experimentação e na capacidade dos alunos de aprenderem através da resolução de problemas e da construção de algo concreto, em vez de apenas absorverem conteúdos pré-determinados.

De acordo com Valente (1993), existe um paradigma instrucionista sobre o uso do computador na educação, o qual se divide em duas abordagens: a máquina de ensinar e a máquina para ser ensinada. O autor destaca que, ao utilizar o computador como um recurso educacional, o aluno assume o papel de protagonista, tornando-se o agente promotor de ações em vez de ser um mero espectador. Nesse sentido Papert (1994, p. 125) reflete sobre instrucionismo e construcionismo:

A atitude construcionista no ensino não é, em absoluto, dispensável por ser minimalista – a meta é ensinar de forma a produzir a maior aprendizagem a partir do mínimo de ensino. Evidentemente, não se pode atingir isso apenas reduzindo a quantidade de ensino, enquanto se deixa todo o resto inalterado. A principal outra mudança necessária se assemelha a um provérbio popular africano: se um homem tem fome, você pode dar-lhe um peixe, mas é melhor dar-lhe uma vara e ensiná-lo a pescar.

Na perspectiva da máquina de ensinar, o computador é utilizado como um meio para transmitir informações e conteúdos aos alunos, de forma sequencial e direcionada. Essa abordagem segue uma lógica de instrução direta, em que os estudantes são receptores passivos de conhecimento. Por outro lado, na abordagem da máquina para ser ensinada, o computador é visto como uma ferramenta que permite ao aluno explorar, criar e construir conhecimento de forma ativa. Nesse caso, o estudante tem a liberdade de utilizar o computador como uma plataforma para desenvolver suas próprias ideias, solucionar problemas e participar ativamente do processo de aprendizagem.

Dessa forma, o uso do computador como recurso educacional, quando adotado sob o paradigma da máquina para ser ensinada, coloca o aluno no centro do processo educacional, permitindo-lhe assumir um papel ativo e participativo em sua jornada de aprendizado. Essa

abordagem construcionista enfatiza o protagonismo do estudante, proporcionando uma experiência de aprendizagem mais envolvente, significativa e enriquecedora.

No (Quadro 2) é apresentado a lista de possíveis benefícios advindos da aplicação de metodologias baseadas no construcionismo, discutidas por Orey (2010) no seu livro.

Quadro 2: Benefícios decorrente do uso do construcionismo.

Maior motivação
Os alunos podem escolher os seus temas, a extensão do conteúdo, e o modo de apresentação, e eles mesmos constroem seus projetos para atender seus próprios interesses e habilidades. Esses tipos de atividades são altamente motivadores para os alunos.
O aumento da capacidade de resolver problemas
A metodologia construcionista incentiva os alunos a se envolverem em contextos complexos e mal definidos. Desde o início, os alunos identificam os seus temas e problemas, para então procurar as possíveis soluções. Ao participar de trabalho tanto independente quanto colaborativo, os alunos melhoram suas habilidades de resolução de problemas, desenvolvendo assim suas habilidades de pensamento crítico.
Melhoria das habilidades de pesquisa
O construcionismo oferece uma conexão real com o contexto. Alunos realizam pesquisas utilizando vários recursos de informação e, ao localizar os recursos por si próprios (ao contrário do instrucionismo, onde todas as informações entendidas como necessárias são apresentadas pelo professor), suas habilidades de pesquisa se desenvolvem e melhoram.
Aumento da colaboração
Nas fases de atividades, os alunos criam e organizam seus grupos. Eles compartilham conhecimentos e constroem artefatos colaborativamente. Através da colaboração, eles desenvolvem habilidades de comunicação sociais e obtém múltiplas perspectivas.
Aumento da habilidade de gerenciar recursos
Um curso construcionista bem desenhado oferece aos alunos experiência em organização de projetos e gerenciamento de tempo com recursos apropriados.

Fonte: Adaptado Orey (2010).

Alicerçada na perspectiva desses estudos, a teoria Construcionista de Papert desencadeia o movimento maker, também conhecido como "faça você mesmo". Esse movimento, conforme apontado por Campos (2013), busca integrar uma variedade de projetos que combinam a criação e a engenharia, projetos voltados para temas e desafios específicos, bem como projetos que estimulam o desenvolvimento de histórias e a organização de mostras e campeonatos. Dentro do contexto do movimento maker, os indivíduos são incentivados a colocar a mão na massa, envolvendo-se ativamente na criação e construção de objetos e projetos significativos.

De acordo com Papert (1993), a robótica educacional é uma forma excelente de colocar em prática o pensamento construcionista. Suas pesquisas mostram que crianças que aprendem por meio da robótica conseguem se imaginar no lugar do robô e compreender o funcionamento da programação de computadores. Os participantes do estudo foram capazes de transferir sua compreensão do mundo real para a compreensão da lógica e dos princípios matemáticos. O construcionismo traz para a educação conceitos de design, de engenharia no processo de construção e programação de objetos.

Papert acreditava que a dificuldade que muitas crianças enfrentam para entender certos conceitos está relacionada à falta de materiais do mundo real que demonstrem esses conceitos de maneira clara. Ele defendia que os robôs programáveis são recursos flexíveis e poderosos o suficiente para demonstrar ideias que anteriormente não apresentavam analogias simples no mundo real. Essa abordagem permitia aos alunos uma aprendizagem mais significativa e prazerosa, além de facilitar a compreensão de conceitos complexos de forma mais concreta e tangível.

2.2. Metodologias Ativas

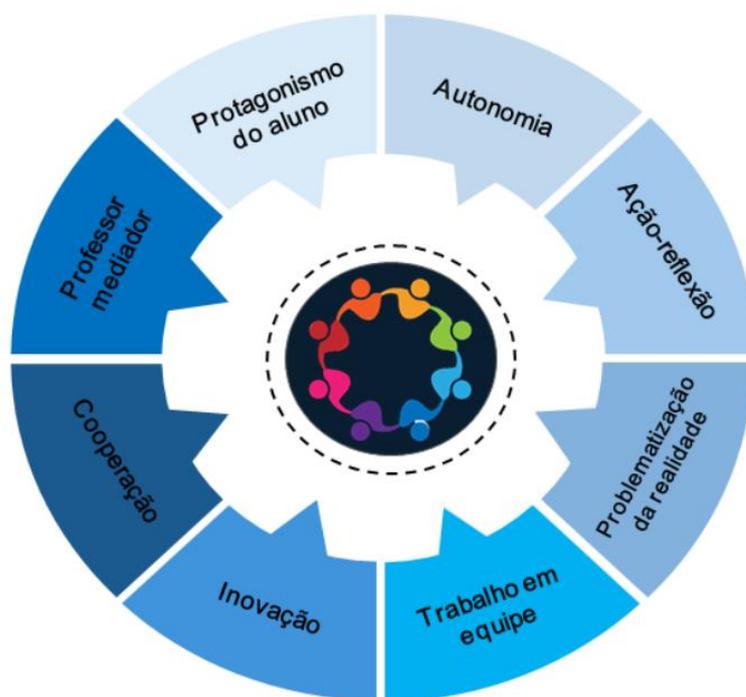
2.2.1. Aprendizagem baseada em problemas

A metodologia de Aprendizagem Baseada em Projetos (ABP) é uma abordagem ativa focada no aluno, ancorada em três fundamentos do construtivismo:

1. Aprendizagem específica do contexto
2. Participação ativa dos estudantes
3. Ênfase nas interações sociais

Conforme Moran (2018), a ABP é uma abordagem metodológica onde os alunos se engajam em tarefas voltadas para a elaboração de um projeto ligado ao seu dia a dia. Como uma abordagem metodológica ativa, a ABP possui certos fundamentos que são ilustrados na (Figura 1).

Figura 1: Fundamentos da ABP.



FONTE: Adaptado de Diesel, Baldez e Martins (2017).

Bender (2014) define a abordagem como um modelo pedagógico onde os estudantes enfrentam questões reais que veem como relevantes, decidindo a melhor forma de lidar com elas e colaborando conjuntamente na busca por respostas, o aluno assume o papel principal na construção do conhecimento.

2.2.2. Aprendizagem baseada em projetos

A Aprendizagem Baseada em Projetos (PBL, do inglês "Project-Based Learning") e a baseada em problemas compartilham semelhanças notáveis, uma vez que ambas focam no aluno como o principal agente na construção do conhecimento. No entanto, existem algumas distinções cruciais entre as duas.

O núcleo da PBL é o projeto. Enquanto a ABP concentra-se na resolução de questões específicas, quando a temática é baseada em projetos pode não necessariamente começar com

um problema a resolver. Em vez disso, pode começar com um objetivo ou produto final em vista e envolver a criação, o design e/ou a invenção de algo (DONNELLY e FITZMAURICE, 2005). Por exemplo, no ensino de programação, pode não ser necessária a resolução de um problema. Os alunos podem ser solicitados a programar uma aplicação com certas características, onde o foco está na construção dessa aplicação e não necessariamente na resolução de um problema específico.

Thomas (2000) ressaltou os benefícios da ABP, afirmando que ela pode transformar a experiência educacional dos alunos. Ela pode incentivar a participação ativa, fortalecer a autoestima e moldar atitudes positivas em relação ao processo de aprendizagem. Quando os alunos estão imersos em um projeto, eles tendem a se envolver mais profundamente, muitas vezes desenvolvendo habilidades que vão além do conteúdo do currículo, como trabalho em equipe, comunicação e pensamento crítico.

2.3. A BNCC e a Programação

Desde meados do Século XX, há um debate em torno da importância de incorporar o ensino da Computação na Educação Básica no Brasil (ELIA, 2021). O surgimento da linguagem de programação "Logo" em 1967, desenvolvida por Seymour Papert, Cynthia Solomon e Wally Feurzeig, desencadeou discussões em diversos países sobre a viabilidade e os motivos para introduzir o ensino da Computação nas escolas (Brasil, 2022). Com o passar do tempo, pesquisadores como Andrea A. DiSessa, por meio do conceito de "Letramento Computacional" (Brasil, 2022), e Jeannette M. Wing, através de seu artigo intitulado "Computational Thinking" (WING, 2006), reavivaram as discussões sobre o assunto. Essas contribuições trouxeram novas perspectivas para o ensino da Computação, ressaltando a importância de desenvolver habilidades e competências relacionadas à compreensão e utilização da tecnologia no contexto educacional. A Base Nacional Comum Curricular (BNCC) é um documento normativo que define uma sequência de aprendizagens essenciais que os alunos do Brasil devem desenvolver na Educação Básica para assegurar os seus direitos de aprendizagem e de desenvolvimento.

No contexto brasileiro, de acordo com Valente (1999 apud Brasil, 2022, p.4), as discussões sobre a introdução do ensino de Computação na Educação Básica surgiram a partir do interesse de educadores de algumas universidades brasileiras, inspirados pelas experiências de outros países como os Estados Unidos da América e a França. Entretanto, foi somente com

a implantação da BNCC no âmbito da Educação Básica (Brasil, 2018) que iniciativas como essas foram completamente incorporadas à Legislação Educacional Brasileira.

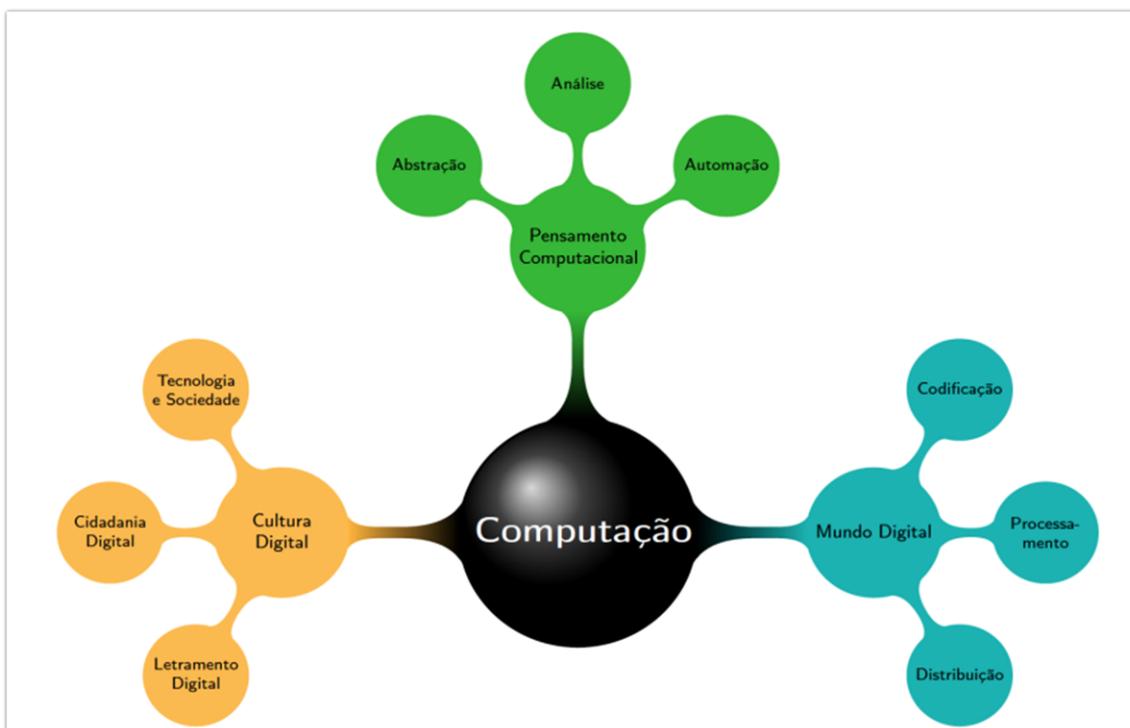
A BNCC estabeleceu como uma das competências do Conselho Nacional de Educação (CNE) a elaboração de normas complementares que contivessem orientações específicas sobre o ensino da Computação na Educação Básica. Essa elaboração foi realizada por meio da Resolução N°1/22 de 4 de outubro de 2022, aprovando as "Normas sobre Computação na Educação Básica - Complemento à BNCC" (Brasil, 2022).

Nunes (2011) questiona: como desenvolver as habilidades fundamentais da era digital (pensamento crítico, resolução de problemas, colaboração, criatividade, ética/responsabilidade) sem a presença da computação na educação? Com a integração do ensino da Computação na BNCC, a educação brasileira passou a reconhecer a relevância da área no desenvolvimento das competências dos estudantes, preparando-os para enfrentar os desafios tecnológicos da atualidade e do futuro. O documento estabelece diretrizes para o ensino de Computação, considerando aspectos como conteúdos, habilidades, metodologias e recursos a serem utilizados em sala de aula, contribuindo para uma educação mais atualizada e alinhada com as demandas da sociedade contemporânea.

A SBC tem apresentado uma sugestão de conjunto de habilidades computacionais a serem desenvolvidas na Educação Básica (Figura 2), essa proposta organiza-se em três eixos principais, conforme listado abaixo:

1. Pensamento Computacional: refere-se à habilidade de compreender, analisar definir, modelar, resolver, comparar e automatizar problemas e suas soluções de forma metódica e sistemática, através do desenvolvimento da capacidade de criar e adaptar algoritmos, aplicando fundamentos da computação para alavancar e aprimorar a aprendizagem e o pensamento criativo e crítico nas diversas áreas do conhecimento.
2. Mundo Digital: envolve aprendizagens sobre artefatos digitais, compreendendo tanto elementos físicos (computadores, celulares, tablets) e virtuais (internet, redes sociais e nuvens de dados). Compreender o mundo contemporâneo requer conhecimento sobre o poder da informação e a importância de armazená-la e protegê-la, entendendo os códigos utilizados para a sua representação em diferentes tipologias informacionais, bem como as formas de processamento, transmissão e distribuição segura e confiável.
3. Cultura Digital: envolve aprendizagens voltadas à participação consciente e democrática por meio das tecnologias digitais, o que pressupõe compreensão dos impactos da revolução digital e seus avanços na sociedade contemporânea; bem como a construção de atitude crítica, ética e responsável em relação à multiplicidade de ofertas midiáticas e digitais, e os diferentes usos das tecnologias e dos conteúdos veiculados; assim como fluência no uso da tecnologia digital para proposição de soluções e manifestações culturais contextualizadas e críticas (Brasil, 2022).

Figura 2: Habilidades computacionais para a Educação Básica.



Fonte: Normas sobre Computação na Educação Básica – Complemento à BNCC.

A equipe do CNE utilizou o conjunto indicativo apresentado pela SBC como ponto de partida e aprimorou as competências e habilidades, levando em consideração as práticas atualmente em vigor no país e estudando iniciativas internacionais relevantes na área de computação na educação. O resultado desse trabalho é um conjunto abrangente de competências e habilidades que engloba todas as etapas da Educação Básica (BRASIL, 2022).

No ensino fundamental, há de se destacar o atendimento à diretriz de compreender a computação como uma área de conhecimento que contribui para explicar o mundo atual e ser um agente ativo e consciente de transformação capaz de analisar criticamente seus impactos sociais, ambientais, culturais, econômicos, científicos, tecnológicos, legais e éticos. No (Quadro 3) é apresentada as competências da computação no ensino fundamental

Quadro 3: competências da computação no ensino fundamental.

COMPETÊNCIAS
1. Compreender a Computação como uma área de conhecimento que contribui para explicar o mundo atual e ser um agente ativo e consciente de transformação capaz de analisar criticamente seus impactos sociais, ambientais, culturais, econômicos, científicos, tecnológicos, legais e éticos.

2. Reconhecer o impacto dos artefatos computacionais e os respectivos desafios para os indivíduos na sociedade, discutindo questões socioambientais, culturais, científicas, políticas e econômicas.
3. Expressar e compartilhar informações, ideias, sentimentos e soluções computacionais utilizando diferentes linguagens e tecnologias da Computação de forma criativa, crítica, significativa, reflexiva e ética.
4. Aplicar os princípios e técnicas da Computação e suas tecnologias para identificar problemas e criar soluções computacionais, preferencialmente de forma cooperativa, bem como alicerçar descobertas em diversas áreas do conhecimento seguindo uma abordagem científica e inovadora, considerando os impactos sob diferentes contextos.
5. Avaliar as soluções e os processos envolvidos na resolução computacional de problemas de diversas áreas do conhecimento, sendo capaz de construir argumentações coerentes e consistentes, utilizando conhecimentos da Computação para argumentar em diferentes contextos com base em fatos e informações confiáveis com respeito à diversidade de opiniões, saberes, identidades e culturas.
6. Desenvolver projetos, baseados em problemas, desafios e oportunidades que façam sentido ao contexto ou interesse do estudante, de maneira individual e/ou cooperativa, fazendo uso da Computação e suas tecnologias, utilizando conceitos, técnicas e ferramentas computacionais que possibilitem automatizar processos em diversas áreas do conhecimento com base em princípios éticos, democráticos, sustentáveis e solidários, valorizando a diversidade de indivíduos e de grupos sociais, de maneira inclusiva.
7. Agir pessoal e coletivamente com respeito, autonomia, responsabilidade, flexibilidade, resiliência e determinação, identificando e reconhecendo seus direitos e deveres, recorrendo aos conhecimentos da Computação e suas tecnologias para tomar decisões frente às questões de diferentes naturezas

Fonte: Anexo ao Parecer CNE/CEB nº 2/2022 (BRASIL, 2022)

Já no ensino médio, uma das premissas diz respeito ao desenvolvimento de projetos para investigar desafios do mundo contemporâneo, construir soluções e tomar decisões éticas, democráticas e socialmente responsáveis, articulando conceitos, procedimentos e linguagens próprias da computação preferencialmente de maneira colaborativa. e (Quadro 4) as competências da computação no ensino médio.

Quadro 4: competências da computação no ensino médio.

COMPETÊNCIAS
<i>1. Compreender as possibilidades e os limites da Computação para resolver problemas, tanto em termos de viabilidade quanto de eficiência, propondo e analisando soluções computacionais para diversos domínios do conhecimento, considerando diferentes aspectos.</i>
<i>2. Analisar criticamente artefatos computacionais, sendo capaz de identificar as vulnerabilidades dos ambientes e das soluções computacionais buscando garantir a integridade, privacidade, sigilo e segurança das informações.</i>
<i>3. Analisar situações do mundo contemporâneo, selecionando técnicas computacionais apropriadas para a solução de problemas.</i>
<i>4. Construir conhecimento usando técnicas e tecnologias computacionais, produzindo conteúdos e artefatos de forma criativa, com respeito às questões éticas e legais, que proporcionem experiências para si e os demais.</i>
<i>5. Desenvolver projetos para investigar desafios do mundo contemporâneo, construir soluções e tomar decisões éticas, democráticas e socialmente responsáveis, articulando conceitos, procedimentos e linguagens próprias da Computação preferencialmente de maneira colaborativa.</i>
<i>6. Expressar e partilhar informações, ideias, sentimentos e soluções computacionais utilizando diferentes plataformas, ferramentas, linguagens e tecnologias da Computação de forma fluente, criativa, crítica, significativa, reflexiva e ética.</i>
<i>7. Agir pessoal e coletivamente com respeito, autonomia, responsabilidade, flexibilidade, resiliência e determinação, identificando e reconhecendo seus direitos e deveres, recorrendo aos conhecimentos da Computação e suas tecnologias frente às questões de diferentes naturezas.</i>

Fonte: Anexo ao Parecer CNE/CEB nº 2/2022 (BRASIL, 2022)

2.4. Robótica Educacional no Ensino de Programação

A programação é uma disciplina que apresenta desafios significativos tanto para os alunos que estão aprendendo quanto para os professores que estão ensinando. Algumas das dificuldades apontadas em pesquisas incluem a falta de contextualização do processo de aprendizagem (FIGUEIREDO e AFONSO, 2006), o método tradicional de ensino que se baseia em leituras e sintaxes específicas da linguagem, e a compreensão abstrata de conceitos básicos de programação, como variáveis, tipos de dados e endereços de memória, que não têm uma representação equivalente na vida real.

Diante desses obstáculos, muitos autores têm procurado alternativas para tornar o aprendizado de programação mais compreensível e eficiente. A robótica é uma dessas alternativas, sendo classificada como facilitadora da aprendizagem de princípios científicos e matemáticos por meio da experimentação de materiais concretos (ROGERS e PORTSMORE, 2004). Além disso, ela incentiva a abordagem de resolução de problemas em sala de aula e promove a aprendizagem cooperativa entre os alunos (ROGERS e PORTSMORE, 2004). A utilização da robótica como ferramenta no ensino de programação pode contribuir para superar alguns dos desafios enfrentados pelos estudantes, tornando o aprendizado mais envolvente, prático e colaborativo. Isso permite que os alunos compreendam melhor os conceitos e princípios da programação, aplicando-os em contextos reais e concretos, o que facilita a assimilação e a retenção do conhecimento.

De acordo com Brito (2019), existem diversas denominações para descrever a utilização da robótica em ambientes educativos, tais como: Robótica Educacional, Robótica Pedagógica, Robótica Educativa e Robótica na Educação. No entanto, no estudo conduzido pelo autor, são identificadas quatro descrições frequentemente associadas à Robótica Educacional: Recurso Tecnológico, Ambiente de Aprendizagem, Criação de Dispositivos Programáveis e Abordagem Pedagógica. O (Quadro 5) detalha cada uma dessas quatro categorizações mencionadas.

Quadro 5: Definições de Robótica Educacional.

Definição	Descrição
Recurso tecnológico	Consiste em um conjunto de ferramentas tecnológicas compostas por componentes reunidos em kits industrializados ou peças reutilizadas das Tecnologias Digitais da Informação e Comunicação (TDIC). Esses recursos são aplicados no contexto pedagógico com a finalidade de intensificar a aprendizagem, promovendo um ensino centrado na interação colaborativa entre os alunos.
Ambiente de aprendizagem	Trata-se de um espaço de aprendizado equipado com computadores, softwares de programação e kits robóticos manipuláveis, incluindo peças reutilizadas. Neste ambiente, sob a supervisão do educador, os alunos têm a oportunidade de montar e programar robôs, visando não só o entendimento da robótica em si, mas também a aquisição de variados conhecimentos.
Criação de dispositivos programáveis	A RE é definida como uma prática educacional que pode ser executada tanto em ambientes escolares quanto em espaços específicos, usando kits de montagem pré-fabricados ou

	materiais reutilizados. Esta abordagem destaca-se pelo projeto, montagem e programação de sistemas robóticos.
Abordagem pedagógica	Um conjunto de abordagens metodológicas ancoradas no uso da tecnologia robótica para facilitar a construção do conhecimento, guiadas pelo docente. Baseia-se na interação entre humano e máquina, visando fomentar a autonomia na aprendizagem científica através de uma pedagogia centrada na experimentação de conceitos.

Fonte: Adaptado de Brito (2019)

Diversas iniciativas têm documentado os benefícios diretos do uso da robótica no ensino de programação de computadores. Uma revisão sistemática da literatura conduzida por Major e colaboradores (2011), identificou mais de 60 estudos que combinaram o ensino de programação introdutória com a robótica, publicados nos últimos 14 anos em congressos e revistas internacionais. A grande maioria desses estudos mostrou resultados positivos e efetivos no processo de aprendizagem de programação quando a robótica foi utilizada em comparação com o ensino tradicional de programação.

Os resultados dessa revisão indicaram que cerca de 68% dos estudos usaram materiais de robótica físicos, como circuitos, placas e fios, enquanto 20% utilizaram simuladores e 12% empregaram ambos. Isso sugere uma preferência popular pelo uso de robótica física em aplicações didáticas. O material físico oferece uma experiência mais tangível e concreta, permitindo que os alunos interajam diretamente com os robôs, o que pode tornar o aprendizado mais envolvente e significativo. A RE tem se mostrado uma poderosa ferramenta no ensino de programação, proporcionando um ambiente de aprendizagem prático, colaborativo e estimulante. Os estudos revisados apontam para resultados positivos no uso da robótica como uma abordagem efetiva para aprimorar o aprendizado de programação em comparação com os métodos tradicionais de ensino.

A preferência por materiais físicos no ensino com robôs, como mencionado por Mauch (2001), está relacionada ao fato de que os alunos muitas vezes veem os robôs como brinquedos e fonte de diversão. Essa abordagem pedagógica utiliza a robótica como uma forma de engajar os estudantes e tornar o processo de aprendizagem mais atraente e lúdico. Com efeito, existe um kit de robótica amplamente empregado não apenas pela comunidade científica, mas também nas escolas, desenvolvido pela Lego, um renomado fabricante de brinquedos de blocos de construção para crianças. De 23 artigos pesquisados, a análise constatou que 14 descrevem a implementação da tecnologia Lego Mindstorms (MAJOR *et al.*, 2011).

Vários pesquisadores têm usado a robótica educacional como ambientes de aprendizagem, podemos listar:

- Fagin e Merkle (2003) e Barnes (2002), usaram na sala de aula robôs para ajudar a ensinar as linguagens de programação Java e ADA. Nos cursos, a principal ênfase foi ensinar as sintaxes e estruturas básicas, com foco nos aspectos de engenharia e mecânica de robôs.
- Goh e Arris (2007), basearam-se na plataforma Lego MindStorms¹ para ministrar curso de construção de robôs com ênfase em competições.
- Chiou (2004), usou a robótica para atacar o crescimento do desinteresse por cursos baseados em matemática, ciências e tecnologia. O autor enfatiza que a implementação da robótica educacional deve ser feita com grande cuidado para evitar o efeito contrário, ou seja, inadvertidamente afastar os alunos interessados.
- Moore (1999), utilizou o tema da robótica para captar a atenção de seus alunos da quinta série, encorajando-os a pensar de forma crítica sobre robôs. Segundo o autor, os alunos foram capazes de construir e programar robôs, entender conceitos de geometria, escrever e compartilhar histórias com colegas, além de comparar sistemas de tecnologia com os sistemas do corpo humano.
- Rogers e Portsmore (2004), também ensinaram jovens estudantes utilizando robôs, e projetaram um currículo utilizando robôs LEGO que ensina alunos da quinta série sobre engenharia.
- Zanetti e Oliveira (2015), discutiram sobre prática pedagógica para o ensino de programação de computadores por meio de RE e Pensamento Computacional. O objetivo foi apresentar uma proposta de ensino que possa amenizar as principais dificuldades de alunos iniciantes em programação, auxiliando nos aspectos relacionados a construção da solução de problemas.
- Silva *et al.* (2021), trabalhou com RE em sala de aula desenvolvendo a montagem e programação de robôs. Para codificá-los usou a abordagem de programação em blocos para facilitar o aprendizado de programação. O experimento foi com alunos do ensino técnico subsequente de informática.

¹ <https://www.lego.com/pt-br/themes/mindstorms>

- Oliveira et al. (2022), realizou um curso online usando a RE e o simulador Tinkercad¹ para prática de programação e simulação dos experimentos, desenvolvendo conceitos de programação, tipos de linguagem e suas principais características, comandos básicos de estruturação e pôr fim a programação em blocos, específica da plataforma.
- No trabalho de dissertação de Torres (2018), a autora investigou como as atividades com RE contribuíram para o desenvolvimento de habilidades como criatividade, noção espacial, lógico-matemático, interpessoal e intrapessoal. Os participantes da pesquisa foram estudantes com idades entre 12 e 16 anos, e as atividades ocorreram no contexto extracurricular. Os resultados do estudo mostraram os benefícios positivos que a RE trouxe para o aprimoramento dessas habilidades nos alunos envolvidos.
- CARDOSO e ANTONELLO (2015), apresentaram os resultados de uma pesquisa que tem teve o objetivo de potencializar o entendimento e a aprendizagem de conceitos básicos da área de algoritmos e programação pelos alunos ingressantes de um curso de Bacharelado em Sistemas de Informação (BSI). Eles usaram dinâmicas de grupo, ferramentas de programação por blocos e Kits de RE alcançando resultados positivos.

Analisando os estudos relacionados ao contexto da dissertação, foi observado que as atividades com robótica podem englobar uma ampla variedade de elementos que contribuem para o desenvolvimento de diferentes competências em diversas áreas. Existem caminhos que possibilitam a utilização significativa da RE em sala de aula, principalmente através da abordagem construtivista que estimula o engajamento dos alunos e o desenvolvimento de habilidades fundamentais para o sucesso em um mundo cada vez mais tecnológico. Stager (2009) explorando o uso do construcionismo em sala de aula, destaca que existem vários caminhos para o uso adequado da RE como:

1. **Robótica como disciplina** – A robótica é ensinada como uma disciplina independente, e competições de robótica populares, como a Competição Brasileira de Robótica², são exemplos desse tipo de abordagem educacional.
2. **Ensinar conceitos STEM específicos** – A robótica é uma ferramenta educacional versátil que possibilita o ensino de diversos conceitos científicos

¹ <https://www.tinkercad.com/>

² <https://www.cbrobotica.org/>

físicos, como máquinas simples, força, torque, potência, fricção e vantagens mecânicas. Também é uma forma prática e interativa de explorar conceitos matemáticos, como frações, variáveis e operações aritméticas.

3. **Ensinar conceitos de computação** – A robótica é utilizada para ensinar conceitos de ciência da computação, como programação, depuração (resolução de erros) e feedback (retorno de sensores). Através da robótica, os alunos têm a oportunidade de aplicar esses conceitos em projetos concretos e desafiadores, tornando o aprendizado mais envolvente e significativo.
4. **Unidades temáticas** – Os estudantes constroem e programam robôs para modelar máquinas e sistemas, como aeroportos, fábricas, parques de diversões ou uma cidade. O objetivo é que, através desses projetos, os conceitos subjacentes de disciplinas escolares tradicionais sejam abordados ou incorporados a esses temas. Dessa forma, a robótica é utilizada como uma ferramenta prática e interativa para promover a aprendizagem integrada, permitindo que os alunos apliquem os conhecimentos teóricos em cenários reais e desafiadores. Ao trabalhar com esses temas, os estudantes adquirem uma compreensão mais profunda dos conceitos em diversas áreas do conhecimento, tornando o processo de aprendizado mais significativo e aplicável à vida cotidiana.
5. **Temas curriculares** – A robótica é empregada como uma ferramenta para abordar problemas específicos relacionados ao currículo formal. Um exemplo ilustrativo seria "Identificar um problema na África Subsaariana e desenvolver um robô para buscar uma solução." Nesse contexto, o foco principal é estimular o pensamento crítico e a criatividade dos alunos ao enfrentarem desafios reais. A solução proposta pode não ser completamente realista, mas o objetivo é que os estudantes explorem a natureza do problema, considerem suas complexidades e apliquem conceitos e habilidades adquiridos para propor abordagens viáveis. A robótica, nesse caso, é utilizada como um veículo para o desenvolvimento de soluções práticas e inovadoras para questões do mundo real.
6. **Estilo livre** – A robótica e a programação de computadores são utilizadas como ferramentas de construção e expressão no laboratório intelectual do aluno. Nesse ambiente, os estudantes têm a liberdade de usar esses materiais

para criar e expressar suas ideias de forma autônoma. Essa abordagem permite que as atividades sejam uma fonte rica para o surgimento de poderosas ideias, à medida que os alunos exploram e experimentam com essas tecnologias de maneira criativa e inovadora.

Apesar de diversos estudos mostrarem perspectivas positivas quanto à aplicação da robótica na educação, essa é uma área de investigação ainda recente, com muitos aspectos pouco explorados. Assim, não está totalmente definido o papel dos robôs no processo educativo. Isso realça a importância de pesquisas, como está, para elucidar questões pendentes sobre a robótica no contexto educacional.

2.5. Software e Hardware Livres

O Movimento Software Livre preza pelo compartilhamento de conhecimento e a solidariedade promovida pela inteligência coletiva, conforme destacado por Silveira (2003). De forma estruturada, surgiu em 1985 com a fundação da FSF¹ (Free Software Foundation), fruto da insatisfação de Richard Stallman do MIT, que se opunha à natureza restritiva dos softwares proprietários (SILVEIRA, 2003). O movimento defende que os usuários devem ter as seguintes liberdades:

1. A liberdade de executar o programa como quiserem.
2. A liberdade de estudar e modificar o programa.
3. A liberdade de redistribuir cópias do programa.
4. A liberdade de distribuir cópias de suas versões modificadas do programa.

O Movimento Software Livre é apoiado e promovido por várias organizações internacionais. Essas entidades desempenham um papel fundamental na promoção, educação, defesa e preservação da filosofia do software livre, entre outras coisas, termos de licenciamento. A licença GNU GPL, formulada pela FSF, é a mais reconhecida. Ela preserva os direitos do criador do software, concedendo, contudo, as liberdades defendidas pelo movimento.

Em uma vertente semelhante, o Movimento Hardware Livre busca democratizar o acesso à tecnologia. O Hardware Livre consiste em equipamentos eletrônicos projetados e

¹ <https://www.fsf.org/>

disponibilizados nos moldes do Software Livre. Um exemplar notável são as placas Arduino (ver 2.6), ferramentas para prototipagem eletrônica.

Os projetos dos componentes eletrônicos das placas Arduino são compartilhados com o público, permitindo que qualquer indivíduo crie dispositivos com base nesses designs. Isso contribui para a difusão e, como resultado, a redução significativa dos custos desses hardwares (SOUZA et al, 2014). Diversos educadores optam por utilizar o Arduino em oposição às plataformas proprietárias. Essa escolha do Arduino em detrimento de plataformas proprietárias é frequentemente impulsionada pela flexibilidade, acessibilidade e comunidade colaborativa que a plataforma oferece. Isso permitiu que o Arduino se tornasse uma das opções mais populares e bem-sucedidas para projetos de hardware e software em todo o mundo.

2.6. Plataforma Arduino

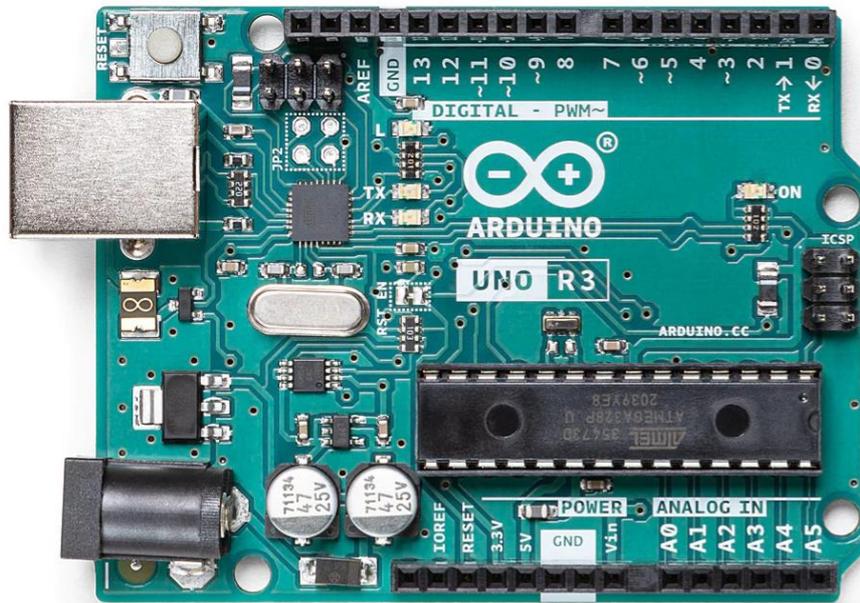
Segundo Oliveira (2015), Arduino é uma plataforma de prototipagem eletrônica de código aberto, projetada para facilitar o desenvolvimento de projetos interativos e controlados por microcontroladores, sendo composta de duas partes: Hardware e Software. Oferece uma linguagem de programação amigável e recursos que possibilitam a elaboração de projetos eletrônicos mesmo para aqueles com pouca ou nenhuma experiência prévia na área.

A plataforma Arduino foi desenvolvida para simplificar a prototipação de projetos interativos. Com ampla adoção entre aficionados, estudantes e especialistas em eletrônica, robótica e automação, sua acessibilidade e adaptabilidade são inquestionáveis. A plataforma facilita a integração de sensores, atuadores e outros componentes eletrônicos, estabelecendo-se como ferramenta essencial em RE e projetos de automação.

Conforme destacado por Silveira (2013), o hardware do Arduino é baseado no microcontrolador Atmel e apresenta uma conexão direta e eficaz com o computador via porta USB. Na (Figura 3), é possível visualizar o modelo Arduino UNO e seus respectivos pinos analógicos e digitais. O software associado, o Arduino IDE (Integrated Development Environment), é um ambiente de desenvolvimento integrado multiplataforma, desenvolvido em Java e liberada sob a licença GPL [GENERAL PUBLIC LICENSE, 2023], que permite programar e enviar comandos à placa Arduino. Depois de transportar a programação para a placa, não será necessário continuar conectado ao computador, pois o sistema funcionará de forma autônoma. Os seus criadores Massimo Banzi, David Cuartielles, Tom Igoe e David

Mellis tinham o objetivo de elaborar um dispositivo barato, funcional e fácil de programar, a fim de ser acessível a estudantes, principiantes e projetistas amadores (ARDUINO TEAM¹).

Figura 3: Arduino Uno.



Fonte: <https://store-usa.arduino.cc/products/arduino-uno-rev3?selectedStore=us>

O Arduino IDE é uma ferramenta fundamental para a criação e desenvolvimento de projetos eletrônicos. Projetada para atender tanto a novos aprendizes quanto a programadores experientes, a Arduino IDE oferece um ambiente intuitivo e amigável para escrever, compilar e carregar código em placas Arduino. Com uma interface de usuário simples, incluindo um editor de código com destaque de sintaxe, facilita a escrita de programas mesmo para aqueles que estão iniciando na programação (WARREN, 2019).

WARREN (2019) relata que uma das principais características da Arduino IDE é sua biblioteca de exemplos e funções pré-programadas. Isso permite que os usuários explorem projetos prontos e reutilizem código para uma variedade de aplicações. Além disso, o recurso de monitor serial integrado oferece a capacidade de monitorar as saídas do Arduino em tempo real, facilitando a depuração e o acompanhamento do comportamento do programa durante a execução. Esse processo eficiente e descomplicado torna a criação de projetos eletrônicos acessível a uma ampla gama de entusiastas e profissionais, contribuindo para a disseminação da inovação e experimentação na eletrônica.

¹ <https://www.arduino.cc/>

2.7. Ambientes Visuais de Programação

A Linguagens de Programação Visual (ou VPL, sigla em inglês para Visual Programming Languages), também conhecida como programação em blocos é uma abordagem popular, especialmente usada em ambientes de aprendizado e ensino de programação para iniciantes (MILNE e LADNER, 2019). Nessa abordagem, em vez de ensinar código em uma linguagem textual, os alunos utilizam interfaces gráficas para selecionar e encaixar blocos pré-definidos que representam funções e ações específicas. A programação em blocos facilita o aprendizado dos conceitos fundamentais da programação, pois elimina a necessidade de entender a sintaxe complexa de uma linguagem de programação textual. Ela reduz a carga cognitiva sobre os estudantes que aprendem sua primeira linguagem de programação (PASTERNAK, 2009), que podem se concentrar na lógica e na sequência das ações, montando o programa de maneira visual e intuitiva.

Ao programar de forma visual, os alunos são incentivados a pensar de forma abstrata, algorítmica e lógica (SILVA e BLIKSTEIN, 2020), ou seja, a decompor um problema complexo em passos menores e sequenciais. Eles aprendem a identificar ações, tomar decisões condicionais e repetir tarefas através de estruturas de controle como loops e condicionantes. É importante ressaltar, no entanto, que a programação em blocos não substitui completamente a programação textual. À medida que os alunos progredem em suas habilidades de programação, é recomendado que eles também aprendam linguagens de programação textuais, que oferecem maior flexibilidade e recursos avançados. Essas habilidades são fundamentais para o desenvolvimento pensamento computacional, pois permitem que os alunos desenvolvam uma abordagem sistemática para resolver problemas. Ao montar blocos em uma sequência lógica, os alunos praticam a habilidade de decompor um problema em partes menores e entender como essas partes se relacionam para alcançar um objetivo maior.

As VPLs surgiram na década de 1960, com melhorias ocorrendo nos anos 1970 e 1980. Seu propósito era proporcionar aos cientistas e entusiastas da computação uma forma mais acessível de programar. Contudo, devido às limitações dos recursos gráficos da época, sua disseminação ficou comprometida. Com o avanço do poder computacional das máquinas, atualmente temos diversas VPLs disponíveis para fins específicos, atendendo a diferentes necessidades (PASTERNAK, 2009).

Na ciência da computação, os ambientes de programação visuais são empregados como uma abordagem diferenciada para o ensino introdutório de programação de computadores em cursos superiores de áreas tecnológicas (MÉLO *et al.*, 2011). Esses ambientes permitem que os estudantes criem programas de maneira mais intuitiva, utilizando elementos gráficos e blocos para representar as estruturas de programação. Essa abordagem busca facilitar o aprendizado e a compreensão dos conceitos de programação, especialmente para aqueles que estão começando a explorar a área. Ao utilizar essas ferramentas, os alunos podem construir algoritmos e soluções de forma mais interativa, estimulando seu interesse e engajamento na disciplina de programação. A programação em bloco é proposta principalmente como uma abordagem para tornar a aprendizagem mais lúdica e motivadora (BARBERO *et al.*, 2011).

As linguagens visuais não são adequadas para todos os usuários ou todas as situações. Muitas vezes, elas são limitadas a um único domínio, o que dificulta sua adaptação para outras áreas e planos de aula dos educadores. No entanto, essas linguagens são poderosas ferramentas para introduzir estudantes de todas as idades à programação e ao pensamento crítico. Ao utilizar essas abordagens visuais, os estudantes podem desenvolver habilidades fundamentais de programação de forma mais acessível e envolvente, independentemente de sua idade ou experiência prévia. Embora possam ter suas limitações, as linguagens visuais oferecem uma oportunidade valiosa para capacitar os alunos a compreenderem a lógica computacional e explorar conceitos essenciais de forma prática e interativa.

2.8. Plataforma Tinkercad

O Tinkercad é uma plataforma Web gratuita onde é possível modelar projetos 3D, simular circuitos eletroeletrônicos e construir blocos de código. É mantida pela AUTODESK, criadora do software de desenho auxiliado por computador mais famoso do mundo: o AutoCAD. A plataforma, dentro da área de projetos de circuitos, disponibiliza diversas bibliotecas de componentes eletrônicos permitindo que sejam simulados vários componentes elétricos e eletrônicos, de forma organizada e lúdica, sem correr risco de danificar os componentes. Dentre as placas de prototipação usadas na REL, encontramos o Arduino e o BBC Micro:Bit, as mais utilizadas em projetos de RE.

Outra grande vantagem do uso do Tinkercad é que é desnecessário ter fisicamente os componentes eletrônicos e as placas para a realização de um experimento. Através da

plataforma conseguimos simular o funcionamento dos projetos, facilitando a fase de prototipagem. Por apresentar uma interface clara, além de ótimos resultados com as simulações, o Tinkercad foi escolhido como plataforma para programação e simulação da capacitação.

A facilidade de uso da plataforma é uma das características marcantes, principalmente quanto ao acesso: para ter uma conta no Tinkercad, basta ter uma conta corporativa educacional com perfil de educador e/ou estudante, cadastro na própria plataforma, conta no Google ou Apple. Na (Figura 4), é exibida a página inicial da plataforma, cujo endereço de acesso é: <https://www.tinkercad.com/>.

Figura 4: Tela inicial do Tinkercad.



Fonte: <https://www.tinkercad.com/>

Outra vantagem do Tinkercad é a opção de programar em blocos. Segundo Kelleher e Pausch (2005), a programação baseada em blocos dispensa a sintaxe, de forma que o programador fique com foco somente na semântica. Por essa característica, programar no Tinkercad se torna mais fácil. A ação de arrastar e soltar os blocos gráficos, corresponde à escrita em linguagem textual. Cada conjunto de blocos tem encaixes específicos possibilitando que o iniciante consiga construir seu algoritmo de forma simples, prática e lúdica.

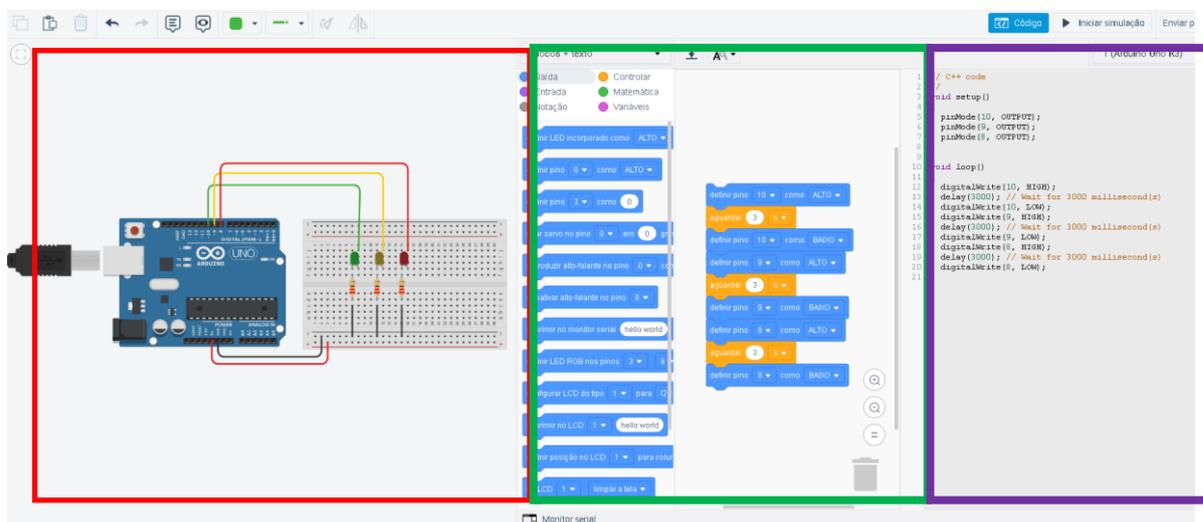
O trabalho publicado por Eryilmaz et al. (2021) enfatiza a relevância do uso do ambiente Tinkercad para inspirar os estudantes no desenvolvimento do pensamento computacional. As habilidades relacionadas ao raciocínio lógico e à criatividade, por meio do Tinkercad, também são abordadas na pesquisa conduzida por Abburi et al. (2021). Além disso, o artigo de Mohapatra et al. (2020) demonstra o uso do Tinkercad como uma ferramenta motivacional para encorajar os estudantes a trabalharem em projetos

colaborativos, especialmente em atividades desenvolvidas em grupos para resolver problemas reais.

Essas pesquisas destacam o valor do mo uma plataforma eficaz no contexto educacional, impulsionando a motivação dos alunos para aprenderem conceitos computacionais de maneira interativa e prática. O ambiente oferece oportunidades para o desenvolvimento de habilidades essenciais, como o pensamento lógico, a criatividade e a colaboração, além de proporcionar uma experiência de aprendizagem mais envolvente e significativa para os estudantes.

Na (Figura 5), observamos a experiência de um circuito que controla um semáforo. No lado esquerdo, representado pela área na cor vermelha, encontra-se a montagem do circuito em uma placa de ensaio, com vários componentes e o Arduino. Na parte central, está a programação em Blocos, dentro da área verde, e, em destaque da cor lilás, é representado o resultado da linguagem textual produzida pela plataforma, no lado direito. Essa codificação resultante é transferida para o Arduino via o software IDE Arduino, para que a placa funcione, de forma autônoma, como um controlador de semáforo.

Figura 5: Projeto de um semáforo.

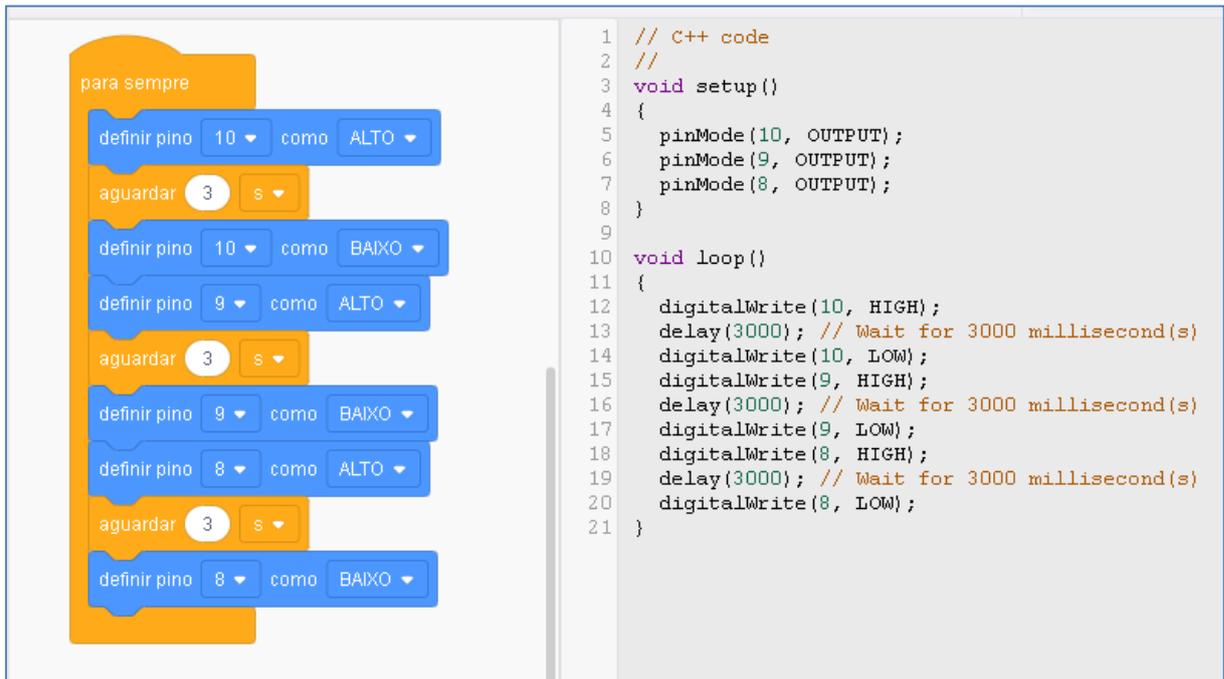


Fonte: <https://www.tinkercad.com/>

É relevante ressaltar que o pensamento computacional experimentado através do Tinkercad é facilitado pelo uso de blocos funcionais, em que suas cores representam instruções específicas do software de forma altamente intuitiva. Na programação em blocos, de um lado temos os blocos funcionais que representam instruções específicas do software de forma visual e intuitiva. Do outro lado, temos o código em texto, que é a representação em

linguagem de programação tradicional das mesmas instruções presentes nos blocos. Essa abordagem híbrida oferece aos usuários a flexibilidade de escolher entre a programação visual com blocos ou a programação textual, permitindo que eles trabalhem de acordo com suas preferências e níveis de habilidade em programação. A (Figura 6) mostra uma representação dessa diferença.

Figura 6: Representação visual e textual de um código.



Fonte: Autor (2023).

A escolha do Tinkercad como plataforma principal para a pesquisa foi baseada em várias razões. Primeiramente, trata-se de uma plataforma online acessível, o que significa que os participantes não precisam instalar nenhum software adicional em seus dispositivos (MOHAPATRA *et al.*, 2020). Isso facilita o acesso e elimina preocupações com a compatibilidade do sistema operacional utilizado por cada participante. Além disso, a capacidade de simular o funcionamento dos projetos foi outro ponto importante considerado. Os participantes podem testar e verificar o comportamento de seus circuitos virtualmente, sem a necessidade de ter os componentes físicos em mãos. Essa funcionalidade economiza tempo e recursos, permitindo que eles experimentem diferentes configurações e aprimorem seus projetos antes de construí-los fisicamente. Essas vantagens tornam o Tinkercad uma escolha ideal para a pesquisa, proporcionando uma experiência prática e interativa de aprendizado em

RE e programação, de forma virtual antes de se comprometerem com a construção física dos projetos.

Ao serem introduzidos ao Tinkercad, os alunos aprenderam a criar projetos eletrônicos utilizando a programação visual em blocos. Eles simplesmente arrastaram e soltaram os blocos de código, conectando-os para criar a lógica do projeto e simular o funcionamento do circuito eletrônico virtualmente. Segundo Kelleher e Pausch (2005), a programação baseada em blocos dispensa a necessidade de se preocupar com a sintaxe, permitindo que o programador se concentre apenas na semântica das instruções. Essa abordagem de programação baseada em blocos tem atraído um interesse crescente, principalmente em relação ao aprendizado precoce e ao ensino de programação para crianças (AHMAD *et al.*, 2021). Ela torna o processo de aprendizado mais acessível e envolvente, permitindo que os alunos criem projetos e resolvam problemas de forma prática e interativa, mesmo sem experiência prévia em programação. A simplicidade da programação em blocos no Tinkercad oferece uma experiência lúdica e desafiadora, possibilitando que os alunos desenvolvam habilidades de pensamento computacional de maneira mais intuitiva e divertida.

2.9. Modelo de Aceitação da Tecnologia

O Modelo de Aceitação de Tecnologia (TAM), também conhecido como Technology Acceptance Model, é um dos modelos aplicado para analisar a receptividade de tecnologia por parte dos usuários. Foi originalmente desenvolvido por Davis (1989) para examinar a adoção da tecnologia da informação. Naquela época, as empresas buscavam avaliar se o investimento em novas tecnologias de computação valeria a pena, em um cenário anterior à atual internet e antes do Windows 3.1. A utilização de tecnologia específica era um fator crítico para avaliar a produtividade. Ter um indicador confiável e válido que pudesse explicar e prever o uso da tecnologia seria valioso tanto para fornecedores de software quanto para gerentes de TI.

O TAM foi fundamentado na Teoria da Ação Racional (TRA), a qual explora as reações e percepções de um indivíduo diante de ações. O modelo, representado na (Figura 7), é definido por dois construtos principais: Utilidade Percebida (UP) e Facilidade de Uso Percebida (FUP). Base do modelo (DAVIS, 1989):

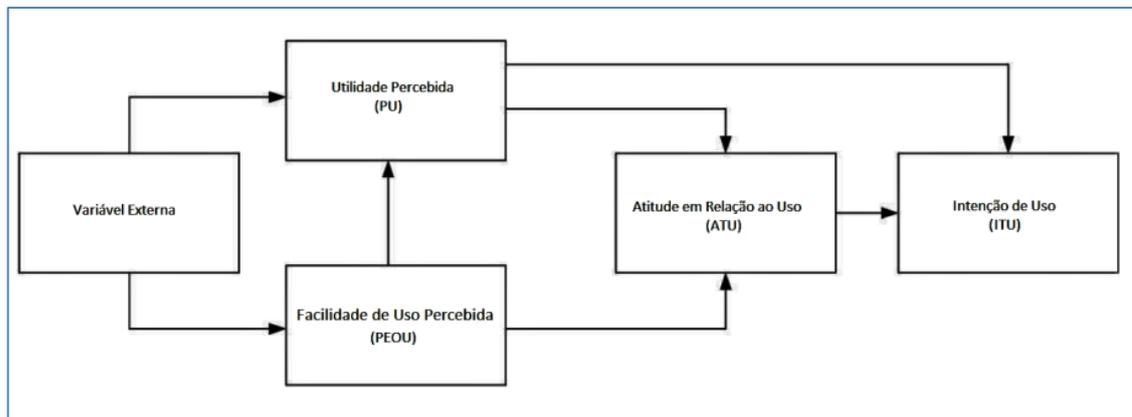
- a) Utilidade Percebida: Refere-se ao grau em que um usuário acredita que o uso de uma tecnologia específica melhorará seu desempenho no trabalho ou na

realização de uma tarefa. Em outras palavras, é a extensão em que a tecnologia é vista como benéfica ou valiosa para o usuário.

- b) Facilidade de Uso Percebida: Refere-se ao grau em que um usuário acredita que o uso de uma tecnologia será livre de esforço. Isso não se trata apenas da interface do usuário, mas também da curva de aprendizado, suporte e outros aspectos que podem tornar uma tecnologia mais ou menos desafiadora para interagir.

A variável externa considera a primeira experiência do usuário com o sistema. Essas primeiras impressões tendem a ser determinantes para determinar se alguém continuará a utilizar a tecnologia.

Figura 7: Modelo de Aceitação de Tecnologia.



Fonte: Davis (1989).

Portanto, o Modelo de Aceitação de Tecnologia oferece uma estrutura valiosa para compreender como as pessoas reagem e interagem com novas tecnologias, destacando a importância das percepções sobre utilidade e facilidade de uso como determinantes fundamentais na adoção e utilização contínua de sistemas tecnológicos.

No TAM, a percepção da utilidade é fundamental. Conforme Davis (1989) observou, “os usuários geralmente estão dispostos a lidar com alguma dificuldade de uso em um sistema que fornece funcionalidades criticamente necessárias”. Isto é ilustrado na (Figura 5) do TAM, onde a facilidade percebida de uso influencia a utilidade, bem como as atitudes em relação ao uso.

Brito e Ramos (2019) relatam que mesmo com a extensa aplicação do TAM, a ausência de referências publicadas para as pontuações globais torna desafiador determinar se um produto (ou tecnologia) atingiu uma pontuação que indique sua aceitação. Sem parâmetros

claros, é complicado avaliar se potenciais ou atuais usuários veem o produto como benéfico e, conseqüentemente, se optarão por adotá-lo ou continuar sua utilização .

O TAM tem sido amplamente aplicado em diversos contextos para estudar a adoção de tecnologia, como a adoção de aplicativos móveis, sites de comércio eletrônico e software empresarial. Também foi estendido e modificado de várias maneiras para levar em consideração fatores adicionais que podem influenciar a aceitação de tecnologia, como variáveis externas, como influência social e condições facilitadoras.

Algumas extensões importantes e modelos relacionados ao TAM incluem:

1. TAM2: Uma versão estendida do TAM que inclui fatores adicionais como norma subjetiva (influência social) e imagem (VENKATESH e DAVIS, 2000).
2. Teoria Unificada da Aceitação e Uso de Tecnologia (UTAUT): Um modelo que integra várias teorias de aceitação de tecnologia, incluindo o TAM, e incorpora variáveis adicionais como expectativa de desempenho, expectativa de esforço, influência social, condições facilitadoras e gênero (VILAR, 2013).
3. TAM3: Uma extensão adicional do TAM que inclui variáveis relacionadas ao aspecto hedônico do uso de tecnologia e o papel do prazer percebido (SETIYANI, 2021).

O TAM e suas extensões têm sido ferramentas valiosas para pesquisadores e profissionais na compreensão e previsão da adoção e aceitação de tecnologia. Eles fornecem insights sobre o comportamento do usuário e ajudam as organizações a projetar e implementar tecnologias que são mais propensas a serem adotadas por seus usuários-alvo.

3. ESTUDO EXPERIMENTAL

Neste capítulo, descrevemos a metodologia escolhida para desenvolver o trabalho. Conforme mencionado anteriormente (ver 1.2), este estudo visa compreender como a Robótica Educacional pode atuar como uma ferramenta de apoio na Programação.

Tópicos que abrangem o escopo deste capítulo:

3.1 Oficina em Programação e Robótica

3.2 Descrição da pesquisa

3.1. Oficina em Programação e Robótica

A oficina de programação e robótica foi concebida como um elemento-chave da estratégia de pesquisa, visando aprofundar nosso entendimento em um campo emergente e dinâmico. A ideia central era investigar como a interação prática com a programação e a robótica poderia influenciar, possivelmente, revolucionar os métodos tradicionais de ensino e aprendizado. A oficina não foi apenas uma oportunidade de ensino, mas também uma ferramenta de observação e análise. Através dela, buscávamos entender como os participantes reagiam, adaptavam-se e assimilavam os conceitos transmitidos, e qual o potencial dessa abordagem para a construção de um aprendizado significativo.

Importante informar que a oficina teve duração de 2 meses com participação de 20 alunos do ensino médio de diversas series, com encontros presenciais de duração média de 3 (três) horas, totalizando 24 (vinte e quatro) horas de atividades distribuídas em 3 (três) módulos distintos, segundo disponibiliza o (quadro 3.1).

Quadro 6: Modulo da Oficina.

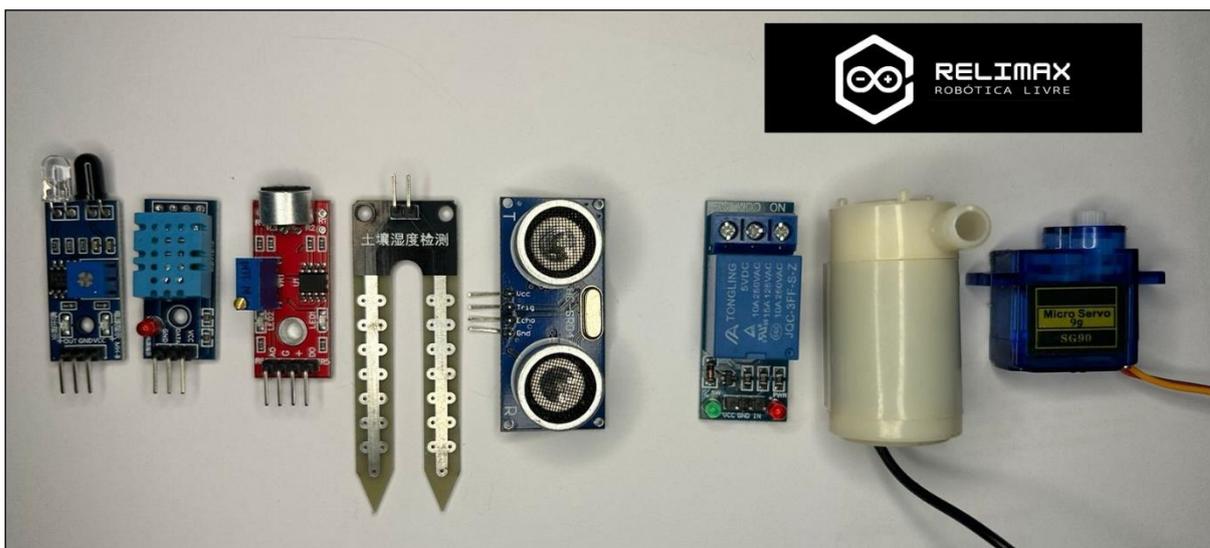
Módulo I	
Descrição	Conteúdos
Introdução a Robótica e Eletrônica (APÊNDICE A)	a) O que é robótica; b) Componentes eletrônicos e Eletricidade Básica c) Placa de Prototipação
Módulo II	
Descrição	Conteúdos
Introdução a Programação (APÊNDICE B)	a) O Ambiente de Desenvolvimento IDE Arduino b) A Programação Visual em Blocos com o Tinkercad c) Uso de Sensores e Atuadores
Módulo III	
Descrição	Conteúdos
Projetos Finais	a) Lixeira Automática (APÊNDICE C) b) Robô Papa-Tudo (APÊNDICE D) c) Irrigador de Plantas (APÊNDICE E) d) Robô Sensor Movimento (APÊNDICE F)
AValiação (APÊNDICE D)	
Descrição	Conteúdos
Aplicação de Questionários	1. Levantamento de dados 2. Aceitação de Tecnologia

Quadro 1: Elaborado pelos autores.

Foi utilizada a metodologia da sala de aula invertida para promover um ambiente de aprendizado mais dinâmico e participativo (BERGMANN e SAMS, 2018). Antes das aulas presenciais, os alunos tiveram acesso ao conteúdo por meio de materiais de estudo em forma de textos disponibilizados, para que pudessem se familiarizar com os conceitos e adquirir o conhecimento básico.

Essa abordagem permitiu que os estudantes explorassem o conteúdo em seu próprio ritmo, podendo revisar e aprofundar os tópicos de acordo com suas necessidades individuais. Eles puderam fazer anotações, destacar pontos importantes e refletir sobre o material antes das discussões nos encontros presenciais. Como recurso, os alunos tiveram acesso aos componentes eletrônicos sensores e atuadores, conforme (Figura 8), no Arduino e num computador com acesso à internet para utilização de uma plataforma de prototipagem e codificação.

Figura 8: Sensores e atuadores.



Fonte: Projeto RELIMAX.

Durante as aulas presenciais, o foco foi direcionado para atividades interativas e colaborativas. Os alunos foram divididos em grupos de trabalho, onde puderam discutir, resolver problemas e aplicar os conceitos aprendidos, a metodologia ativa ABP foi importante neste contexto. O instrutor e os bolsistas estiveram presentes para esclarecer dúvidas, fornecer orientações e promover discussões enriquecedoras. O emprego desse modelo permitiu uma maior participação ativa dos alunos, incentivando-os a se envolverem mais no processo de aprendizagem. Além disso, os estudantes desenvolvem habilidades de trabalho em equipe,

comunicação e pensamento crítico, pois são desafiados a aplicar o conhecimento adquirido em situações reais, ressalta Bergmann e Sams (2018).

Ao final das aulas os grupos realizaram rodas de conversa com objetivo de refletir sobre o aprendizado adquirido, compartilhar experiências e feedbacks individuais. Esse momento proporciona uma oportunidade valiosa para que os alunos expressem suas opiniões, dúvidas e sentimentos em relação ao conteúdo abordado. Além disso, a roda de conversa fortalece o senso de comunidade entre os participantes, encorajando a troca de ideias e a colaboração contínua. É também uma ferramenta pedagógica eficaz para os educadores avaliarem a eficácia de suas estratégias de ensino e fazerem os ajustes necessários no planejamento das próximas aulas. Gomes e Alves (2020), elucidaram que a coleta de dados é facilitada através da utilização de rodas de conversa, tornando-se uma ferramenta valiosa na pesquisa qualitativa.

O módulo III teve como objetivo avaliar o entendimento e a compreensão dos alunos em relação aos assuntos abordados na oficina, utilizando o ambiente Tinkercad. Para isso, foram realizados os seguintes experimentos práticos:

Lixeira Automática: Neste experimento, os alunos desenvolveram um sistema automatizado para a abertura e fechamento de uma lixeira. Eles utilizaram sensores para detectar a presença de objetos e atuadores para controlar o mecanismo de abertura e fechamento da lixeira. O objetivo foi explorar o uso de sensores e atuadores em um contexto prático e desenvolver habilidades de programação para controlar o sistema.

Robô Papa-Tudo: Neste experimento, os alunos criaram um robô capaz de recolher objetos colocados em sua frente e depositá-los em local específico. Eles utilizaram servomotores, sensores e programação para controlar os movimentos do robô e a detecção de objetos. O objetivo foi aprimorar a compreensão de conceitos de movimentação e interação com o ambiente, bem como desenvolver habilidades de programação para controlar o robô.

Irrigador de Plantas: Neste experimento, os alunos desenvolveram um sistema automático de irrigação para plantas. Eles utilizaram sensores para monitorar a umidade do solo e atuadores para controlar o fluxo de água. O objetivo foi explorar a aplicação de sensores e atuadores em um contexto de cuidado com as plantas, bem como aprimorar habilidades de programação para controlar o sistema de irrigação.

Robô com Sensor de Movimento: Neste experimento, os alunos criaram um robô com um sensor de movimento que era capaz de detectar a presença de pessoas ou objetos em sua frente. Eles utilizaram o sensor de movimento, servomotores, pets reciclados e programação para controlar os movimentos do robô de acordo com as detecções realizadas. O objetivo foi explorar o uso de sensores de movimento em aplicações robóticas e desenvolver habilidades de programação para interação com o ambiente.

É verdade que a questão de custos foi levada em consideração na hora de construção da oficina e, dentro desse contexto foram adotadas as soluções livres tanto para o hardware quanto para o software, como ilustra o (Quadro 7).

Quadro 7: Soluções livres da oficina.

Ferramenta/Plataforma	Descrição
Arduino / IDE	É uma plataforma de prototipagem eletrônica de código aberto, projetada para facilitar o desenvolvimento de projetos interativos e controlados por microcontroladores, sendo composta de duas partes: Hardware e Software (OLIVEIRA, 2015).
Tinkercad	É uma plataforma virtual, aberta e gratuita que permite a prática de simulação, montagem e a programação de protótipos de robótica e circuitos eletrônicos digitais (OLIVEIRA <i>et al.</i> , 2022).

Fonte: Elaborado pelos autores.

Ao final do curso, foi desenvolvido um questionário. As perguntas formuladas podem ser encontradas no APÊNDICE G. O questionário visa coletar dados relevantes para o estudo de caso em questão. Por fim, foi elaborado questionário aplicado ao final do curso. No (APÊNDICE D) 3 estão listadas as perguntas. O primeiro questionário pretende fazer o levantamento de dados pertinentes ao estudo em causa.

3.2. Descrição da pesquisa

Ensinar e aprender programação de computadores é um desafio significativo tanto para professores quanto para alunos. A jornada pedagógica associada à programação pode ser complexa, conforme evidenciado em estudos anteriores que destacam as principais barreiras enfrentadas nesse campo. Diversas estratégias têm sido propostas para superar esses obstáculos. Uma dessas estratégias é a incorporação da robótica, que tem se mostrado

particularmente envolvente, pois apresenta em tempo real as respostas dadas aos robôs. Dentro dessa perspectiva, sugere-se uma atividade que introduz os alunos aos conceitos básicos de programação, proporcionando-lhes uma prática com uma linguagem de programação baseada em blocos e a robótica livre.

Durante a realização deste projeto investigativo, houve dedicação para garantir precisão, validade nas situações propostas e uma documentação detalhada de todo o processo. A análise foi baseada em dados coletados por meio de questionários e rodas de conversas, sempre com o cuidado de não permitir que os preconceitos pessoais do pesquisador afetassem os desfechos. Afinal, o estudo desenvolvido tenta responder a questão sob investigação dessa pesquisa :

- **A Robótica Educacional auxilia no processo de aprendizagem da programação?**

Campos (2017) argumenta que a RE representa uma evolução na forma como os conceitos de ciência, tecnologia, engenharia e matemática são ensinados, promovendo o aprendizado através da prática, interatividade e resolução de problemas. Contudo, como qualquer inovação tecnológica, sua eficácia não é determinada apenas pelas capacidades inerentes ao recurso, mas também pela sua aceitação entre educadores e alunos. Em relação à hipótese listada, espera-se entender melhor como as atitudes e percepções dos usuários influenciam diretamente a aceitação e a utilização da RE.

4. ESTUDO DE CASO E DISCUSSÕES

Neste capítulo, expomos e examinamos os resultados obtidos. Através do questionário, utilizamos a escala Likert nas opções de múltipla escolha propostas aos alunos. O questionário coletou dados para análise do estudo de caso proposto para investigação, o modelo foca na RE como uma ferramenta de suporte ao ensino de programação. A amostra é composta por 20 alunos, sendo 12 homens e 8 mulheres, de diversas turmas do Ensino Médio.

Tópicos que abrangem o escopo deste capítulo:

4.1 Estudo de caso

4.2 Discursão

4.1. Estudo de caso

O questionário desenvolvido para a pesquisa foi estruturado em três seções: dados sociodemográficos (informações do usuário), dados relacionados à prática inicial e, por último, dados associados ao resultado da aplicação da RE. A seção sociodemográfica englobou perguntas sobre gênero, idade e série em que o aluno está matriculado. As outras seções abordaram experiências relacionadas, conforme apresentado no (Quadro 7).

Quadro 7: Instrumento de pesquisa.

<p>DADOS SOCIODEMOGRÁFICOS</p> <p>Qual o seu nome ou e-mail ?</p> <p>Qual a sua idade</p> <p>Qual o seu gênero?</p> <p>Você cursa que série no COLUN ?</p> <p>Já tinha participado de alguma capacitação em Robótica?</p> <p>Você já tinha programado alguma vez?</p>
<p>PRÁTICA INICIAL</p> <p>1) Você compreendeu o propósito da tarefa?</p> <p>2) Teve dificuldade em programar o circuito para que ele operasse como um semáforo?</p> <p>3) Na hora de simular o circuito na plataforma Tinkercad funcionou tudo normalmente?</p> <p>4) Na hora de montar o circuito na placa de montagem houve problemas?</p> <p>5) Conseguiu entender o funcionamento dos componentes eletrônicos, sensores e atuadores?</p>
<p>RESULTADOS</p> <p>6) A prática realizada auxiliou na clareza de novos conceitos de programação, você compreendeu o propósito da tarefa?</p> <p>7) Os conceitos de programação em bloco abordados foram adequados para a tarefa proposta?</p> <p>8) O uso da Plataforma de Simulação e a programação em blocos facilitou sua compreensão na programação?</p>

- 9) Você acredita que é mais simples aprender programação através de uma linguagem textual?
- 10) Você acredita que é mais simples aprender programação usando os blocos ?
- 11) Você acredita que é mais simples iniciar um projeto de robótica programando com blocos?
- 12) O uso da Robótica contribuiu para a sua compreensão nos princípios de programação?
- 13) Você considera que a robótica torna a programação mais envolvente e divertida?
- 14) Você considera que a montagem do semáforo foi relevante para sua aprendizagem?

Fonte: Elaborado pelos autores

A escala de Likert foi usada para oferecer uma ferramenta quantitativa e estruturada, permitindo aos participantes expressarem o grau de concordância ou discordância em relação a uma série de afirmações. Essa escala facilita a coleta de opiniões e sentimentos de maneira gradativa, proporcionando insights mais detalhados sobre as percepções e atitudes dos respondentes em relação ao tema investigado. Além disso, sua aplicação padronizada permite uma análise comparativa e estatística dos dados coletados, garantindo maior confiabilidade e precisão nas interpretações. Vilar (2013) declara que quando se trata de avaliar atitudes e opiniões, a escala de Likert é frequentemente a escolha preferida de muitos pesquisadores. Para esta pesquisa, a escala está representada entre:

- a) 1 = Discordo Totalmente
- b) 2 = Discordo Parcialmente
- c) 3 = Neutro
- d) 4 = Concordo Parcialmente
- e) 5 = Concordo Totalmente

Ainda em relação ao questionário, ele foi construído com o apoio da ferramenta Google Formulários¹, uma plataforma online que permite a criação de inquéritos personalizados de forma simples e intuitiva. Uma estratégia utilizada como termômetro foi a roda de conversa, que se provou ser uma ferramenta de apoio inestimável durante o processo.

¹ <https://workspace.google.com/>

Estas sessões proporcionaram um espaço aberto para discussão, permitindo que os participantes compartilhassem suas experiências, ideias e preocupações de maneira orgânica e colaborativa (GOMES e ALVES, 2020). Além de gerar insights valiosos para o projeto, as rodas de conversa também fortaleceram o trabalho de equipe. Através destes encontros, foi possível obter feedback em tempo real, identificar áreas de melhoria e ajustar o rumo da iniciativa conforme as necessidades do grupo.

Entre os participantes da oficina, apenas 15% já haviam participado de uma capacitação em RE, e 35% já haviam participado de um treinamento sobre programação. Para o propósito da oficina esse percentual pode ser elevado já que se busca apresentar a RE como uma ferramenta para apresentar os fundamentos da programação, (Figura 9).

Figura 9: Questionário 1 – Capacitação e Programação.

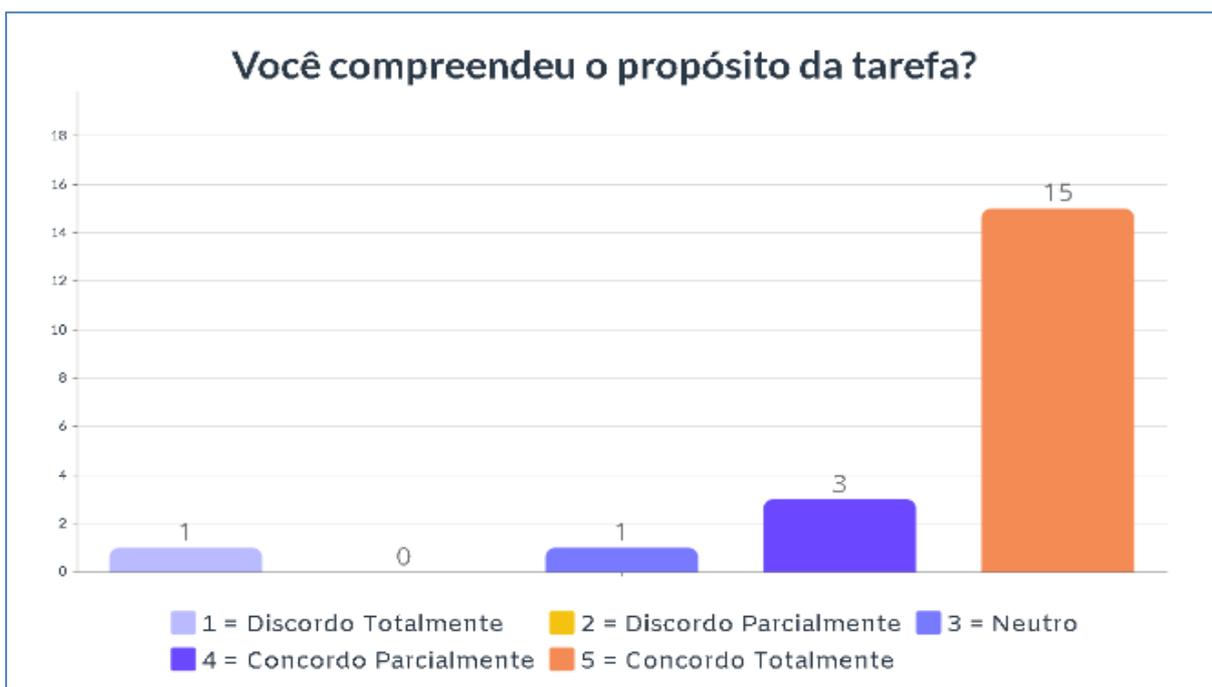


Fonte: Elaborado pelos autores.

Antes de responder ao questionário, os alunos foram desafiados a montar um semáforo tanto no simulador quanto com componentes reais. A programação no Arduino deveria ser elaborada de forma que o comportamento do circuito replicasse um semáforo real. Na prática, os participantes utilizaram: Placa de prototipação, Arduino nano, led's, resistores de 220R e cabinhos.

Em relação ao entendimento da tarefa, a grande maioria conseguiu ter compreensão do objeto que estaria a construir e programar (Figura 10). Kaminski e Boscaroli (2019) afirmam que o ensino de programação propicia o desenvolvimento de habilidades importantes no que se refere à formação de um cidadão crítico, atuante e participativo.

Figura 10: Questionário - Gráfico 1.



Fonte: Elaborado pelos autores.

O Tinkercad oferece um paradigma de programação visual, atrativo e de fácil compreensão, que possibilita aos usuários, especialmente aos iniciantes, uma experiência imersiva e intuitiva no mundo da codificação, tornando o aprendizado mais lúdico e envolvente. Deste modo, a programação em blocos ganha mais adeptos a cada dia, principalmente devido à sua facilidade. Nessa perspectiva, 55% dos participantes relataram não ter dificuldade na programação do semáforo (Figura 12). A mesma tendência foi observada quando questionados na pergunta 3 "na hora de simular o circuito na plataforma Tinkercad, tudo funcionou normalmente?". 80% afirmaram que tudo funcionou sem problemas.

Figura 11: Questionário – Gráfico 2.



Fonte: Elaborado pelos autores.

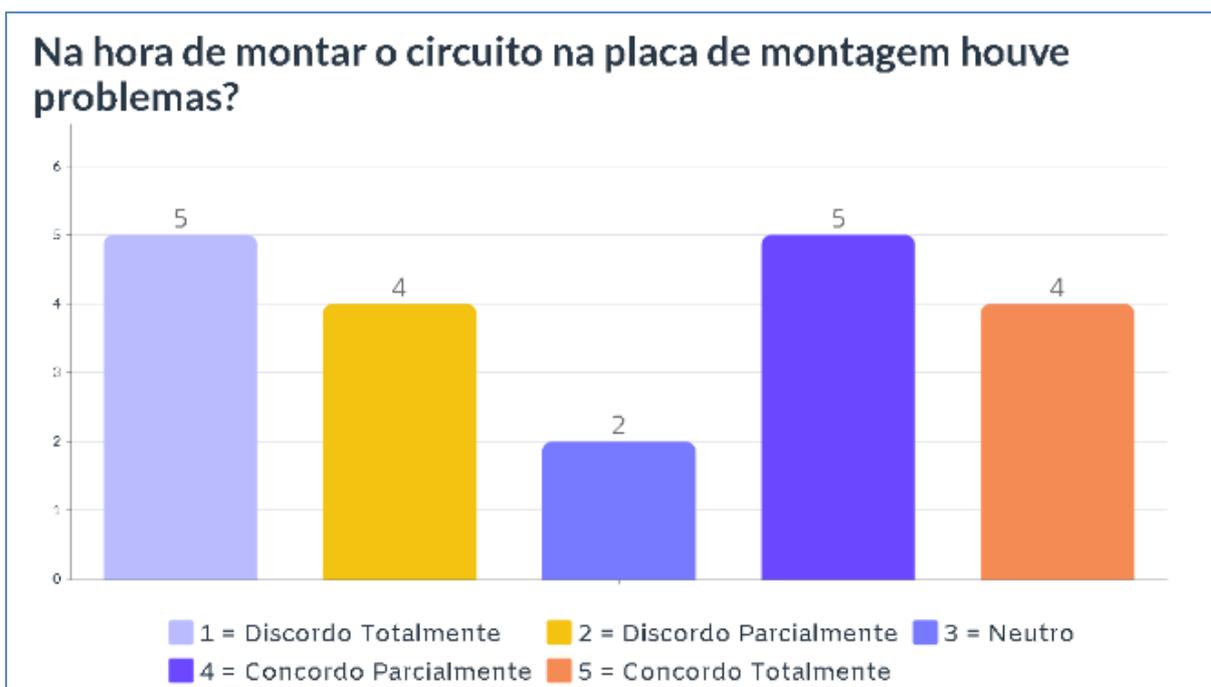
Na hora de montar o circuito na placa, replicando o que foi simulado no Tinkercad, a complexidade aumenta significativamente, especialmente para aqueles que estavam iniciando na área. Esse salto do virtual para o prático pode apresentar desafios inesperados, exigindo dos novatos uma adaptação e compreensão mais aprofundada dos componentes e suas interações no mundo real. 55% dos estudantes relatam dificuldade nesta fase (Figura 12).

Ao participarem da roda de conversa, os alunos relataram que o primeiro contato com as placas e os componentes eletrônicos foi bastante desafiador. A primeira transferência da programação para o Arduino revelou-se uma tarefa complexa, em que muitos sentiram a diferença entre a simulação e a prática real. Muitos se depararam com detalhes e nuances não presentes no ambiente virtual, como a correta conexão dos pinos, a integridade dos componentes e a precisão na montagem do circuito do semáforo. Esse processo ressaltou a importância do entendimento prático e da experimentação direta, para além da teoria e da simulação. Apesar das dificuldades iniciais, essa experiência foi crucial para o amadurecimento e a aprendizagem efetiva dos estudantes no mundo da eletrônica e programação.

À medida que avançavam na atividade, muitos alunos começaram a valorizar a importância da persistência e da resolução de problemas em tempo real. A interação direta

com os componentes físicos proporcionou momentos de insight e compreensão mais profunda sobre o funcionamento dos circuitos e da programação do Arduino. Além disso, trabalhar com desafios práticos fortaleceu a coesão do grupo, incentivando a colaboração e o compartilhamento de soluções entre os participantes.

Figura 12: Questionário – Gráfico 3.

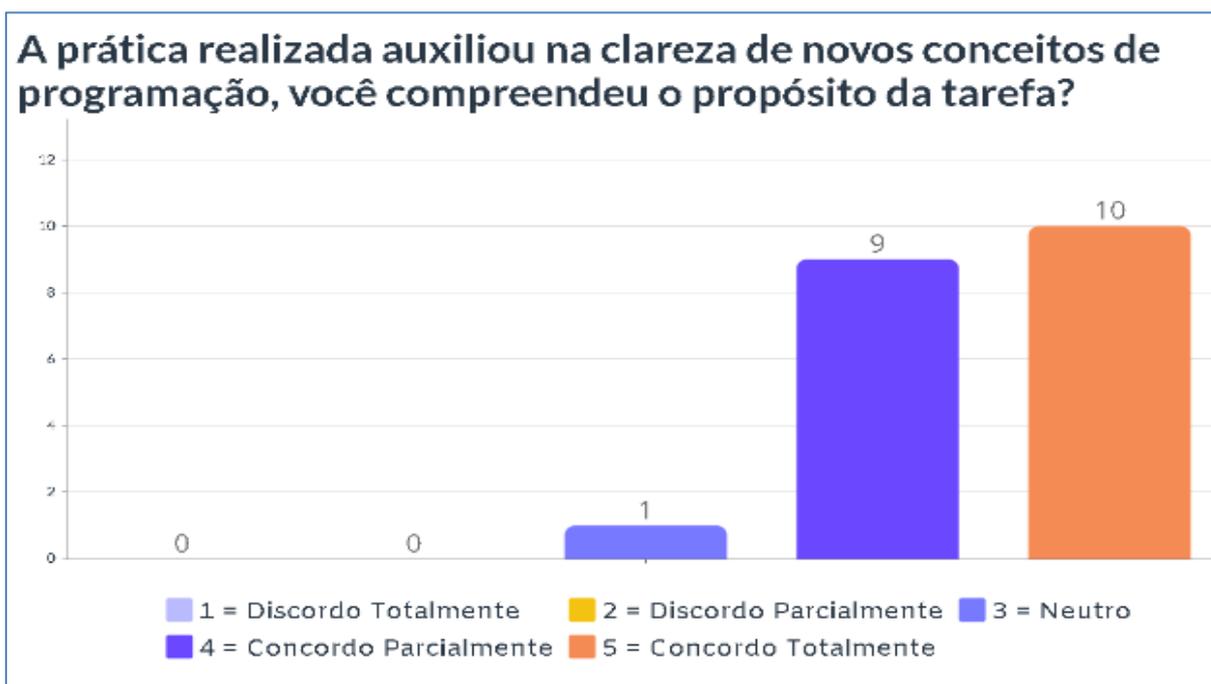


Sobre os componentes eletrônicos, sensores e atuadores, 95% dos participantes conseguiram compreender plenamente o funcionamento. Esse mesmo percentual, 95%, encontrou clareza nos conceitos de programação abordados até o momento, confirmando sua eficácia pedagógica. A familiaridade com tais conceitos é um indicativo promissor, visto que a compreensão destes é fundamental para um progresso consistente no universo do Arduino. A escolha didática dos tópicos e a metodologia aplicada parecem estar alinhadas com as necessidades e capacidades dos aprendizes, facilitando assim a absorção do conteúdo e incentivando a exploração autônoma por parte deles. Esta sintonia entre conteúdo e método pode ser a chave para futuras inovações e descobertas por parte dos alunos neste campo (SILVA, 2020).

A realização da prática foi fundamental para iluminar o entendimento de novos conceitos no vasto universo da programação. A cada etapa concluída, era perceptível como os conhecimentos anteriormente abstratos começavam a ganhar contornos mais claros e

tangíveis. Pela experiência, os alunos comentavam nas rodas de conversa após a prática, “aprendi, mas também compreendi profundamente o propósito da tarefa”, reconhecendo sua importância no desenvolvimento e aprimoramento de habilidades técnicas. Esta abordagem prática mostrou-se uma ferramenta valiosa no processo de aprendizado, consolidando a teoria e permitindo uma aplicação real e significativa dos conceitos estudados. A grande maioria dos participantes, 95% concordaram que “a prática realizada auxiliou na clareza de novos conceitos de programação”, (Figura 13).

Figura 13: Questionário – Gráfico 4.



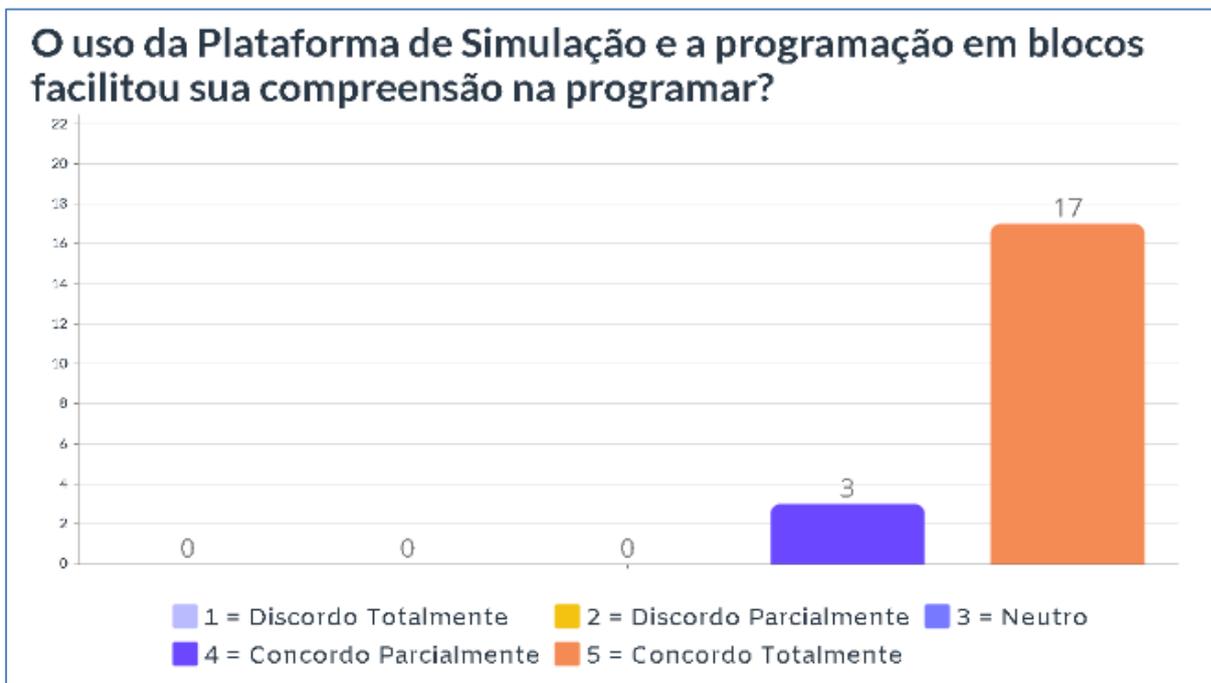
Fonte: Elaborado pelos autores.

Perguntados se “os conceitos de programação em bloco abordados foram adequados para a tarefa proposta”, dos participantes, a maioria expressou satisfação e concordância com a adequação dos conceitos apresentados. Contudo, sempre há espaço para reflexão e melhoria. Um único participante optou por permanecer na zona de neutro, não se inclinndo nem para concordância nem para discordância. Esse feedback, por mais que possa parecer insignificante para alguns, é essencial.

Conforme apontado por Dias et al. (2021), a ferramenta Tinkercad desempenha um papel crucial no refinamento do processo de ensino e aprendizagem dos princípios de programação, especialmente quando há uma combinação de recursos simulados em software e hardware. Em vista dessa observação, não é surpreendente que, ao serem questionados sobre

"O uso da Plataforma de Simulação e a programação em blocos facilitou sua compreensão na programação?", 85% dos participantes afirmaram concordar com essa proposição, (Figura 14).

Figura 14: Questionário – Gráfico 5.



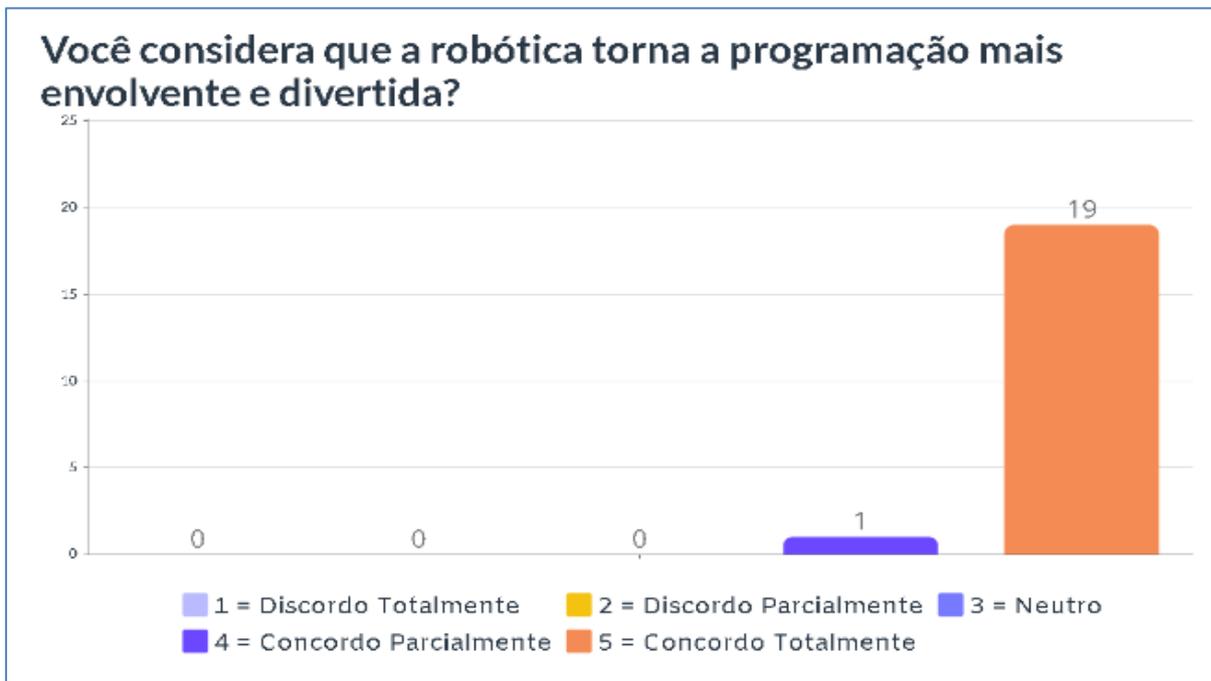
Fonte: Elaborado pelos autores.

Quanto ao paradigma de programar em modo texto ou visual, 90% dos participantes acreditam que é mais simples começar com a linguagem visual. Enquanto isso, 45% se opõem à ideia de aprender por meio da linguagem textual e 35% mantiveram uma postura neutra. Esse feedback dos participantes evidencia uma inclinação significativa para métodos de ensino mais intuitivos e gráficos, especialmente entre aqueles que estão dando seus primeiros passos no mundo da programação. A resposta também destaca a importância de oferecer múltiplos métodos de ensino, reconhecendo as variadas preferências e estilos de aprendizado dos alunos.

Em relação a contribuição da "Robótica no processo de compreensão nos princípios de programação", 95% concordam. Em consonância com a perspectiva de Silva et al. (2021), que defende que "A robótica é um atrativo para alunos de todas as idades escolares, promovendo interação, práticas e raciocínio lógico", (Figura 15). Nas rodas de conversa, o entusiasmo em programar em modo visual é grande, principalmente pelo fato de permitir uma compreensão mais imediata e intuitiva dos conceitos, reduzindo barreiras iniciais

frequentemente associadas à sintaxe e à linguagem textual. Esse formato visual oferece uma representação melhor do fluxo lógico do código.

Figura 15: Questionário – Gráfico 6.



Fonte: Elaborado pelos autores.

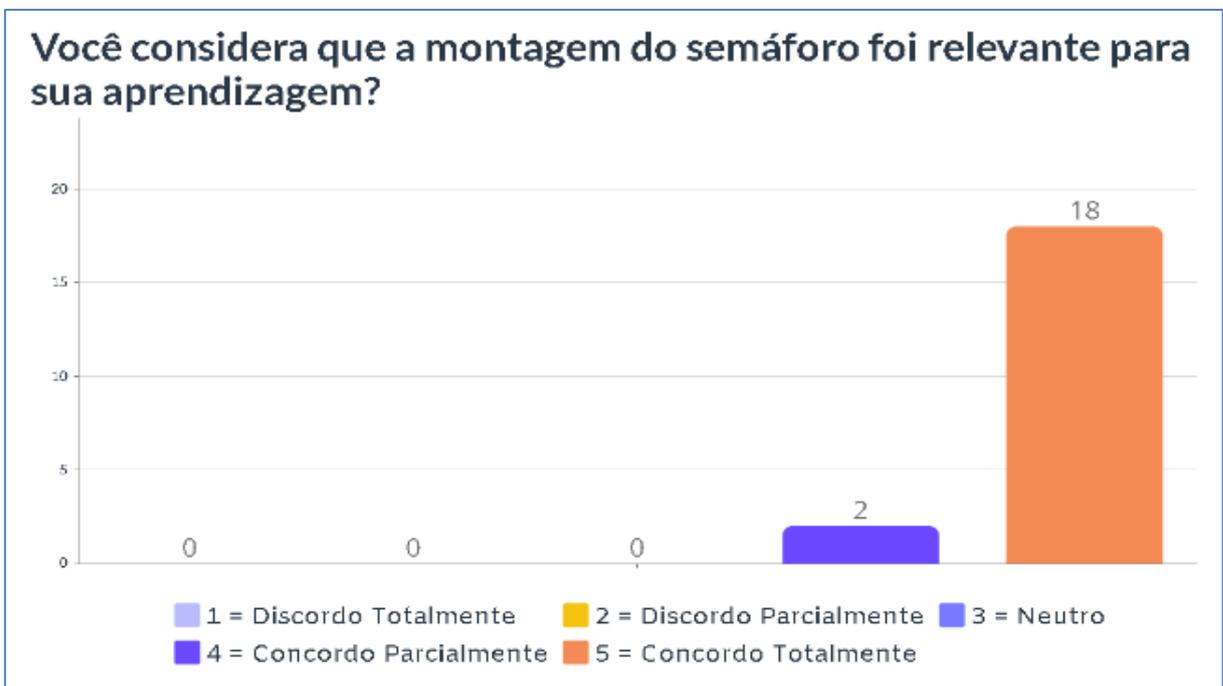
A questão 14 considera a montagem do semáforo como relevante para o processo de ensino aprendizagem porque proporciona uma compreensão prática dos conceitos teóricos, permitindo uma visão direta do funcionamento dos componentes e da lógica por trás do sistema. O desafio de montar o semáforo estimulou o raciocínio crítico e a resolução de problemas, aspectos cruciais para o desenvolvimento do pensamento computacional. Nesta questão 95% dos participantes concordaram (Figura 16).

A montagem do semáforo é fundamental para o processo de aprendizagem por várias razões:

- 1) Aplicação Prática de Teoria: A montagem permite que os alunos coloquem em prática os conceitos teóricos aprendidos em sala de aula.
- 2) Desenvolvimento do Raciocínio Lógico: O processo exige que os alunos pensem sequencialmente e em etapas, o que ajuda a cultivar o raciocínio lógico.
- 3) Resolução de Problemas: Em qualquer projeto de montagem, os alunos inevitavelmente encontrarão desafios ou erros que precisarão ser corrigidos.

- 4) Trabalho em Equipe: Montar um semáforo, especialmente em grupos, requer colaboração. Os alunos aprendem a comunicar suas ideias, aceitar feedback e trabalhar coletivamente para atingir um objetivo comum.
- 5) Estímulo à Criatividade: Mesmo com instruções claras, sempre há espaço para inovação.
- 6) Conexão com o Mundo Real: Semáforos são elementos comuns no ambiente urbano.
- 7) Confiança e Autonomia: Ao concluir a montagem com sucesso, os alunos ganham confiança em suas habilidades e sentem-se mais preparados para enfrentar tarefas semelhantes no futuro.

Figura 16: Questionário – Gráfico 7.



Fonte: Elaborado pelos autores.

Os dados coletados foram organizados, submetidos a uma análise estatística e estão apresentados na (Tabela 1). Aqui, "N" indica o número total de participantes da pesquisa. As colunas de "1" (discordo totalmente) até "5" (concordo totalmente) exibem a quantidade de respostas para cada um dos 5 níveis da escala Likert. Estes foram reclassificados em 3 níveis: os valores 1 e 2 são interpretados como discordância "d", o valor 3 é visto como neutralidade "n" e os valores 4 e 5 são vistos como concordância "c". As colunas d%, n% e c% mostram os percentuais de discordância, neutralidade e concordância, respectivamente.

Tabela 1: Estatística descritiva dos constructos (N = 20).

Item	1	2	3	4	5	d	d%	n	n%	c	c%
P1	1	0	1	3	15	1	5%	1	5%	18	90%
P2	8	4	3	2	3	12	60%	3	15%	5	25%
P3	0	1	3	4	12	1	5%	3	15%	16	80%
P4	5	4	2	5	4	9	45%	2	10%	9	45%
P5	0	0	1	8	11	0	0%	1	5%	19	95%
P6	0	0	1	9	10	0	0%	1	5%	19	95%
P7	0	0	1	3	16	0	0%	1	5%	19	95%
P8	0	0	0	3	17	0	0%	0	0%	20	100%
P9	4	5	7	1	3	9	45%	7	35%	4	20%
P10	0	0	2	5	13	0	0%	2	10%	18	90%
P11	1	0	2	5	12	1	5%	2	10%	17	85%
P12	0	0	1	6	13	0	0%	1	5%	19	95%
P13	0	0	0	1	19	0	0%	0	0%	20	100%
P14	0	0	0	2	18	0	0%	0	0%	20	100%

Legenda: 1 = discordo totalmente; 2 = discordo parcialmente; 3 = nem discordo e nem concordo; 4 = concordo parcialmente; 5 = concordo totalmente; d = discordância; n = neutralidade; c = concordância

Fonte: Elaborado pelos autores,2023.

4.2. Discursão

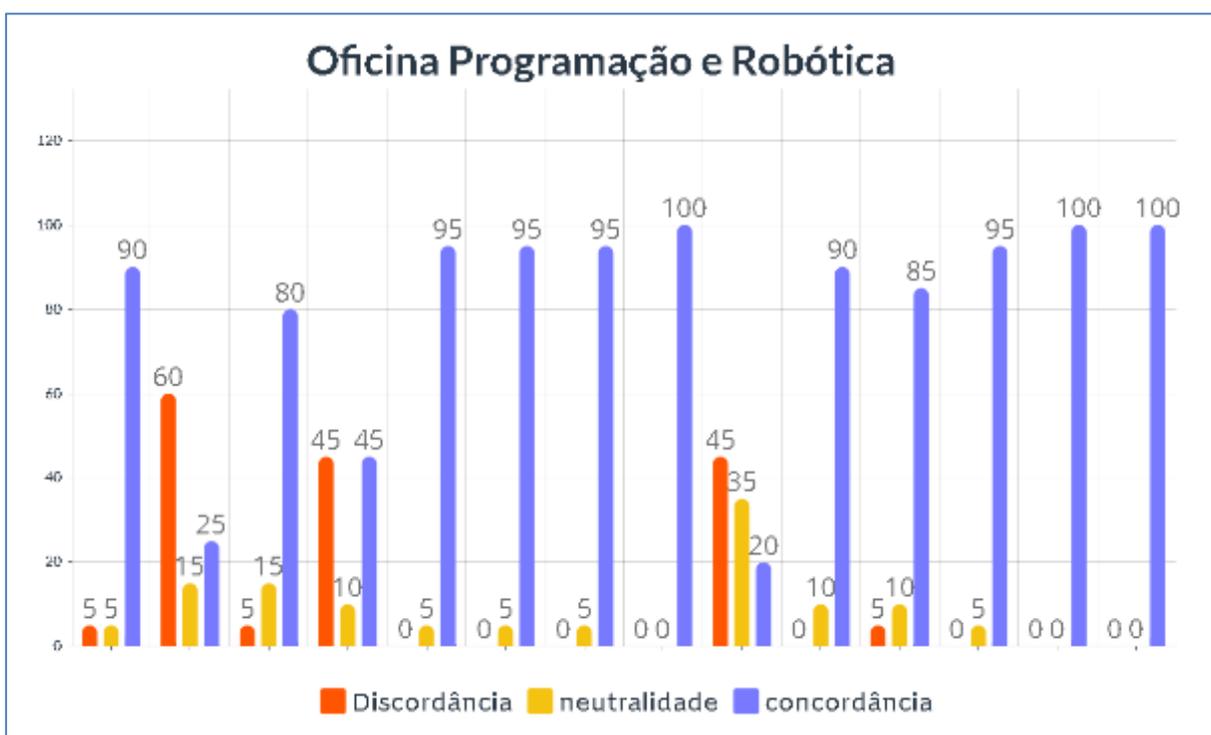
O estudo mostra que o método empregado teve um efeito benéfico nos alunos. A grande maioria dos participantes percebeu que o modelo proposto ofereceu uma abordagem mais envolvente e atrativa para aprender programação. O método adotado engloba o desenvolvimento e avaliação de práticas de maneira sistemática e estruturada, utilizando instrumentos e fundamentos teóricos ligados à Robótica Educacional, Eletrônica, Plataformas de Simulação e à Programação em Blocos. Tal abordagem ainda é pouco explorada, especialmente em estudos relacionados à aplicação da RE no ensino e aprendizado de programação.

Observa-se claramente que a Robótica Educacional desempenha um papel significativo no ensino e aprendizagem da programação, conforme ilustrado na (Figura 17). A

maioria das respostas (cerca de 80%) está distribuída na categoria "c = concordância". Isso significa que a grande maioria dos participantes concordou com alguma afirmação ou posição relacionada ao uso da Robótica. Um grupo menor, representando 9%, demonstrou "neutralidade". Isso indica que esse grupo não expressou uma opinião clara a favor ou contra. A discordância é mínima, já que somam apenas 11%. Isso significa que apenas 11% dos participantes discordaram ou expressaram uma opinião contrária ao uso da Robótica, enquanto a grande maioria concordou ou permaneceu neutra.

Essas informações sugerem que, na amostra em questão, a aceitação e a neutralidade em relação à Robótica são predominantes, com uma minoria relativamente pequena discordando. Esses dados podem ser úteis para entender a opinião ou atitude das pessoas em relação à Robótica em um determinado contexto.

Figura 17: Gráfico Descritivo.



Fonte: Elaborado pelos autores, 2023.

A integração da robótica como ferramenta pedagógica no ensino e aprendizado da programação pode se revelar uma abordagem inovadora e eficaz. Isso ocorre, em parte, porque essa proposta vem ao encontro dos interesses dos estudantes, incentivando-os a se tornarem ativos e responsáveis por sua própria trajetória de aprendizado. A robótica não

apenas engaja, mas também empodera os alunos a dirigirem sua própria aprendizagem, instigando a autonomia e o desenvolvimento contínuo.

A robótica é indiscutivelmente uma disciplina multifacetada, que cruza fronteiras entre áreas como informática e eletrônica. Nesse contexto, ela se apresenta como um recurso valioso para a promoção de atividades interdisciplinares. Tais atividades proporcionam uma aprendizagem que perpassa diversos temas e domínios, permitindo que os estudantes conectem conceitos e habilidades de diferentes áreas do conhecimento. A abordagem interdisciplinar promovida pela robótica estimula uma visão holística e integrada, fundamental para a formação de profissionais adaptáveis e versáteis no futuro.

A implementação de práticas com a Robótica inevitavelmente leva em conta a possibilidade de inconsistências nas operações com os componentes eletrônicos, sensores e atuadores. Isso acontece, pois, estes componentes são sensíveis a variáveis externas à programação, como a alteração na luminosidade que afeta os sensores de luz. Muitas das reações ou respostas negativas podem originar-se dessas incongruências, que não foram previstas ou abordadas durante a atividade. Além disso, é vital considerar o nível de experiência dos alunos. O seu histórico e familiaridade com a matéria influenciam diretamente na sua capacidade de avaliação e percepção crítica em relação aos desafios e resultados apresentados. Portanto, ao avaliar o desempenho e as respostas dos alunos, é crucial levar em consideração tanto os desafios técnicos quanto o nível de expertise de cada indivíduo.

A segunda pergunta da avaliação foi formulada de uma maneira que carregava uma conotação negativa, especificamente, “teve dificuldade em programar o circuito para que ele operasse como um semáforo”. Esta formulação pode ter levado a uma interpretação ambígua por parte dos respondentes. Portanto, a alta taxa de discordância observada para esta questão, paradoxalmente, indica um cenário positivo. Ou seja, a discordância, neste caso, sugere que muitos alunos não encontraram dificuldades e que a programação foi realizada com sucesso. Assim, é essencial interpretar os dados à luz da formulação da questão, entendendo que a "discordância" aqui, na verdade, reflete uma experiência positiva no processo de programação do semáforo.

5. CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS

Na sequência, trazemos a conclusão desta dissertação, ressaltando as valiosas contribuições da pesquisa, as perspectivas para trabalhos futuros e nossas considerações finais. Inicialmente, focamos nas principais conclusões obtidas, seguidas pelas contribuições significativas da pesquisa em relação às questões que nortearam toda a investigação e os progressos decorrentes desta análise. Posteriormente, abordamos os possíveis rumos para trabalhos futuros, enfatizando as limitações do estudo atual e do método adotado. Neste contexto, também se destaca a busca e a necessidade de novos estudos e pesquisas no campo. Por fim, as considerações finais são delineadas, incorporando reflexões de natureza pessoal e uma análise crítica detalhada dos resultados e do impacto da pesquisa no domínio em questão.

Tópicos que abrangem o escopo deste capítulo:

5.1 Conclusões

5.2 Contribuições da Pesquisa

5.3 Trabalhos Futuros e Limitações

5.1. Conclusões

A avaliação derivada dos dados deste estudo aponta, que a estratégia implementada gerou um impacto benéfico no processo de aprendizagem dos alunos, conforme mencionado anteriormente. A grande maioria dos participantes percebeu o modelo como uma ferramenta valiosa para a aprendizagem da programação, pois o tornou mais envolvente. A metodologia empregada englobou a concepção e avaliação de práticas mediante um procedimento estruturado e formal, utilizando instrumentos e fundamentações teóricas ancoradas na Programação e Robótica Educacional. Tal abordagem, ainda pouco explorada na literatura, se mostra promissora em pesquisas que versam sobre a integração da Robótica Educacional no processo educacional da programação.

Ela se alinha efetivamente aos interesses dos estudantes, colocando-os no centro do seu próprio processo de aprendizagem e crescimento. A robótica, por sua natureza, é um campo multidisciplinar, abarcando áreas como Informática e Eletrônica. Assim, torna-se um meio propício para fomentar atividades interdisciplinares que impulsionem um aprendizado integrado em diversas temáticas. As iniciativas que incorporam a robótica proporcionam uma vasta gama de desafios a serem enfrentados pelos alunos, muitos dos quais são imprevistos, surpreendendo até mesmo o educador à frente do projeto. A emergência destes desafios a partir de contextos reais, e o fato de que alguns podem ser insuperáveis, elevam sua importância, diferenciando-os das tradicionais tarefas de sala de aula. Em quase todas as atividades robóticas, os alunos cooperam em grupos, trabalhando coletivamente rumo a um objetivo compartilhado.

O engajamento dos alunos nas atividades de robótica não apenas fortalece habilidades técnicas e cognitivas, mas também cultiva habilidades socioemocionais, como trabalho em equipe, comunicação e resolução de conflitos. A realidade dinâmica das tarefas os estimula a pensar criticamente, a adaptar-se rapidamente a mudanças e a abordar problemas de ângulos variados. Este tipo de aprendizado, que é intrinsecamente prático e imersivo, facilita a consolidação do conhecimento, tornando-o mais duradouro. O papel do educador, neste contexto, evolui para o de um facilitador, guiando os alunos através de suas descobertas, incentivando a autonomia e a tomada de decisão. A combinação de teoria e prática oferecida pela RE potencializa o processo educacional, preparando os alunos não apenas para exames ou para cumprir currículos, mas para os desafios multifacetados do mundo real.

A análise das respostas do questionário revela um panorama bastante elucidativo sobre a percepção dos participantes em relação à robótica como ferramenta pedagógica. Há uma clara inclinação, quase unânime, em direção à valorização da robótica como meio de tornar a programação mais tangível e atraente. Com impressionantes 100% dos respondentes acreditando que a robótica torna o processo de aprendizagem da programação mais envolvente e lúdico, isso reitera a ideia de que a robótica, ao trazer a programação para o mundo físico, permite que os alunos vejam os efeitos práticos de seu código, tornando a experiência mais significativa.

Além disso, a proporção (95%) afirmou que a robótica auxilia na compreensão dos princípios fundamentais da programação. Isso sugere que a robótica não apenas engaja os alunos, mas também facilita a internalização dos conceitos programáticos. A interação direta com dispositivos robóticos pode ajudar a desmistificar a programação, tornando-a menos abstrata e mais compreensível, especialmente para aqueles que estão dando seus primeiros passos nesse domínio.

Por fim, o fato de que 90% dos entrevistados acreditam que iniciar com linguagem visual é mais simples reforça a ideia de que as ferramentas visuais oferecem uma entrada menos intimidante para a programação. Linguagens visuais, muitas vezes, reduzem a barreira da sintaxe rigorosa, permitindo que os iniciantes se concentrem na lógica e no fluxo da programação. Ao oferecer uma introdução mais amigável e intuitiva, é provável que mais alunos se sintam confiantes para explorar e se aprofundar na programação.

Em conclusão, o objetivo primordial desta pesquisa foi atingido, permitindo a aquisição de informações sobre as variáveis que influenciam a adoção da Robótica pelos estudantes. Adicionalmente, observa-se que a pergunta central da investigação foi satisfatoriamente respondida, caminhando na direção que "**A adoção da Robótica Educacional auxilia no processo de aprendizagem da programação**" e que, sob o ponto de vista dos alunos, essa contribuição é moldada por variáveis ancoradas no modelo de aceitação tecnológica.

5.2. Contribuições da Pesquisa

Com a finalidade de dar resposta à pergunta de pesquisa proposta no começo desta dissertação:

- **A Robótica Educacional auxilia no processo de aprendizagem da programação?**

Este estudo focou na avaliação das capacidades da Robótica Educacional como instrumento pedagógico, integrando-a efetivamente em um contexto de ensino de Programação. Estabeleceu-se uma conexão direta entre esses dois domínios. A atividade foi avaliada considerando-se seus objetivos gerais e específicos, bem como as práticas empregadas. Assim, conseguiu-se propor uma solução que fornece respostas tangíveis aos desafios associados.

Entre as principais contribuições desta pesquisa, podemos salientar:

- Criação de um método estruturado que guia a execução e análise de práticas no campo da Programação, fundamentado em uma abordagem inovadora que incorpora estratégias da Robótica Educacional, Eletrônica, Plataformas de Simulação e à Programação em Blocos;
- Uma abordagem que pode ser adotada pela perspectiva de um investigador ou executor. Oferecendo diversas alternativas que podem ser personalizadas e ajustadas a contextos específicos do ensino e aprendizado de Programação, conforme as necessidades do público;
- Análise e descobertas sobre o efeito da Robótica Educacional no ensino de Programação no Ensino Básico, principalmente no Ensino Médio.

Em resumo, os achados advindos dessas experiências indicam que a Robótica Educacional possui um significativo potencial no âmbito do ensino de Programação. Contudo, as restrições na implementação desse método estão ligadas ao uso de artefatos específicos nas atividades práticas.

5.3. Trabalhos Futuros e Limitações

O tamanho da amostra em estudos de pesquisa desempenha um papel vital na validade e precisão dos resultados obtidos. Neste estudo específico, a amostra de dados, composta por apenas 20 participantes, apresentou uma limitação significativa, principalmente quando se considerou a aplicação do modelo TAM (Modelo de Aceitação da Tecnologia). Este modelo, amplamente reconhecido por sua eficácia em avaliar a aceitação de tecnologias, exige uma amostra mais ampla para gerar resultados mais robustos e confiáveis.

Devido a esta restrição amostral, surgiram desafios ao tentar estabelecer padrões claros e conclusões definitivas a partir dos dados coletados. Dessa forma, tornou-se evidente a necessidade de desenvolver um modelo específico e adaptado, focado principalmente na aceitação da Robótica Educacional. Tal modelo, ajustado às peculiaridades e nuances deste campo educacional emergente, poderia oferecer insights mais precisos e aprofundados sobre a percepção e a receptividade dos alunos em relação à Robótica no ambiente educativo.

Em pesquisas futuras relacionadas à Robótica Educacional, é fundamental ampliar o número de participantes. Ao fazê-lo, não só se poderia aplicar o modelo TAM de maneira mais eficaz, mas também enriquecer a análise através da inclusão de diversas perspectivas e experiências, tornando as conclusões mais abrangentes e representativas da população estudantil. No Apêndice H, está disponível um Modelo de Aceitação de Tecnologia que foi proposto para ser aplicado em futuros estudos que envolvam o uso da Robótica educacional como uma ferramenta tecnológica.

Outra limitação, mais evidente ao longo do estudo, ocorreu durante as aulas de montagem de circuitos e sistemas. Invariavelmente, surgiam imprevistos e desafios, muitos dos quais estavam diretamente relacionados a falhas ou incompatibilidades nas conexões entre os componentes, incluindo cabos, sensores e outros elementos eletrônicos. Esses contratempos não apenas atrasavam o progresso das atividades planejadas, mas também podiam desviar a atenção dos alunos do objetivo pedagógico principal. Além disso, essas falhas inesperadas exigiam soluções criativas e, por vezes, uma reestruturação das tarefas propostas, o que podia levar a momentos de frustração tanto para os educadores quanto para os alunos. No entanto, é válido ressaltar que, enquanto tais desafios representavam obstáculos, eles também ofereciam oportunidades de aprendizado real, onde os alunos eram encorajados a diagnosticar problemas, pensar criticamente e colaborar para encontrar soluções práticas.

Durante o período da pesquisa, encontramos diversos trabalhos destacando os benefícios da implementação da robótica para auxiliar no ensino dos fundamentos da programação. Muitos destes estudos abordam as vantagens das plataformas de simulação e das linguagens visuais. Estas plataformas e linguagens facilitam a compreensão dos conceitos por serem mais intuitivas, permitindo que os alunos visualizem e manipulem diretamente os elementos da programação. Além disso, proporcionam uma imersão prática, na qual os estudantes podem ver suas criações em ação, reforçando a aprendizagem através da

experimentação. Isso gera um engajamento maior dos alunos, tornando o processo de aprendizado mais dinâmico e motivador.

Em conclusão, após uma reflexão aprofundada, estou firmemente convencido da relevância de abordagens que integram a Robótica Educacional e a Programação, particularmente no contexto do ensino básico. O projeto "Robótica Educacional Livre no Processo de Ensino e Aprendizagem de Programação no Ensino Básico" não só sublinha essa intersecção, mas também demonstra o potencial educacional desta combinação. Esta iniciativa oferece uma perspectiva única e valiosa para o campo da Informática na Educação, um domínio ainda em crescimento e com espaço para investigações mais detalhadas. Espero que este trabalho sirva como uma faísca, incentivando outros pesquisadores a aprofundar e expandir este tema, ampliando a compreensão de como a Robótica Educacional pode revolucionar o ensino da programação e, por extensão, a educação como um todo.

6. BIBLIOGRAFIA

ABBURI, R., PRAVEENA, M., PRIYAKANTH, R. **TinkerCad – a Web Based Application for Virtual Labs to Help Learners Think, Create and Make.** Journal of Engineering Education Transformations. 34. 535-541. 2021.

AFONSO, Nuno Miguel Machado. **Da tarefa ao projeto: uma visão construtivista do ensino da programação orientada a objetos.** Dissertação (Mestrado em Ensino de Informática) Universidade do Minho. Braga, Portugal. 2013. 136p.

AHMAD, Ashfaq; IDREES, Muhammad; BUTT, Muhammad Arif ; DANISH, Hafiz.. **BBVPL: A Block-Based Visual Programming Language Built on Google's Blockly.** International Journal of Advanced Trends in Computer Science and Engineering. DOI:10.30534/ijatcse/2021/1441032021. ISSN 2278-3091. Volume 10, No.3, May - 2021.

ALLEN, K. Using drones to save lives in Malawi. **BBC News**, mar 2016. Disponível em: <<http://www.bbc.com/news/world-africa-35810153>>. Acesso em: 15 mar 2023.

ANTONACHI, André Bueno. **O USO DA ROBÓTICA COMO ESTRATÉGIA DIDÁTICA PARA O ENSINO DA FÍSICA NO ENSINO MÉDIO.** Dissertação (Mestrado em Educação). Universidade Municipal de São Caetano do Sul. São Caetano do Sul, 2020. 176 p.

ARENDET, R. **Construtivismo ou construcionismo? Contribuições deste debate para a Psicologia Social.** Estudos de Psicologia 2003, 8(1), 5-13. 2003.

AUGUSTO, C. A. et al. **Pesquisa Qualitativa: rigor metodológico no tratamento da teoria dos custos de transação em artigos apresentados nos congressos da Sober (2007-2011).** Revista de Economia e Sociologia Rural, Brasília, v. 51, n. 4, p. 745- 764, out-dez 2013.

BANZI, M. How **Arduino is open-sourcing imagination.** TEDGlobal, 2012. Disponível em: <https://www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination/transcript?language=pt-br>. Acesso em: 31 mar 2023.

BARBERO, A.; DEMO, B. ; VASCHETTO, F. **A contribution to the discussion on informatics and robotics in secondary schools.** In: INTERNATIONAL CONFERENCE ON ROBOTICS IN EDUCATION, 2., 2011, Vienna. Proceedings ... Vienna: Austrian Society for Innovative Computer Sciences, 2011.

BENITTI, F. B. V. (2012). **Exploring the educational potential of robotics in schools: A systematic review.** *Computers & Education*, 58(3), 978-988.

BERGMANN, Jonathan; SAMS, Aaron. **Sala de aula invertida: uma metodologia ativa de aprendizagem.** Rio de Janeiro: LTC, 2018.

BRASIL, CNE/CEB. Res. N° 1/2022. **Normas sobre Computação na Educação Básica – Complemento à Base Nacional Comum Curricular (BNCC).** Disponível em: <<http://portal.mec.gov.br/docman/outubro-2022-pdf/241671-rceb001-22/file>>. Acesso em: 25 abr. 2023.

BRASIL, CNE/CEB. Res. N° 2/2022. **Normas sobre Computação na Educação Básica – Complemento à Base Nacional Comum Curricular (BNCC).** Disponível em: <http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=235511-pceb002-22&category_slug=fevereiro-2022-pdf&Itemid=30192>. Acesso em: 25 abr. 2023

BRASIL. Ministério da Educação. Conselho Nacional de Educação. Normas sobre Computação na Educação Básica – Complemento à BNCC. Processo N° 23001.001050/2019-18. 2019.

BRASIL. Ministério da Educação. Base Nacional Comum Curricular. Brasília, 2018.

Brito, J. V. da C. S. de & Ramos, A. S. M. **Limitações dos Modelos de Aceitação da Tecnologia: um Ensaio sob uma Perspectiva Crítica.** *Gestão.Org*, 17(EE), 210–220. 2019. <https://doi.org/10.21714/1679-18272019v17esp.p210-220>.

BUCKINGHAM, D. **Aprendizagem e Cultura Digital.** *Revista Pátio*, Ano XI, No. 44, jan. 2008. Disponível em: <https://www.academia.edu/2748122/Aprendizagem_e_cultura_digital?auto=download>. Acesso em: 16/02/2020.

CAMPOS, F. R. **Paulo Freire e Seymour Papert: educação, tecnologias e análise do discurso.** Curitiba: Editora CRV, 2013. *E-book*.

CAMPOS, F. R. **Robótica educacional no Brasil: questões em aberto, desafios e perspectivas futuras.** *Revista Ibero-Americana de Estudos em Educação*, Vol 12, Iss 4, Pp 2108-2121 (2017). 4, 2108, 2017. ISSN: 2446-8606.

CAMPOS, F. R. **A robótica para uso educacional**. Editora Senac. São Paulo - São Paulo – 2019. 208 p.

CAMPOS, F. R.; LIBARDONI, Gláucio Carlos. Investigação em robótica na educação brasileira: o que dizem as dissertações e teses. *In: Robótica educacional - experiências inovadoras na educação brasileira*. Porto Alegre: Penso, 2020. p. 21–45. *E-book*.

CARDOSO, Rogério; ANTONELLO, Sérgio. **Interdisciplinaridade, programação visual e robótica educacional**: relato de experiência sobre o ensino inicial de programação. I Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação. DOI:10.5753/cbie.wcbie.2015.1255.

CHEMIN, Beatris Francisca. **Manual da Univates para trabalhos acadêmicos: planejamento, elaboração e apresentação**. 3a ed. Lajeado: Ed. da Univates, 2015. E-book.

CHIOU, A. **Teaching Technology Using Educational Robotics**. UniServe Science Scholarly Inquiry Symposium Proceedings. 2004.

CRESWELL, J. W. **Projeto de pesquisa: métodos qualitativo, quantitativo e misto**. 2. ed. Porto Alegre: Artmed, 2007.

CRUZ, Marcia Elena Jochims Kniphoff da. **Introduzindo a robótica na escola**. Santa Cruz do Sul: EDUNISC, 2013.

DARGAINS, André Rachman. **Estudo exploratório sobre o uso da robótica educacional no ensino de programação introdutória**. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, Brasil, 2015.

DAVIS, F. D. **Perceived usefulness, perceived ease of use, and user acceptance of information technology**. *MIS Quarterly*, v. 3, p. 319-340, 1989.

DIESEL, A.; BALDEZ, A. L. S.; MARTINS, S. N. **Os princípios das metodologias ativas de ensino: uma abordagem teórica**. *Revista Thema*, vol. 14, nº 1, Pág. 268- 288, 2017.

DONNELLY, R., FITZMAURICE, M. **Collaborative project-based learning and problem-based learning in higher education**: a consideration of tutor and student roles in learner-focused strategies. In O’Neill, G., Moore, S., McMullin, B. (eds): *Emerging Issues in the Practice of University Learning and Teaching*. Dublin: AISHE. 2005.

ELIA, M.F. **A História da Informática na Educação no Brasil**: uma narrativa em construção. Porto Alegre: SBC, 2021. (Série Informática na Educação CEIE-SBC, v.4) Disponível em: <<https://ieducacao.ceie-br.org/historiainformaticaeducacao>>. Acesso em: 14 mai. 2023.

ERYILMAZ, S.; DENIZ, G. **Effect of Tinkercad on Students' Computational Thinking Skills and Perceptions**: a Case of Ankara Province. *The Turkish Online Journal of Educational Technology*, 20 (1). 25-38. 2021.

FAGIN, B.; MERKLE, L. **Measuring the Effectiveness of Robots in Teaching Computer Science**. *ACM SIGCSE Bulletin* 35.1, 2003.

FIGUEIREDO, A.; AFONSO, A. **Managing Learning in Virtual Settings: the Role of Context**. Information Science Publishing. Hershey, PA, USA, 2006. 333 p.

FLANNERY, L.; SILVERMAN, B; KAZAKOFF, E; Bers, M; BONTÁ, P; RESNICK, M. **Designing scratchjr**: Support for early childhood learning through computer programming. In *Proceedings of the 12th International Conference on Interaction Design and Children*, pages 1–10, 2013.

GENERAL PUBLIC LICENSE. Disponível em: <<http://www.gnu.org/>>. Acesso em: 15 jan. 2023.

GUIA DE TECNOLOGIAS EDUCACIONAIS 2009 / organização Cláudio Fernando André. – Brasília : Ministério da Educação, Secretaria de Educação Básica, 2009.

GUIA DE TECNOLOGIAS EDUCACIONAIS 2013 / organização Cláudio Fernando André. – Brasília : Ministério da Educação, Secretaria de Educação Básica, 2013.

GOH, H.; ARIS, B. (2007) **Using robotics in education**: lessons learned and learning experiences. *Smart Teaching & Learning: Re-engineering ID, Utilization and Innovation of Technology*, 2. ISSN 983-42733-2-3.

GOMES, A., AREIAS, C., HENRIQUES, J., MENDES, A. J. **Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte**. *Revista Portuguesa de Pedagogia*, v. 42, n. 2. 2008.

GOMES, A. L. R.; ALVES, M. P. Robótica, ecologia dos saberes e CTSA. **Revista Brasileira de Educação Ambiental (RevBEA)**, [S. l.], v. 15, n. 7, p. 410–422, 2020. DOI: 10.34024/revbea.2020.v15.10203. Disponível em: <https://periodicos.unifesp.br/index.php/revbea/article/view/10203>. Acesso em: 21 mar. 2023.

GOMES, Raquel Silva. **Aplicação do modelo de aceitação da tecnologia (TAM) para analisar os fatores que afetam o uso do Google Classroom entre estudantes do ensino médio**. 2022. 37 f. Monografia (Especialização em Informática na Educação) - Instituto Federal do Espírito Santo, Vitória, 2022.

GROVER, S., PEA, R. **Computational Thinking in K–12: A Review of the State of the Field**. Educational Researcher. Journal Article. 38–43. 2013. <https://doi.org/10.3102/0013189X12463051>.

HARGREAVES, A. **Os professores em tempos de mudança. O Trabalho e a Cultura dos Professores na Idade Pós-Moderna**. Lisboa: Mc Graw-Hill, 1998.

HEDLER, H. C., FERNEDA, E., DUARTE, B. S., PRADO, H. A., & Gutierrez, C. E. C. **APLICAÇÃO DO MODELO DE ACEITAÇÃO DE TECNOLOGIA À COMPUTAÇÃO EM NUVEM**. *Perspectivas em Gestão & Conhecimento*, 6 (2), 188–207. 2016.

Hill, M. M. e Hill, A. (2002). *Investigação por questionário*. Segunda edição revista e corrigida. Edições Sílabo Ltd., Lisboa.

KAMINSKI, M. R.; BOSCARIOLI, C. **Uso do Ambiente Code.org para Ensino de Programação no Ensino Fundamental I - Uma Experiência no Desafio Hora do Código**. *Revista ENCITEC*, v. 9, n. 1, p. 63-76, 2019.

KELLEHER, C.; PAUSCH, R. **Lowering the Barriers to Programming: a survey of programming environments and languages for novice programmers**. *ACM Computing Surveys* 37, 2 , 2005, 83–137.

MALHOTRA, N. K. **Pesquisa de marketing**. 3. ed. Porto Alegre: Bookman, 2001.

MAJOR, L.; KYRIACOU, T.; BRERENTON, O.P. **Systematic Literature Review: Teaching Novices Programming Using Robots**. *Proceedings of EASE*, 2011.

MALIUK, Karina Disconsi. **Robótica educacional como cenário investigativo nas aulas de matemática. 2009.** Dissertação (Mestrado) - Ensino de matemática, Instituto de Matemática, Universidade Federal do Rio Grande do Sul. Porto Alegre. E-book. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/17426>>. Acesso em: 10/02/2020.

MAUCH, E. **Using technological innovation to improve the problem solving skills of middle school students.** The Clearing House, Vol. 74, No. 4, 2001. pp. 211-213.

MILNE, L. R. LADNER, Richard E. **Position:** Accessible Block-Based Programming: Why and How. IEEE Blocks and Beyond Workshop (B&B), Memphis, TN, USA, 2019, pp. 19-22, doi: 10.1109/BB48857.2019.8941230.

MCROBERTS, Michael. **Arduino básico;** [tradução Rafael Zanolli]. São Paulo: Novatec Editora, 2011.

MEC. **Guia de tecnologias educacionais da educação integral e integrada e da articulação da escola com seu território 2013.** Brasília, DF, 2013. 55 p.

MÉLO, F. É. N. ; CUNHA, R. R. M. ; SCOLARO, D. R. ; CAMPOS, J. L. **Do Scratch ao Arduino:** uma proposta para o ensino introdutório de programação para cursos superiores de tecnologia, In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, 39., 2011, Blumenau. Anais. Blumenau: ABENGE/FURB, 2011.

MILNE, L. R. LADNER, Richard E. **Position:** Accessible Block-Based Programming: Why and How. IEEE Blocks and Beyond Workshop (B&B), Memphis, TN, USA, 2019, pp. 19-22, doi: 10.1109/BB48857.2019.8941230.

MINISTÉRIO DA EDUCAÇÃO (MEC). **Base Nacional Comum Curricular.** Disponível em: <<http://basenacionalcomum.mec.gov.br/abase>>. Acesso em: 11/01/2023.

MOHAPATRA, B. N., MOHAPATRA, R.K., JOSHI, J. , ZAGADE, S. **Easy performance based learning of Arduino and sensors through Tinkercad.** International Journal of Open Information Technologies, 8 (10). 73-76. 2020.

NUNES, D. Ciência da computação na educação básica. *Jornal da Ciência – Sociedade Brasileira de Computação.* (SBC), 2011.

OLIVEIRA, Cláudio Luís Vieira; ZANETTI, Humberto Augusto Piovesana. **Arduino Descomplicado**: como elaborar projetos de eletrônica. São Paulo: Érica, 1ª edição, 2015.

OLIVEIRA, K. L. R. de; SILVA, M. A. de F. da; OLIVEIRA, M. G. de; SCARPATI, R.; BATTESTIN, V. **Formação Online de Professores em Robótica Educacional com Práticas no Simulador Tinkercad**. Revista Brasileira de Ensino de Ciências e Matemática, [S. l.], v. 5, n. especial, 2022. DOI: 10.5335/rbecm.v5iespecial.12850.

OREY, M. **Emerging Perspectives on Learning, Teaching and Technology**. 2010.

Disponível em:

https://textbookequity.org/Textbooks/Orey_Emergin_Perspectives_Learning.pdf. Acessado em 05 de maio de 2023.

PAPERT, S. **Mindstorms: Children, Computers, and Powerful Ideas**. Basic Books (AZ); 2nd Revised ed. edição. 1993

PAPERT, S. **Situating constructionism**. In S. Papert and Is. Harel (Eds.), Constructionism (pp. 1-11). Norwood, NJ: Ablex. 1991.

PAPERT, S. **The children's machine**: Rethinking school in the age of the computer. Basic books. 1993.

PASTERNAK, E. **Visual programming pedagogies and integrating current visual programming language features**. Dissertation (Master's Degree) – Robotics Institute, Carnegie Mellon University, Pittsburgh, 2009. Disponível em: <http://www.ri.cmu.edu/pub_files/2009/8/Thesis-1.pdf>. Acesso em: maio. 2023.

PAIVA, Deise de Lacerda; ANDRADE, Jéssica Zacarias de. **A identificação das competências digitais na Base Nacional Comum Curricular para o uso das tecnologias da informação e comunicação na educação básica**. In: CIET – CONGRESSO INTERNACIONAL DE EDUCAÇÃO E TECNOLOGIA: EnPED - ENCONTRO DE PESQUISADORES EM EDUCAÇÃO A DISTÂNCIA, [S.l.], maio 2018. ISSN 2316-8722. E-book. Disponível em: <<http://cietenped.ufscar.br/submissao/index.php/2018/article/view/381>>. Acesso em: 11/02/2020.

PERALTA, Deise Aparecida (Org.). **Robótica e Processos Formativos: da epistemologia aos kits**. Porto Alegre, RS: Editora Fi, 2019.

PIAGET, J. **O nascimento da inteligência na criança**. Rio de Janeiro: Guanabara, 1987.

PRESNSKY, M., **Digital natives, digital immigrants**, On th Horizon, MCB University Press, USA, 2001.

PRODANOV, C. C.; FREITAS, E. C. *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico*. 2. ed. Novo Hamburgo: Editora Feevale, 2013. 277 p. ISBN 9788577171583.

PROJECT MANAGEMENT INSTITUTE. **Um Guia Do Conhecimento Em Gerenciamento De Projetos - Guia PMBOK**. Pennsylvania, EUA, 2017.

RAABE, André Luís Alice. **Pensamento Computacional na Educação**: Para todos, por todos!. Revista Computação Brasil, SBC, p. 54 - 63, 01 jul. 2017.

RESNICK, M. Forewords; Mitchel Resnick; Mcmanus, S. **Scratch programming in easy steps**: covers Scratch 2.0 and Scratch 1.4. In Easy Steps Limited, 2013. 216 p.

REEVES, T. **How do you know they are learning?**: the importance of alignment in higher education. Int. J. Learning Technology, Vol 2, No. 4, 2006.

RUSK, N. et al. New Pathways into Robotics: Strategies for Broadening Participation. **Journal of Science Education and Technology**, v. 17, n. 1, p. 59–69, fev 2008. Disponível em: <<http://link.springer.com/article/10.1007/s10956-007-9082-2>>. Acesso em: 11 jan 2023.

SANTOS, M. F. **A robótica educacional e suas relações com o ludismo: por uma aprendizagem colaborativa**. Dissertação de Mestrado, Universidade Federal de Goiás, Brasil, 2010.

SBC. SBC realiza reunião para discutir a nova versão da Base Nacional Comum Curricular. Sociedade Brasileira de Computação, 2016a. Disponível em: <<http://www.sbc.org.br/noticias/1693-sbc-realiza-reuniao-para-discutir-a-novaversao-da-base-nacional-comum-curricular>>. Acesso em: 16 jan 2023.

SBC. SBC realiza reunião com o Ministro da Educação. Sociedade Brasileira de Computação, 2016b. Disponível em: <<http://www.sbc.org.br/noticias/1868-sbcrealiza-reuniao-com-o-ministro-da-educacao>>. Acesso em: 16 jan 2023.

SETIYANI, L.; EFFENDY, F.; SLAMET, A. A. **Using Technology Acceptance Model 3 (TAM 3) at Selected Private Technical High School: Google Drive Storage in E-Learning**. Utamax : Journal of Ultimate Research and Trends in Education, v. 3, n. 2, p. 80-89, 28 Jul. 2021.

SILVA, Wilson Rogério Soares, FREITAS, Natalle do Socorro da Costa, ALMEIDA, Klévia Letícia. **Robótica em Tempos de Pandemia: Uma Abordagem Usando Programação em Blocos**. DOI 10.37885/210203196. 2021.196-205.

SILVA, Rodrigo Barbosa; BLIKSTEIN, Paulo. **Robótica Educacional: Experiências Inovadoras na Educação Brasileira**. Porto Alegre: Penso, 2020.

SILVEIRA, J. de A. **Construcionismo e inovação pedagógica: uma visão crítica das concepções de Papert sobre o uso da tecnologia computacional na aprendizagem da criança**. THEMIS: Revista da Esmec, Fortaleza, v. 10, n. 1, p. 119–138, 2012. Disponível em: <http://revistathemis.tjce.jus.br/index.php/THEMIS/article/view/87/85>. Acesso em: 19 jul. 2020.

SILVEIRA JUNIOR, C. R., COELHO, J., BARRA, A. . **Construtivismo e Robótica Educacional: A Construção de Conceitos Matemáticos**. ENCICLOPEDIA BIOSFERA, 11(22).2015. Disponível em <https://conhecer.org.br/ojs/index.php/biosfera/article/view/1403>. Acesso em 15/01/2023.

SILVEIRA, Sérgio Amadeu da. **Software livre: a luta pela liberdade do conhecimento**. São Paulo : Editora Fundação Perseu Abramo, 2004. 82 p.

STAGER, G. **A Constructionist Approach to Teaching with Robotics**. 9th IFIP World Conference on Computers in Education. 2009. Disponível em: <http://www.stager.org/articles/stagerconstructionism2010.pdf>. Acesso em 11/01/2023.

TCHÁPEK, K. **A Fábrica de Robôs**. Tradução de Vera Machac. São Paulo: Hedra, 2012. 148 p.

TORI. **Educação sem distância: as tecnologias interativas na redução de distâncias em ensino e aprendizagem**. São Paulo: Ed. Senac, 2010.

VALENTE, J. A. **Diferentes usos do computador na educação**. In: Computadores e conhecimento: repensando a educação. 1. ed. Campinas, NIED-Unicamp, 1993.

VENKATESH, Viswanath; DAVIS, Fred D. **A theoretical extension of the technology acceptance model**: Four longitudinal field studies. *Management science*, v. 46, n. 2, p. 186-204, 2000.

VILAR, M. A. da Silva. **Modelo de Aceitação da Tecnologia adaptado às compras online**. Dissertação de Mestrado, Universidade Fernando Pessoa, Porto, Portugal, 2013.

VYGOTSKY, L. **A Formação Social da Mente**. São Paulo: Martins Fontes, 2007.

VYGOTSKY, L. S. **Mind in Society**: The development of higher psychological processes. Harvard university press. 1980.

WARREN, John-David. **Arduino para robótica**. São Paulo: Blucher, 2019.

WING, Jeannette M. **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: 14 mar. 2023.

YIN, R. K. **Estudo de Caso: Planejamento e Métodos**. 2. ed. Porto Alegre: Bookman, 2001. 205 p.

ZANETTI, H., OLIVEIRA, C. **Práticas de ensino de Programação de Computadores com Robótica Pedagógica e aplicação de Pensamento Computacional**. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 4(1), 1236. doi:<https://doi.org/10.5753/cbie.wcbie.2015.1236>

ZILLI, S. R. **A robótica educacional no ensino fundamental**: Perspectivas e prática. Dissertação de Mestrado em Engenharia, Universidade Federal de Santa Catarina, Centro Tecnológico, Florianópolis-SC. 2004.



RELIMAX
ROBÓTICA LIVRE



COLUN

TODOS OS DIREITOS RESERVADOS

Curso

PROGRAMAÇÃO e ROBÓTICA

Introdução a Robótica e Eletrônica



NELIO ALVES GUILHON

2023

1. O QUE É ROBÓTICA?

A robótica é uma atividade do mundo da tecnologia, eletrônica, mecânica e computação. A robótica lida com sistemas criados por máquinas e partes mecânicas automáticas, controladas manualmente ou de modo automático controlado. Grande parte dos robôs exercem um trabalho obrigatório e repetitivo.



A palavra robô foi utilizada pela primeira vez por Karel Capek na peça de teatro *Rossum's universal robots* (*robôs universais de Rossum*), escrita em 1920 e encenada em 1921, tendo sua primeira edição em inglês publicada em 1923.

O termo robô vem do vocábulo checo e resulta da combinação das palavras *robot*, que significa servidão, trabalho forçado, ou escravidão e *robotnik*, que significa servo. Era usado especialmente para designar os chamados trabalhadores emprestados, que viveram no Império Austro-Húngaro até 1848. Na peça de Capek, os robôs eram humanos artificiais orgânicos que tratavam de construir mais robôs. A propósito, a peça

introduziu o conceito de fabricação em série executada por robôs, conceito esse que se tornou amplamente utilizado na economia e na filosofia.

Na cultura atual, a palavra robô, quase sempre utilizada para se referir a humanoides mecânicos, partindo dela a criação de outros termos correspondentes, como o androide (podendo se referir tanto a humanos artificiais orgânicos quanto a humanoides mecânicos), um grande exemplo disso são os filmes da franquia *Blade Runner* (1982 e 2017), e o *cyborg* (um organismo cibernético ou homem biônico, uma criatura que combina partes orgânicas e mecânicas), bastante propagado na série de filmes *O exterminador do futuro* (1984, 1991, 2003, 2009 e 2015).



2. INTRODUÇÃO À ELETRÔNICA

A construção de projetos com o Arduino envolve o conhecimento básico de eletrônica, pois isso permitirá a identificação dos componentes que serão utilizados e o entendimento de seu funcionamento. Sendo assim, preparamos este tópico com conceitos sobre eletrônica, visando um melhor aproveitamento do curso por vocês alunos.

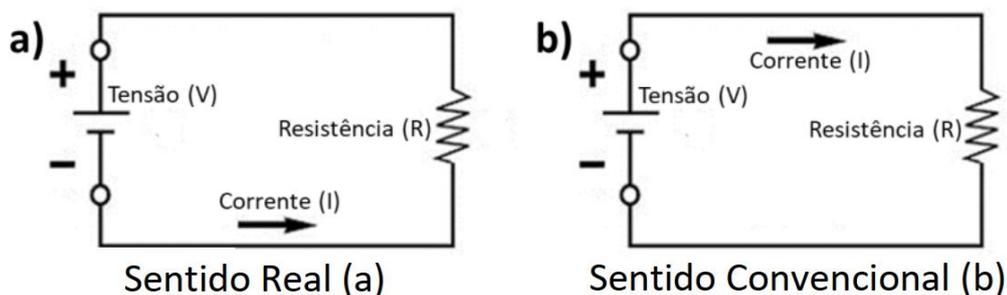
A eletrônica está fundamentada sobre os conceitos de tensão, corrente e resistência. Segundo Silva (2013) podemos entender tensão como a diferença de energia estabelecida entre dois pontos que motiva a movimentação de cargas elétricas, gerando assim a corrente elétrica. Observe a Figura abaixo que, como analogia, podemos pensar na água armazenada em dois recipientes conectados por um cano. A água irá fluir do recipiente com maior quantidade de água para o menor.



Em eletrônica o princípio é o mesmo, por exemplo, os polos positivos e negativos de uma pilha indicam o sentido na qual a corrente elétrica irá fluir. Desta forma, podemos definir que a corrente elétrica é a movimentação ordenada de cargas elétricas num condutor.

Na figura abaixo, observamos que a corrente elétrica poderá circular em dois sentidos.

- a) sentido real, que é resultante do movimento de cargas negativas
- b) sentido convencional – resultante do movimento de cargas positivas.



A movimentação das cargas elétricas através do condutor pode encontrar elementos que oferecem certa resistência a sua passagem. Na figura acima, por exemplo, imagine que a resistência seja uma lâmpada. A resistência a passagem da corrente elétrica faz com que a lâmpada conceba calor no seu filamento e fique incandescente. É esse mesmo efeito que permite que a água de um chuveiro seja aquecida ao passar pela resistência.

Para seu melhor entendimento, listamos os três conceitos abaixo:

1. **Corrente elétrica** é o movimento ordenado de partículas portadoras de carga elétrica — **os elétrons**. Esse fluxo ocorre em um condutor quando houver uma diferença de

potencial entre suas extremidades, essa diferença é chamada tensão. A unidade padrão para medida de intensidade de corrente é o ampère (A).

2. **Resistência elétrica** é a capacidade de um corpo qualquer se opor a passagem de uma corrente elétrica mesmo quando existe uma diferença de potencial aplicada. Segundo o Sistema Internacional de Unidades (SI), é medida em ohms (Ω).
3. A **tensão elétrica** é uma diferença entre o potencial elétrico de dois pontos, ou traduzindo de uma forma bem simples e comparativa, seria a força necessária para movimentar os elétrons e criar assim uma corrente elétrica. Segundo o SI, a tensão é medida em volts (V).

Sabendo um pouco sobre estes três conceitos, já é possível fazer muitas coisas na área da eletrônica. Reuniremos então esses três conceitos para aprender como descobrir o valor de cada um.

Os cálculos mostrados abaixo são feitos a partir da 1ª lei de Ohm, que com os valores da corrente, da tensão e da resistência, consegue realizar os cálculos necessários para a conservação dos equipamentos que serão utilizados, evitando que sejam danificados.

A primeira fórmula serve para calcular a tensão (V), partindo da multiplicação entre a resistência (R) e a corrente (I). Para encontrar os valores da Resistência (R) e da corrente (I), a fórmula torna-se semelhante, mudando apenas a operação.

$$V = R * I$$

$$I = V / R$$

$$R = V / I$$

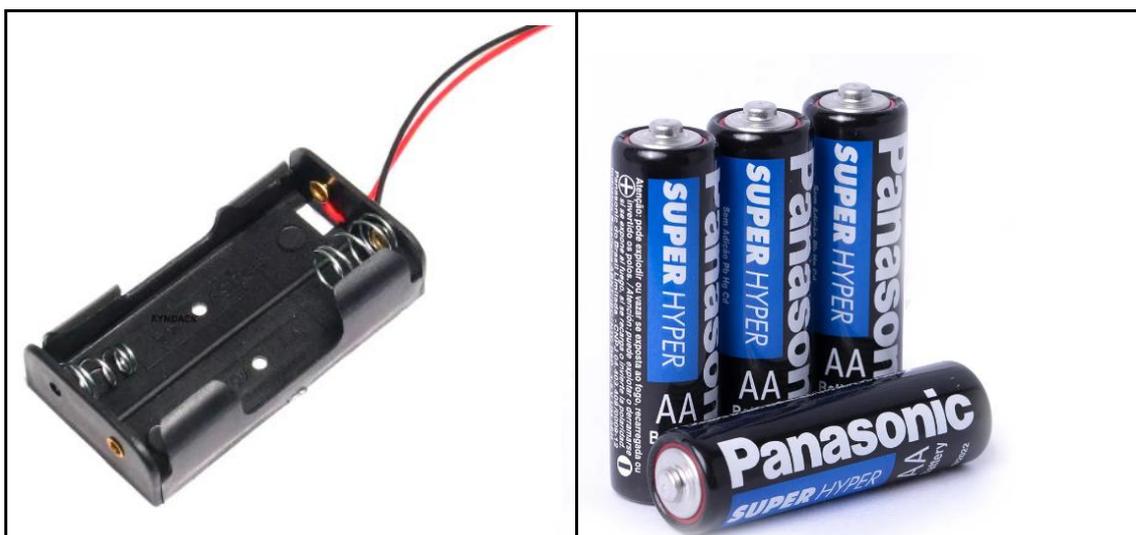
Para calcular o valor da corrente (I), por exemplo, utilizaremos o valor da tensão (V), dividido pelo valor da resistência (R), e para calcular o valor da Resistência (R) dividimos a tensão (V) pela corrente (I).

FONTE DE ALIMENTAÇÃO

Existem vários tipos de fontes de alimentação. Pilha, Baterias e carregadores de dispositivos, como por exemplo o carregador de aparelho celular.

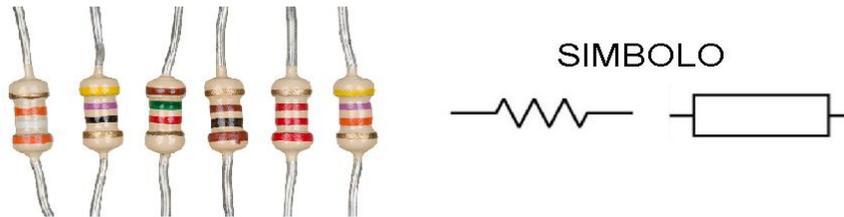
BATERIAS E PILHAS

Uma pilha ou bateria é um dispositivo que transforma energia química em energia elétrica. Os termos são usados indistintamente, no entanto, Pilha é constituída por dois únicos elétrodos, Bateria é composta por um conjunto de pilhas agrupadas em série ou paralelo.



RESISTOR

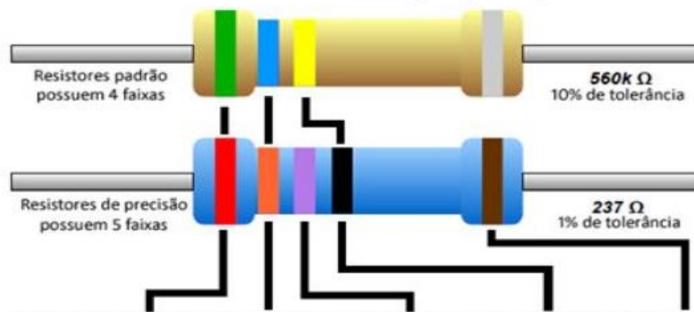
É um componente formado por carbono e outros elementos resistentes usados para limitar a corrente elétrica em um circuito. Por seu tamanho muito reduzido, é inviável imprimir nos resistores as suas respectivas resistências. Optou-se então pelo uso do código de cores.



SIMBOLO

Código de Cores

A extremidade com mais faixas deve apontar para a esquerda



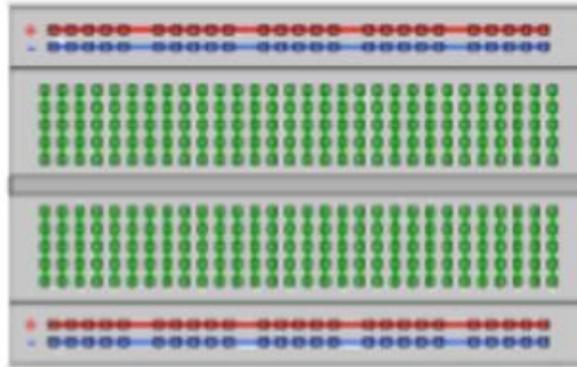
Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Tolerância
Preto	0	0	0	x 1 Ω	
Marrom	1	1	1	x 10 Ω	+/- 1%
Vermelho	2	2	2	x 100 Ω	+/- 2%
Laranja	3	3	3	x 1K Ω	
Amarelo	4	4	4	x 10K Ω	
Verde	5	5	5	x 100K Ω	+/- 5%
Azul	6	6	6	x 1M Ω	+/- 25%
Violeta	7	7	7	x 10M Ω	+/- .1%
Cinza	8	8	8		+/- .05%
Branco	9	9	9		
Dourado				x .1 Ω	+/- 5%
Prateado				x .01 Ω	+/- 10%

Para melhor entendermos a figura acima, é importante sabermos que cada cor possui um valor diferente. O resistor tem diferentes faixas coloridas. Vamos ver o que isso significa? Cada uma dessas faixas tem uma função:

- A primeira, indica o primeiro algarismo do valor da resistência.
- A segunda, indica o segundo algarismo da resistência.
- A terceira faixa nos mostra o terceiro algarismo da resistência, porém nem todos os resistores utilizam esta terceira faixa, como por exemplo os resistores que vão ser utilizados nas simulações do THIKERCAD.
- A quarta faixa mostra o multiplicador, e a quinta faixa mostra o valor da tolerância.

PLACA DE TESTES

Outro componente que utilizamos bastante em projetos é a placa de testes que também é conhecida como **protoboard**, ela é ótima para a montagem dos circuitos, particularmente durante a fase de desenvolvimento de um projeto, pelo fato de ser de fácil a inserção dos componentes, uma vez que, com seu uso, não é necessário o uso de soldagem.



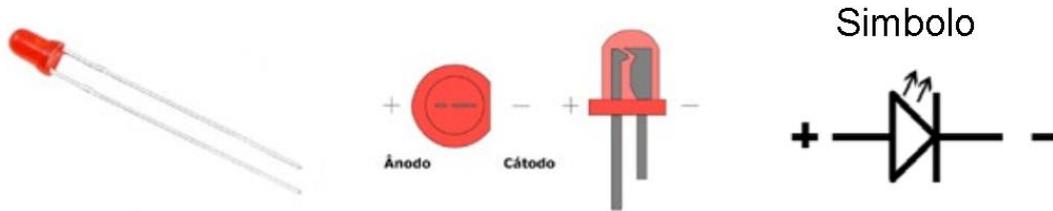
No layout típico da protoboard, figura abaixo, existem faixas de barramentos que são usadas para o fornecimento de tensão ao circuito, elas estão conectadas verticalmente nas duas faixas das laterais. uma utilizada para o condutor negativo ou terra (faixa azul), e a outra para o positivo (faixa vermelha). Na parte central da placa, os barramentos são conectados horizontalmente (faixas verdes), onde serão inseridos os componentes eletrônicos.

Além das faixas de barramento, também existem as faixas de contatos que ficam posicionadas horizontalmente na placa, nas quais são instalados os componentes eletrônicos.

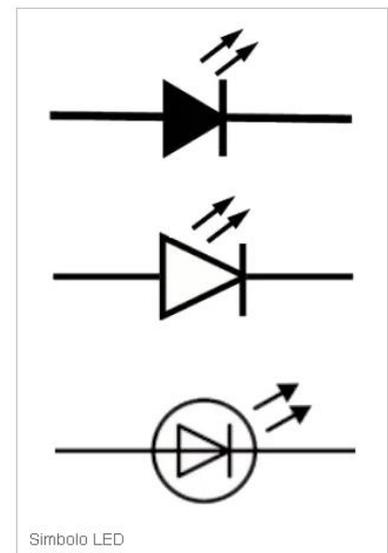
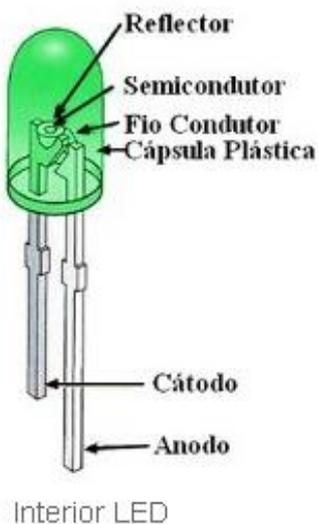
As cinco colunas de contatos do lado esquerdo são separadas das cinco colunas do lado direito por meio de um intervalo que fica localizado no centro das colunas, esse entalhe como é comumente chamado, dá um corte nas colunas, fazendo assim com que não tenha uma ligação direta entre as duas colunas.

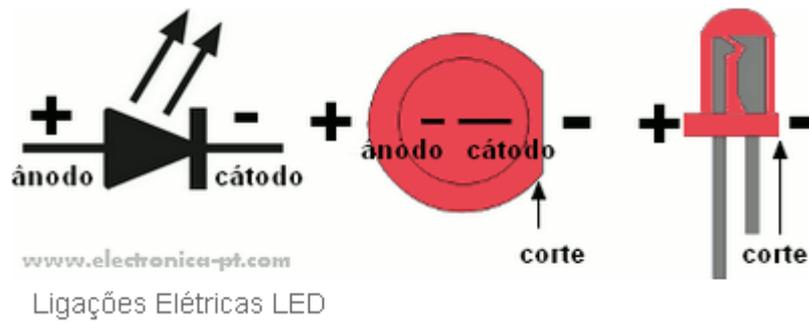
Para realizar tais ligações é necessário utilizar os fios **Jumpers**, os Jumpers são utilizados para fazer a maioria das conexões entre os componentes.

DIODO EMISSOR DE LUZ - LED

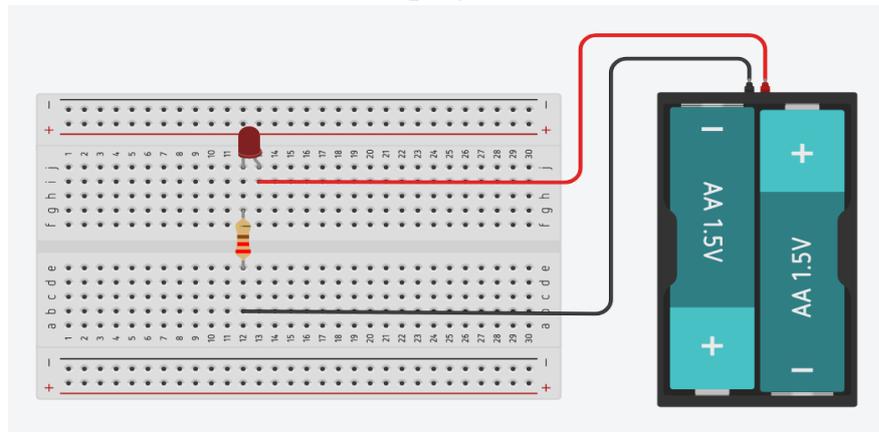


O LED (Light Emitting Diode) é um diodo que emite luz quando energizado. Os LED's apresentam muitas vantagens sobre as fontes de luz incandescentes como um consumo menor de energia, maior tempo de vida, menor tamanho, grande durabilidade e confiabilidade. uma polaridade, uma ordem de conexão. Ao conectá-lo invertido não funcionará corretamente. Revise os desenhos para verificar a correspondência do negativo e do positivo. São especialmente utilizados em produtos de microeletrônica como sinalizador de avisos. Também é muito utilizado em painéis, cortinas e pistas de led. Podem ser encontrados em tamanho maior, como em alguns modelos de semáforos ou displays. Um LED deve ser ligado de forma correta, o circuito de ligação deve ter o + para o ânodo e - para o cátodo. O cátodo é a ponta mais curta e deve ter um corte no lado da cápsula do LED. Se olharmos para o interior do led o ânodo é o elétrodo maior (embora não seja uma forma standard de identificação pode ser utilizada)





Vamos acessar o THIKERCAD e abrir o projeto abaixo:



Vamos abrir o link abaixo:

AULA 1

<https://www.tinkercad.com/things/29QOF63o91u?sharecode=0k9mNkJ4YfXevnbaDoz2bMER4sa7FNwuMCuTkVRwJEY>

Características de alguns leds

Tipo	Cor	I_F max.	V_F typ.	V_F max.	V_R max.	Intensidade Luminosa	Angulo visualização	Comprimento onda
Standard	Vermelho	30mA	1.7V	2.1V	5V	5mcd @ 10mA	60°	660nm
Standard	brilhante vermelho	30mA	2.0V	2.5V	5V	80mcd @ 10mA	60°	625nm
Standard	Amarelo	30mA	2.1V	2.5V	5V	32mcd @ 10mA	60°	590nm
Standard	Verde	25mA	2.2V	2.5V	5V	32mcd @ 10mA	60°	565nm
Alta intensidade	Azul	30mA	4.5V	5.5V	5V	60mcd @ 20mA	50°	430nm
Super brilho	Vermelho	30mA	1.85V	2.5V	5V	500mcd @ 20mA	60°	660nm
Baixa corrente	Vermelho	30mA	1.7V	2.0V	5V	5mcd @ 2mA	60°	625nm

- **I_F max.:** Corrente máxima com o led ligado corretamente.
- **V_F typ.:** Voltagem típica, V_L - É aproximadamente 2V, excepto para os leds azuis que é 4V.
- **V_F max.:** Tensão máxima.
- **V_R max.:** Tensão máxima inversa - Este valor pode ser ignorado se o led estiver ligado corretamente.
- **Intensidade luminosa:** Brilho do led com a corrente normal de funcionamento, mcd = millicandela.
- **Ângulo de projecção de luz:** Standard LEDs têm um ângulo de 60°.

- **Comprimento de onda:** O pico de comprimento de onda visual determina a cor da luz emitida pelo LED. (nm = nanometre.)

CÁLCULO DE RESISTOR (RESISTÊNCIA) LIMITADOR DE UM LED.

O led e a resistência estão em série, a tensão no led somada com a tensão sobre o resistor será igual a tensão da fonte de alimentação (**V_f**). Para calcular precisamos saber o valor da tensão sobre a resistência.

V_f = Vfonte = tensão da fonte em volt(**V**);

R = resistência em ohms (**Ω**);

i_{led} = corrente sobre o led em amperes (**A**);

V_{led} = tensão do led em volts(**V**);

Cálculo da resistência:

Um led vermelho (FLV 110), tem tensão de 1,7 V, (tensão da fonte é de acordo com a simulação) e uma corrente de 15mA ou 0,015 A.

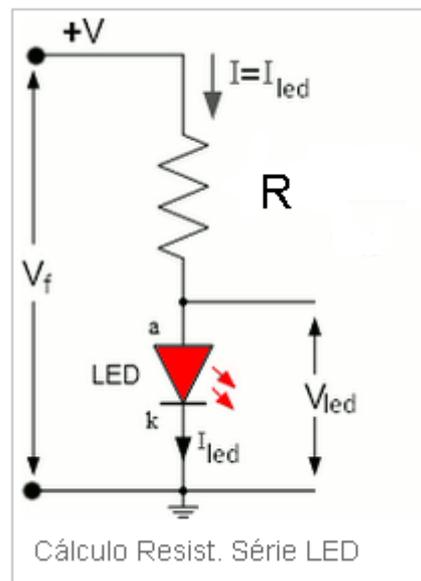
Se a fonte for de 6V, então teremos:

$$V_{res} = V_{fonte} - V_{led}$$

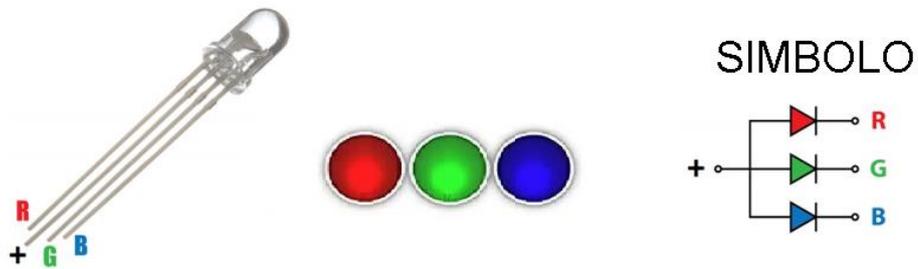
$$V_{res} = 6 - 1,7 = 4,3$$

$$R = V_{res}/i_{led}$$

Logo, **R** é igual a **290Ω**
(valor comercial).

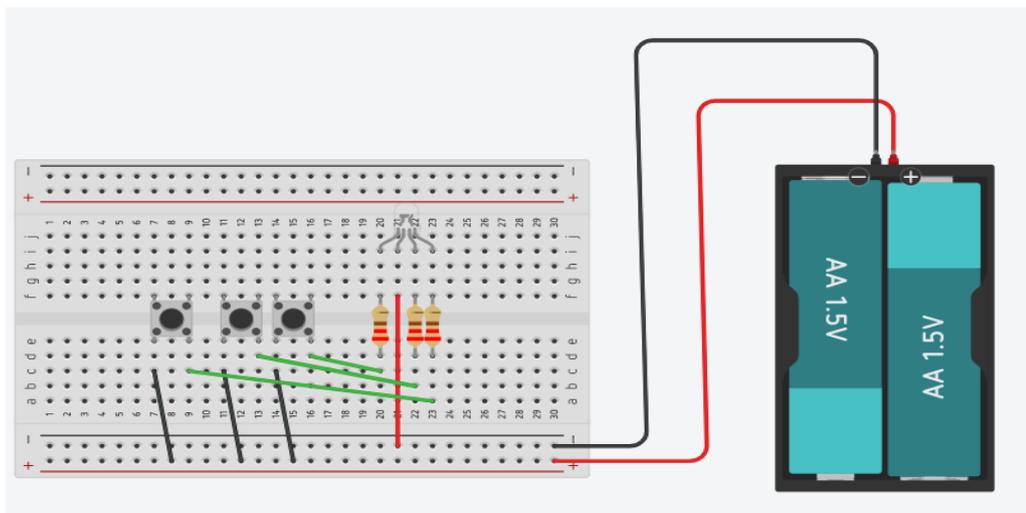


LED RGB



Um LED RGB é um LED que incorpora em um mesmo encapsulamento três LED's, um vermelho (Red), um verde (Green) e outro azul (Blue). Desta forma é possível formar milhares de cores ajustando de maneira individual cada cor. Os três LED's estão unidos por um negativo ou cátodo.

Vamos acessar o THIKERCAD e abrir o projeto abaixo:

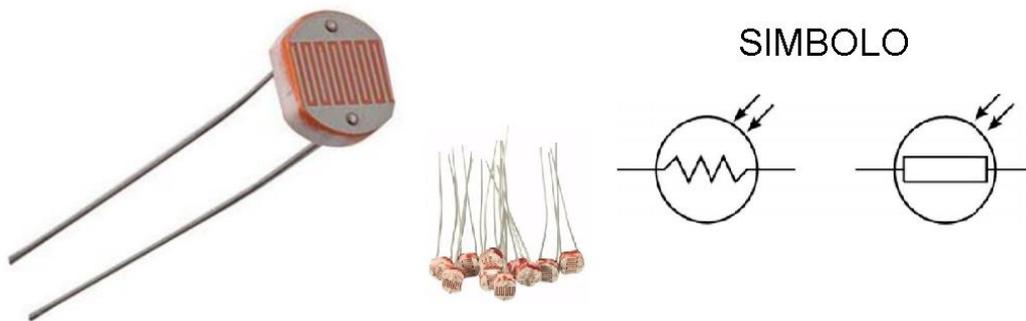


Vamos abrir o link abaixo:

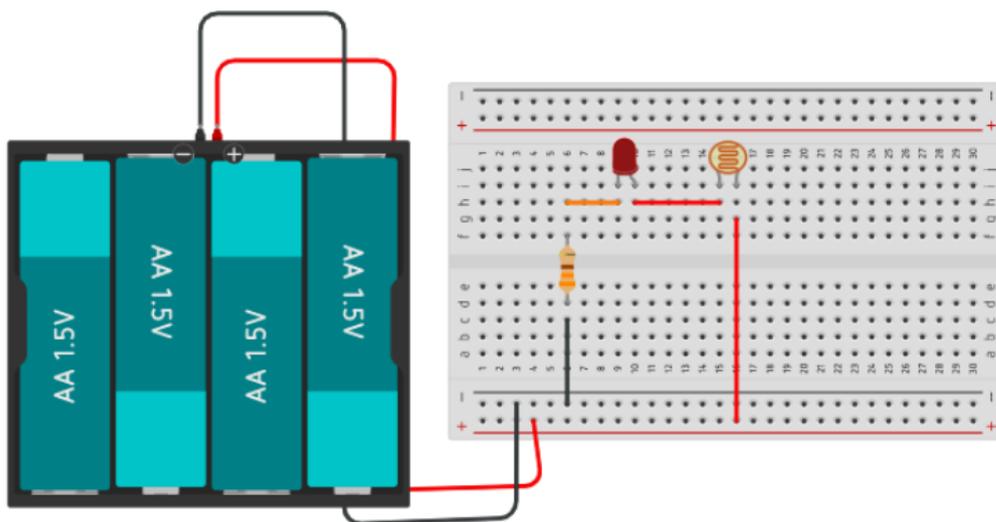
AULA 2

<https://www.tinkercad.com/things/9qEHsymR1Zz?sharecode=mDnt1iMTTv5pOdSVluuTey-kVwham8xYOupwmGXkVo>

RESISTOR SENSÍVEL A LUZ - LDR



O LDR, sigla em inglês de Light-Dependent Resistor, que significa resistor dependente de luz, nada mais é do que o próprio nome diz. Quando maior a luz incidente neste componente, menor será sua resistência.



Vamos abrir o link abaixo:

AULA 3

<https://www.tinkercad.com/things/de3nbVGpjS6>

PARA QUE SERVE O ARDUINO?

O Arduino é sem dúvidas uma placa muito eficiente e poderosa. Pode ser utilizado para fazer qualquer coisa, a imaginação é o limite (e claro, as leis da física também!). É possível utilizá-lo para controlar, monitorar, automatizar etc. Por exemplo, existem projetos de monitoramento da qualidade do ar, medição da temperatura de um líquido, sistemas de irrigação, robôs, impressoras 3D, dentre vários outros.

HARDWARE

O hardware (as placas) do projeto possui diferentes modelos, alguns deles são menores que um cartão de crédito. Sim, tudo isso falado anteriormente cabe na palma de sua mão! Para o desenvolvimento deste material utilizaremos o modelo Uno, que é mais comumente utilizado em projetos básicos. Existe uma placa voltada para cada projeto, algumas são menores e mais compactas e outras são maiores, permitindo controlar um maior número de dispositivos eletrônicos.

O Arduino é um computador como qualquer outro, possuindo:

- Microprocessador (responsável pelos cálculos e tomada de decisão)
- Memória ram (utilizada para guardar dados e instruções, volátil)
- Memória flash (utilizada para guardar o software, não volátil)
- Temporizadores (timers)
- Contadores
- Clock, e etc.

Ou seja, é um computador, porém em menor escala. Possui inclusive menos memória e menor poder de processamento. O Arduino Uno, por exemplo, possui as seguintes especificações:

- Microcontrolador: ATmega328
- Portas Digitais: 14
- Portas Analógicas: 6
- Memória Flash: 32KB (0,5KB usado no bootloader?)
- SRAM: 2KB
- EEPROM: 1KB
- Velocidade do Clock: 16MHz



Figura 1.1 — Arduino Uno R3

Bootloader: Para dispensar o uso de um gravador externo, a gravação da Flash é feita por um software pré-gravado, o Bootloader. O Bootloader é o primeiro software executado pelo microcontrolador após um Reset (Boot) e carrega na Flash um software que recebe pela serial (loader).

ALIMENTAÇÃO DO ARDUINO

O circuito interno do Arduino é alimentado com uma tensão contínua de 5V, isto quando é conectado a uma porta Usb do computador. Esta conexão fornece a alimentação e também a comunicação de dados. Caso seja necessário é possível utilizar uma fonte de alimentação externa, que forneça uma saída dentre 7.5V e 12V contínua com um plug P4, ou pode ser ligada diretamente na placa utilizando os pinos Vin e Gnd.

PORTAS DIGITAIS

Utilizamos as portas digitais quando precisamos trabalhar com valores bem definidos de tensão. Apesar de nem sempre ser verdade, geralmente trabalhamos com valores digitais binários, ou seja, projetamos sistemas que utilizam apenas dois valores bem definidos de tensão. Existem sistemas ternários, quaternários, mas focaremos no binário, já que é esse o utilizado pelo Arduino.

Como o sistema é binário, temos que ter apenas duas tensões. São elas: 0V e 5V. Dessa forma, as portas digitais do Arduino podem trabalhar apenas com essas duas tensões { e o software que desenvolvermos poderá requisitar ao microcontrolador do Arduino que:

- Coloque uma determinada porta em 0V;
- Coloque uma determinada porta em 5V;
- Leia o valor de uma determinada porta (terá 0V ou 5V como resposta).

O Arduino UNO possui 14 portas e apesar de ser possível, não é recomendável utilizar as portas 0 e 1 pois elas estão diretamente ligadas ao sistema de comunicação do Arduino (pinos RX e TX { recepção e transmissão) e, por isso, seu uso pode conflitar com o upload do software. Caso queira utilizá-las, certifique-se de desconectar quaisquer circuitos conectados a ela no momento de carregar o programa para a placa. Utilizamos as funções **digitalRead** e **digitalWrite** para ler e escrever, respectivamente, nas portas digitais.

PORTAS DIGITAIS PWM

As portas PWM (Pulse Width Modulation, do inglês Modulação por Largura de Pulso) se diferenciam das portas digitais binárias pois podem trabalhar não apenas com as tensões 0V e 5V, mas com uma escala que vai de 0 a 255 entre essas tensões, onde o '0' quer dizer 0V e '255' quer dizer 5V. Ou seja, as portas PWM permitem obter resultados analógicos com meios digitais e são capazes de controlar a potência de saída de um sinal.

Pode se controlar, por exemplo, a potência em um Led, permitindo aumentar ou diminuir sua intensidade luminosa.

PORTAS ANALÓGICAS

As portas Analógicas são utilizadas para entrada de dados. Diferentes das portas digitais, permitem não apenas ler os valores 0V e 5V, mas qualquer valor entre eles dentro de uma escala de 0 a 1023, onde o 0 representa 0V e o 1023 representa 5V, ou seja. Isto porque os conversores ADC (do Inglês Analog Digital Converter) são de 10 bit (10 bit = $2^{10} = 1024$ valores) e fornecem uma precisão de 0.005V ou 5mV. Essas portas são utilizadas, por exemplo, para ler os valores de um sensor.

A função **analogRead** é usado para ler as portas analógicas, que nos devolve um valor entre 0 e 1023, onde 0 corresponde a uma leitura de 0V na porta analógica e 1023 corresponde a 5V (para valores intermediários).

COMUNICAÇÃO SERIAL

A Comunicação Serial permite a comunicação entre o Arduino e o computador, possibilitando o envio de mensagens entre ambos. As mensagens podem ser enviadas através do teclado, ou até mesmo de algum programa instalado no computador, permitindo monitorar e controlar uma aplicação do Arduino.

As principais funções para manipular a comunicação serial são:

- `Serial.begin` (velocidade)

Inicia a interface serial. O parâmetro velocidade é a taxa de transferência, por padrão utiliza-se 9600.

- `Serial.print ("Mensagem")`

Exibe uma mensagem no monitor serial. Esta mensagem pode ser, por exemplo, a leitura de um sensor.

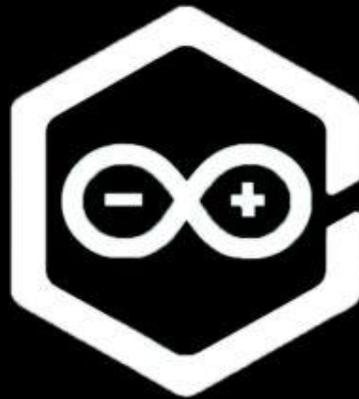
- `Serial.available()`

Retorna a quantidade de bytes disponíveis para leitura na porta serial.

- `Serial.read()`

Lê os dados na porta serial. Por exemplo, uma mensagem digitada a partir do teclado.

APENDICE B



RELIMAX
ROBÓTICA LIVRE



COLUN

TODOS OS DIREITOS RESERVADOS

Curso

PROGRAMAÇÃO e ROBÓTICA

Introdução a Robótica e Eletrônica

2

NELIO ALVES GUILHON

2023

PROGRAMANDO O ARDUINO - SOFTWARE

O Software é utilizado basicamente para escrever o código do programa, salvá-lo, compilá-lo, e realizar a gravação do código compilado no Arduino (memória flash) através da porta Usb do computador. A IDE do Arduino será utilizada para realizar estes passos. Este ambiente de desenvolvimento é baseado no Framework Wiring e na linguagem de programação C/C++.

Uma vez gravado o programa no Arduino, o computador não é mais necessário. A partir do momento em que se utiliza uma fonte de alimentação externa, o Arduino se torna uma placa totalmente independente. Mas antes de tudo é preciso obter os arquivos de instalação e drivers, que vêm juntos no mesmo pacote, que pode ser obtido no site oficial do Arduino.

O download deve ser selecionado de acordo com o sistema operacional utilizado, sendo ele compatível com Windows, Linux e Mac OS.

Página de Download: <https://www.arduino.cc/en/software>

Após baixar e instalar execute o Arduino.exe, localizado no desktop do seu computador. Em seguida conecte o seu Arduino ao computador através de uma porta Usb. Ao conectá-lo, um Led de power (pwr) acenderá, isto significa que a placa está energizada. Agora já é possível instalar os Drivers e usar o Arduino.

VISÃO GERAL DA IDE

Pronto! Após a instalação será possível abrir a IDE do Arduino, que tem a seguinte aparência:

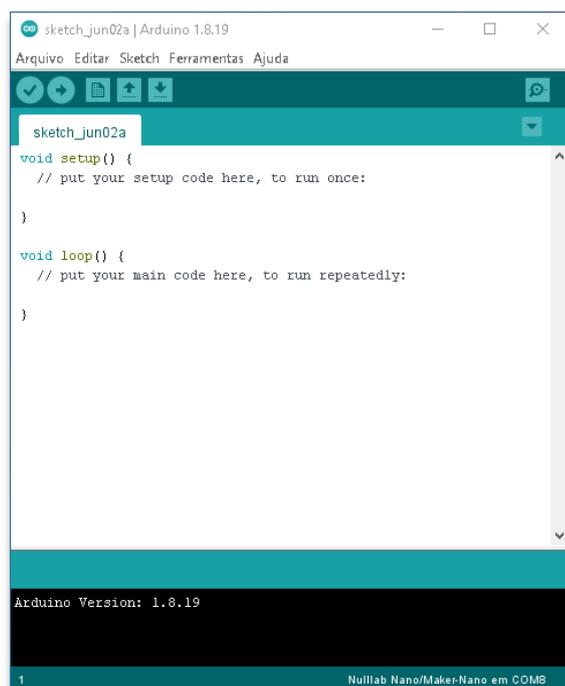


Figura 2 — IDE do Arduino versão 1.8.19 executada no Windows 10.

A IDE do Arduino é muito simples e objetiva, tornando todo o processo de desenvolvimento e gravação bastante intuitivos. Além do espaço em branco destinado ao desenvolvimento do programa, existem 6 botões na parte superior: Verifica, Carregar, Novo, Abrir, Salvar e Serial Monitor.



Figura 3 — Botões IDE do Arduino.

Eles são utilizados, respectivamente para verificar se existem erros no código, enviar (gravar) o programa no Arduino, criar um novo código, abrir um código existente e um monitor de dados da porta serial.

Um código desenvolvido para Arduino é chamado de Sketch, traduzindo do inglês de forma literal seria algo como “esboço” ou “rascunho”. Isso nos dá uma ideia de que nunca terminamos um código, sempre haverá melhorias e novas funcionalidades. O Sketch possui a extensão “.pde”.

A partir de agora sempre que for necessário gravar um novo programa no Arduino, basta conectá-lo na porta Usb, Selecionar a placa utilizada em **Ferramentas > Placa** e em seguida selecionar a Porta Serial (COM) associada a placa.

Em seguida, após abrir o programa desejado, basta clicar em “Carregar”. Após executar estes passos, a IDE deverá exibir uma mensagem no final “Carregado”.

FUNDAMENTOS DA PROGRAMAÇÃO NO ARDUINO

ALGORITMO

Pode se dizer que Algoritmo é uma sequência de passos que devem ser seguidos para atingir um objetivo bem definido. Exemplo, uma receita de bolo.

ESTRUTURA DO FLUXOGRAMA PARA O ARDUINO



Esse fluxograma representa genericamente qualquer programa de um Arduino.

O primeiro bloco, a elipse de cor cinza escuro serve apenas para indicar o nome do projeto, programa, exercício ou atividade que será desenvolvida.

O segundo bloco, na cor vermelha, representa uma etapa de pré-configuração, quando precisamos incluir bibliotecas (programas previamente desenvolvidos), declarar variáveis e constantes globais que farão parte do programa e outras funções específicas.

Conforme veremos em seguida, esse bloco não é necessário em projetos muito simples.

O terceiro bloco, na cor amarela, é obrigatório em todo programa de Arduino. Nessa etapa, o Arduino é configurado (setup) para operar de acordo com as necessidades do projeto.

O quarto bloco, na cor verde, é também obrigatório, pois é nessa etapa que efetivamente se encontra o programa principal, formando um loop.

O quinto bloco, na cor cinza claro, só faz parte do projeto caso haja necessidade de se criar funções específicas ou sub-rotinas que se repetem constantemente no processamento do programa.

VARIÁVEIS E CONSTANTES

Um dado é constante quando não sofre nenhuma alteração ao decorrer do programa. Ou seja, do início ao fim da execução do programa o seu valor permanece o mesmo, inalterado. A declaração de constantes pode ser feita de duas maneiras:

- Usando a palavra reservada “const”.

Exemplo:

```
constint x = 10;
```

- Usando a palavra reservada “define”. Exemplo:

```
#define X = 10
```

Existem algumas constantes pré-definidas, cujos nomes não podem ser utilizados para a declaração de variáveis. Estas são chamadas palavras reservadas. Por exemplo:

- **True** - indica um valor lógico verdadeiro
- **false** - indica um valor lógico falso
- **HIGH** — indica que uma porta está ativada, ou seja, está em 5V.
- **LOW** indica que uma porta está desativada, ou seja, está em OV.
- **INPUT** — indica que uma porta será utilizada como entrada de dados.
- **OUTPUT** — indica que uma porta será utilizada como saída de dados.

Variáveis são posições da memória (lugares) cujo principal objetivo é armazenar dados.

- As variáveis são acessadas através de um identificador único.
- O conteúdo de uma variável pode variar ao longo do tempo durante a execução de um programa.
- Uma variável só pode armazenar um valor a cada vez.
- Um identificador para uma variável é formado por um ou mais caracteres, obedecendo a seguinte regra: o primeiro caractere deve ser uma letra.

Importante: Um identificador de uma variável ou constante não pode conter caracteres especiais ou palavras reservadas da linguagem.

TIPOS DE VARIÁVEIS

- void - Indica um tipo indefinido. Usado geralmente para informar que uma determinada função não retorna nenhum valor.
- boolean - Os valores possíveis são true (1) e false (0). Ocupa um byte de memória.
- char - Pode ser uma letra ou um número. A faixa de valores válidos é de -128 a 127. Ocupa um byte de memória.
- unsigned char - O mesmo que o char, porém a faixa de valores válidos é de 0 a 255.
- Byte - A faixa de valores é de 0 a 255. Ocupa 8 bits de memória.
- Int - Armazena números inteiros e ocupa 16 bits de memória (2bytes). A faixa de valores é de -32.768 a 32.767.
- unsigned int - O mesmo que o int, porém a faixa de valores válidos é de 0 a 65.535.
- word - O mesmo que um unsigned int.
- long - Armazena números de até 32 bits (4 bytes). A faixa de valores é de -2.147.483.648 até 2.147.483.647.
- unsigned long - O mesmo que o long, porém a faixa de valores é de 0 até 4.294.967.295.
- short - Armazena número de até 16 bits (2 bytes). A faixa de valores é de -32.768 até 32.767.
- float - Armazena valores de ponto flutuante (com vírgula) e ocupa 32 bits (4 bytes) de memória. A faixa de valores é de -3.4028235E+38 até 3.4028235E+38
- double - O mesmo que o float.

DECLARAÇÃO DE VARIÁVEIS

A atribuição de valores a variáveis e constantes é feito com o uso do operador de atribuição “=”. Exemplos:

- int valor=10;
- const float pi = 3.14;

Importante:

- O operador de atribuição não vale para o comando #define.
- A linguagem de programação do Arduino, como dito anteriormente, é baseada no C/C++ portanto é Case Sensitive, diferenciando letras maiúsculas de minúsculas. Portanto a declaração “int valSensor;” é diferente de “int ValSensor;”.

VETORES E MATRIZES

Uma variável escalar pode armazenar muitos valores ao longo da execução do programa, porém, não ao mesmo tempo. Existem variáveis que podem armazenar mais de um valor ao mesmo tempo, essas variáveis são conhecidas como “variáveis compostas homogêneas”. A programação do Arduino permite trabalhar com dois tipos de variáveis compostas homogêneas: Vetores e Matrizes.

VETOR

A declaração de um vetor é feita da mesma maneira que uma variável escalar, entretanto é necessário definir o seu tamanho (quantidade de itens). Exemplo:

- `int vetor[5];`

- Cria um vetor de tamanho 5 (cinco) do tipo inteiro.

Para atribuir um valor a uma determinada posição do vetor, basta usar o índice, ou seja, a posição onde o valor será armazenado no vetor. Exemplo:

- `vetor[0] = 3;`

- Atribui o valor 3 a posição 0 do vetor.

Para acessar um valor em uma determinada posição do vetor, basta usar o índice, ou seja, a posição onde o valor está armazenado no vetor. Exemplo:

- `digitalWrite(vetor[0], HIGH);`

-Ativa a porta cujo número está definido na posição 0 do vetor.

MATRIZ

Uma matriz é similar a um vetor, entretanto pode ser formada por duas ou mais dimensões. Este elemento do tipo bidimensional possui um determinado número de linhas e de colunas. Exemplo:

- `int matriz[3][6];`

-Matriz com 3 (três) linhas e 6 (seis) colunas de elementos do tipo inteiro.

Para atribuir um valor a uma determinada posição da matriz, basta usar o Índice da linha e o índice da coluna, ou seja, a posição onde o valor será armazenado na matriz.

Exemplo:

- `matriz[1][2] = 6;`

-Atribui o valor 6 a posição 1 (linha), 2 (coluna) da matriz.

Para acessar um determinado valor em uma posição da matriz, basta usar o Índice da linha e o da coluna, ou seja, a posição onde o valor está armazenado na matriz. Exemplo:

- `digitalWrite(matriz[0][0], HIGH);`

-Ativa a porta cujo número está definido na posição 0 (linha), 0 (coluna) da matriz.

OPERADORES

Em uma linguagem de programação existem vários operadores que permitem operações do tipo: Aritmética, Relacional, Lógica e Composta.

ARITMÉTICOS

- + Adição
- - Subtração
- * Multiplicação
- / Divisão
- % Módulo (resto da divisão inteira)

RELACIONAIS

- > Maior
- < Menor
- >= Maior ou igual
- <= Menor ou igual
- == Igual
- != Diferente

LÓGICOS

- && E (AND)
- || OU (OR)
- ! NÃO (NOT)

COMPOSTOS

- ++ Incremento
- -- Decremento
- += Adição com atribuição
- -= Subtração com atribuição
- *= Multiplicação com atribuição
- /= Divisão com atribuição
- != Não Igual

COMENTÁRIOS

Muitas vezes é necessário comentar alguma parte do código do programa. Existem duas maneiras de adicionar comentários a um programa em Arduino.

- Comentário de Linha: //
// Este é um comentário de linha

- Comentário de Bloco /* */

/* Permite acrescentar comentários com mais de uma linha */

Nota: Quando o programa é compilado os comentários são automaticamente eliminados do arquivo executável (que será gravado no Arduino).

COMANDOS DE SELEÇÃO

Em vários momentos em um programa precisamos verificar uma determinada condição, possibilitando a seleção de uma ou mais ações que serão executadas. Um comando de seleção também é conhecido por desvio condicional, ou seja, dada um condição um parte do programa é executada. Os comandos de seleção podem ser do tipo:

- If - Seleção simples
- If/Else - Seleção composta
- Switch/Case/Break - Seleção de múltipla escolha

Simple

Um comando de seleção simples avalia uma determinada condição, ou expressão, para executar uma ação ou conjunto de ações. O comando de seleção simples é:

```

1  if (coloque aqui a expressão) {
2  comandos que serão executados ;
3  }

```

onde:

expressão — representa uma expressão a ser avaliada que pode ser do tipo lógica, relacional ou aritmética. O resultado da avaliação de uma expressão é sempre um valor lógico.

Comandos — comandos a serem executados.

COMPOSTA

Um comando de seleção composta é complementar ao comando de seleção simples. O objetivo é executar um comando mesmo que a expressão avaliada pelo comando **if** retorne um valor falso. O comando de seleção composta é:

```

1  if (expressão) {
2  comando 1;
3  }

```

```
4   else {
5   comando 2;
6   };
```

onde:

expressão — representa uma expressão a ser avaliada que pode ser do tipo lógica, relacional ou aritmética. O resultado da avaliação de uma expressão é sempre um valor lógico.

Comando 1 e 2 — comandos a serem executados.

MÚLTIPLA ESCOLHA

Na seleção de múltipla escolha é possível avaliar mais de um valor. O comando de seleção de múltipla escolha é:

```
1   switch (valor) {
2   case x: comando 1;
3   break;
4   case y: comando 2;
5   break;
6   default: comando3;
7   }
```

onde:

valor — é um dado a ser avaliado. É representado por uma variável de memória.

Comando 1, 2 e 3 — comandos a serem executados.

Case — indica a opção a ser executada.

default — opção padrão, no caso nenhuma alternativa tenha sido selecionada.

COMANDOS DE REPETIÇÃO

Muitas vezes é necessário repetir um ou mais trechos do código mais de uma vez, nestes casos devem ser utilizados os comandos de repetição para manter um “laço” em uma instrução ou conjunto de instruções. Os comandos de repetição podem ser:

- For — Baseado em um contador
- While — Baseado em uma expressão com teste no início

FOR - (REPETIR)

Este tipo de comando de repetição deve ser utilizado quando se sabe a quantidade de vezes que um determinado trecho do código deve ser executado. Exemplo:

```
1   for (contador início; expressão; incremento do contador) {  
2   comando 1;  
3   }
```

onde:

contador — é uma variável do tipo int

expressão — é uma expressão relacional

incremento do contador — passo de incremento do contador

WHILE - (ENQUANTO)

Este tipo de comando de repetição avalia uma expressão, enquanto for verdadeira, um determinado trecho do código permanece sendo executado. Exemplo:

```
1   while (expressão) {  
2   comando ;  
3   }
```

onde:

expressão — é uma expressão que pode ser lógica, relacional ou aritmética. A permanência de execução do “laço” é garantida enquanto a expressão for verdadeira.

Nota: Neste tipo de comando de repetição a avaliação da expressão é realizada no início do laço, ou seja, pode ser que o comando não execute nenhuma vez.

BIBLIOTECAS

Biblioteca é uma coleção de subprogramas utilizados no desenvolvimento de programas. Contém código e dados auxiliares, que provém serviços a programas independentes, o que permite o compartilhamento e a alteração de código e dados de forma modular.

TIPOS DE PORTAS E COMUNICAÇÃO SERIAL

O Arduino possui dois tipos de portas: Analógicas e Digitais, sendo este último tipo dividido entre binárias e PWM. Existe esta distinção de portas, que devem ser designadas de acordo com o resultado esperado com os componentes ligados a elas.

PORTAS DIGITAIS

As portas digitais trabalham com apenas dois valores de tensão: 0V e 5V. É importante frisar que os componentes ligados a estas portas só podem trabalhar com estas duas tensões, seja enviando ou recebendo dados. As principais funções para manipular as portas digitais são:

- **digitalRead (pino)**

Verifica a porta "pino" e retorna HIGH caso esteja em 5V e LOW caso esteja em 0V.

- **digitalWrite (pino, valor)**

Atribui os valores HIGH (5V) ou LOW (0V) para a porta "pino".

Nota: Apesar de possível não é recomendável utilizadas as portas digitais 0 e 1, pois elas estão conectadas diretamente ao sistema de comunicação do Arduino (pinos TX e RX, transmissão e recepção de dados, respectivamente).

Exemplo:

Geralmente quando aprendemos uma nova linguagem de programação, o primeiro programa mostrado é o famoso "Hello World", mas como ainda não aprendemos a trabalhar com o Display LCD, o nosso primeiro programa será piscar um Led.

estrutura:

```
1  int led = 13;
2
3  void setup() {
4  pinMode(led, OUTPUT);
5  }
6
7  void loop() {
8  digitalWrite (led, HIGH);
9  delay(1000);
10 digitalWrite(led, LOW);
11 delay(1000);
12 }
```

Após abrir este código, e configurar corretamente a placa e a porta COM utilizada, clique em Carregar. A mensagem esperada é "Carregado". Note que agora existe um led na placa

Arduino piscando com intervalos de 1 segundo. Gostou? :) Agora vamos explicar como tudo funciona.

O Arduino Uno, conforme comentado anteriormente possui 14 portas digitais, que podem ser usadas como Entrada ou Saída. Quando utilizamos uma porta como Entrada (INPUT) significa que será feita uma leitura desta porta, já como Saída (OUTPUT) significa que vamos “escrever” alguma coisa nela, ou seja, o pino poderá ser alimentado.

Na **linha 1** criamos uma variável do tipo “int” (vamos falar dos tipos de variáveis mais pra frente) e atribuímos o valor 13 a ela. Nada de mais até aí, certo? Mas quando dizemos **pinMode(led, OUTPUT)** que dizer “ModoDoPino (13, Saída)” significa que a porta 13 será de saída, sendo assim será possível atribuir um valor (HIGH = 5V ou LOW = 0V). Resumindo, será possível que tenha uma saída de 0V (led apagado) e 5V (led aceso).

No caso do Arduino Uno, a porta 13 possui um Led ligado internamente à placa, sendo assim quando esta porta fica em nível lógico alto (HIGH) a tensão nela será de 5V fazendo com que o Led acenda.

Em seguida chama-se a função Delay com o parâmetro 1000 que quer dizer algo como “Espere 1000 milissegundos (ou 1 segundo)”, e em seguida coloca-se a porta 13 novamente em nível lógico baixo (LOW) e executa novamente a função Delay.

Você pode observar que mesmo para um exemplo simples, existem duas funções:

- void setup() - serve para definir as configurações iniciais do programa, e é executada apenas uma vez.
- void loop() - é a função principal do programa e como o próprio nome já diz, é executada infinitamente até que a placa seja reiniciada ou desligada.

Estas são funções básicas, e devem ser utilizadas em todos os códigos, pois são necessárias para que a compilação do programa seja realizada corretamente. Caso esqueça uma delas, o código não compila.

Resumindo o funcionamento, o programa liga o Led, espera 1 segundo, desliga o led, espera 1 segundo. Ele faz isso sucessivamente até que a placa seja desligada ou reiniciada.

APENDICE C



RELIMAX
ROBÓTICA LIVRE



COLUN

COLÉGIO UNIVERSITÁRIO

TODOS OS DIREITOS RESERVADOS

Manual de Atividades - Laboratório
Sistema Automatizado



NELIO ALVES GUILHON

2023

COMPONENTES ELETRÔNICOS

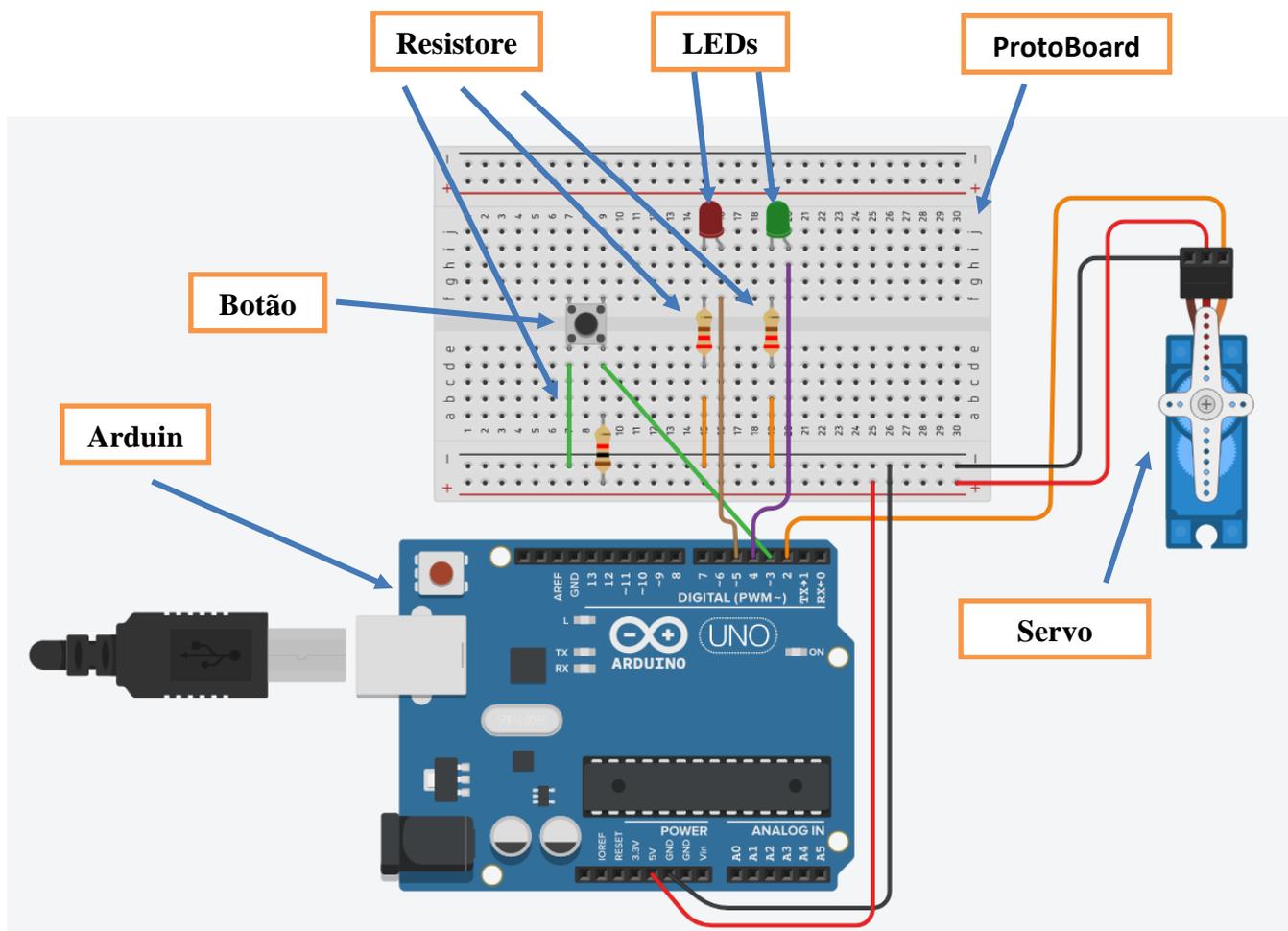
Este manual de atividades se concentra na popular plataforma de código aberto Arduino. Você irá conhecer vários componentes aplicados neste kit para construção de um circuito automatizado controlado por um **Sensor de Obstáculos Reflexivo Infravermelho**, que poderá ser usado para as mais diversas experiências, use a sua cabeça e vamos trabalhar!

QTD	DESCRIÇÃO	COMPONENTE ELETRÔNICO
01	Arduino nano	
01	Sensor de Obstáculos Reflexivo Infravermelho	
01	Micro Servo Motor 9g SG90	
Vários	Cabinhos Jumper: - Macho – Fêmea - Fêmea – Fêmea - Macho - Macho	
03	Resistores: - 220 Ω - 220 Ω - 1 K Ω	
02	LED's: - Verde - Vermelho	

ESQUEMA DE MONTAGEM

Com esse circuito podemos fazer vários projetos, bastando para isso trocar a programação e o **Botão 1** por qualquer outro sensor digital.

No caso do nosso projeto iremos trocar por um **Sensor de Obstáculos Reflexivo Infravermelho** conforme a página 3 e a codificação segundo as páginas 5 e 6.



LISTA DE PEÇAS:

- Arduino
- Micro servo motor
- Botão 1 (Substituir pelo Sensor de Obstáculos Reflexivo Infravermelho)
- Resistor 1 kΩ
- 2 X Resistor 220 Ω
- LED Vermelho
- LED Verde

CODIFICAÇÃO EM BLOCOS

The image shows a Scratch script for controlling a servo motor based on a digital sensor's state. The script is organized into several sections:

- no início** (at start):
 - definir temp como 0
 - girar servo no pino 3 em 0 graus
 - definir pino 4 como ALTO
 - definir pino 5 como BAIXO
- definição de estadoSensor**:
 - definir estadoSensor como ler pino digital 10
- se estadoSensor = 0, então** (if-then block):
 - se temp = 0, então** (if-then block):
 - definir pino 4 como BAIXO
 - definir pino 5 como ALTO
 - contagem para cima por 1 para i de 1 a 90 fazer:
 - girar servo no pino 3 em i graus
 - aguardar 0.1 s
 - definir temp como 1
 - aguardar 1 s
 - outro** (else block):
 - definir pino 4 como ALTO
 - definir pino 5 como BAIXO
 - contagem para baixo por 1 para i de 90 a 1 fazer:
 - girar servo no pino 3 em i graus
 - definir temp como 0
 - aguardar 2 s

CODIFICAÇÃO EM TEXTO

```
#include <Servo.h>

int temp = 0;
int i = 0;
int unnamed = 0;
int estadoSensor = 0;
Servo servo_3;

void setup()
{
  servo_3.attach(3, 500, 2500);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(10, INPUT);

  temp = 0;
  servo_3.write(0);
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
}

void loop()
{
  estadoSensor = digitalRead(10);
  if (estadoSensor == 0) {
    if (temp == 0) {
      digitalWrite(4, LOW);
      digitalWrite(5, HIGH);
      for (i = 1; i <= 90; i += 1) {
        servo_3.write(i);
        delay(100); // Wait for 100 millisecond(s)
      }
    }
  }
}
```

```
temp = 1;
delay(1000); // Wait for 1000 millisecond(s)
} else {
digitalWrite(4, HIGH);
digitalWrite(5, LOW);
for (i = 90; i >= 1; i -= 1) {
  servo_3.write(i);
}
temp = 0;
delay(2000); // Wait for 2000 millisecond(s)
}
}
}
```

TAREFAS

1ª ETAPA

- 1 – Abra a plataforma de **TINKERCAD DE SIMULAÇÃO** em:
<https://www.tinkercad.com/>
- 2 - Monte o circuito da página 4;
- 3 – Faça a programação segundo a página 5 (em bloco);
- 4 – Observe como fica a programação em texto resultante da programação em blocos na página 6;
- 5 – Depois de tudo montado clique no botão **Iniciar Simulação**
- 6 – Observe o que aconteceu e faça um relatório.

2ª ETAPA

- 1 – Faça a montagem do circuito na placa **ProntoBoard** ;
- 2 – Troque o Botão 1 pelo **Sensor de Obstáculos Reflexivo Infravermelho**;
- 3 – Observe a pinagem do sensor: **OUT – GND – VCC**;
- 4 – Abra o programa **Arduino IDE**;
- 5 – Depois cópia e cole a codificação em texto do **TINKERCAD** para o programa **Arduino IDE**;
- 6 – Siga a orientação do professor e configure a **PLACA, PROCESADOR e PORTA**;
- 7 – Clique em **Carregar** e acompanhe as mensagens do programa.
- 8 - Observe o que aconteceu e faça um relatório.





RELIMAX
ROBÓTICA LIVRE



COLUN

COLÉGIO UNIVERSITÁRIO

TODOS OS DIREITOS RESERVADOS

Manual de Atividades - Laboratório
Sistema Automatizado



NELIO ALVES GUILHON

2023

COMPONENTES ELETRÔNICOS

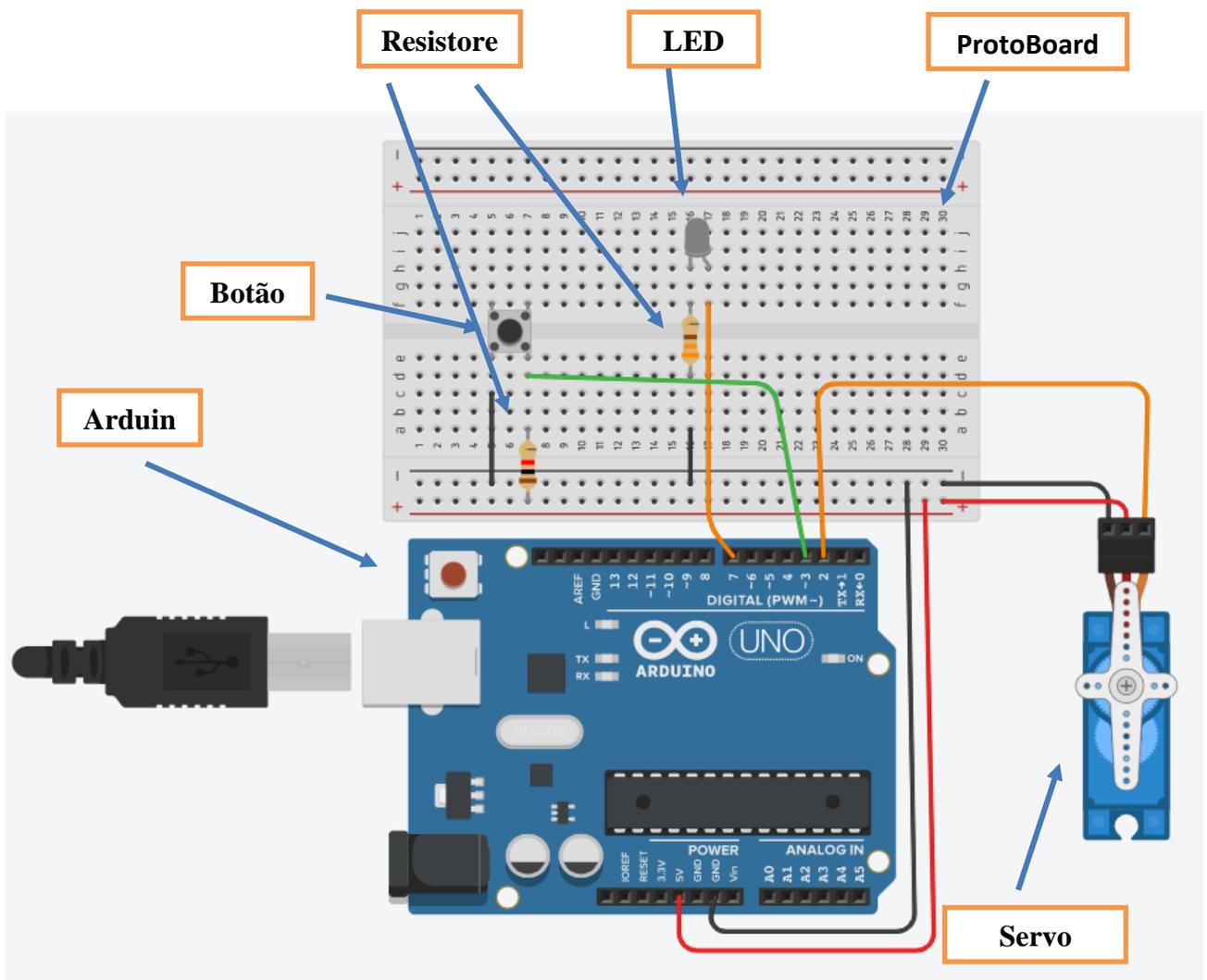
Este manual de atividades se concentra na popular plataforma de código aberto Arduino. Você irá conhecer vários componentes aplicados neste kit para construção de um circuito automatizado controlado por um **Sensor de Obstáculos Reflexivo Infravermelho**, que poderá ser usado para as mais diversas experiências, use a sua cabeça e vamos trabalhar!

QTD	DESCRIÇÃO	COMPONENTE ELETRÔNICO
01	Arduino nano	
01	Sensor de Obstáculos Reflexivo Infravermelho	
01	Micro Servo Motor 9g SG90	
Vários	Cabinhos Jumper: - Macho – Fêmea - Fêmea – Fêmea - Macho - Macho	
03	Resistores: - 220 Ω - 220 Ω - 1 K Ω	
02	LED's: - Verde - Vermelho	

ESQUEMA DE MONTAGEM

Com esse circuito podemos fazer vários projetos, bastando para isso trocar a programação e o **Botão 1** por qualquer outro sensor digital.

No caso do nosso projeto iremos trocar o **Botão 1** por um **Sensor de Obstáculos Reflexivo Infravermelho** conforme a página 3 e a codificação segundo as páginas 5 e 6.



LISTA DE PEÇAS:

- Arduino
- Micro servo motor
- Botão 1 (Substituir pelo Sensor de Obstáculos Reflexivo Infravermelho)
- Resistor 1 kΩ
- Resistor 220 Ω
- LED Vermelho

CODIFICAÇÃO EM BLOCOS



CODIFICAÇÃO EM TEXTO

```
#include <Servo.h>

int sensor = 0;
Servo servo_2;

void setup()
{
  servo_2.attach(2, 500, 2500);
  pinMode(3, INPUT);

  servo_2.write(90);
  delay(150); // Wait for 150 millisecond(s)
  servo_2.write(70);
}
void loop()
{
  sensor = digitalRead(3);
  if (sensor == 0) {
    servo_2.write(0);
    delay(300); // Wait for 300 millisecond(s)
    servo_2.write(70);
    delay(300); // Wait for 300 millisecond(s)
    servo_2.write(50);
    delay(300); // Wait for 300 millisecond(s)
    servo_2.write(60);
    delay(300); // Wait for 300 millisecond(s)
  } else {
    servo_2.write(70);
  }
}
```

TAREFAS

1ª ETAPA

- 1 – Abra a plataforma de **TINKERCAD DE SIMULAÇÃO** em:
<https://www.tinkercad.com/>
- 2 - Monte o circuito da página 4;
- 3 – Faça a programação segundo a página 5 (em bloco);
- 4 – Observe como fica a programação em texto resultante da programação em blocos na página 6;
- 5 – Depois de tudo montado clique no botão **Iniciar Simulação**
- 6 – Observe o que aconteceu e faça um relatório.

2ª ETAPA

- 1 – Faça a montagem do circuito na placa **ProntoBoard** ;
- 2 – Troque o Botão 1 pelo **Sensor de Obstáculos Reflexivo Infravermelho**;
- 3 – Observe a pinagem do sensor: **OUT – GND – VCC**;
- 4 – Abra o programa **Arduino IDE**;
- 5 – Depois cópia e cole a codificação em texto do **TINKERCAD** para o programa **Arduino IDE**;
- 6 – Siga a orientação do professor e configure a **PLACA, PROCESADOR e PORTA**;
- 7 – Clique em **Carregar** e acompanhe as mensagens do programa.
- 8 - Observe o que aconteceu e faça um relatório.





RELIMAX
ROBÓTICA LIVRE



COLUN

COLÉGIO UNIVERSITÁRIO

TODOS OS DIREITOS RESERVADOS

Manual de Atividades - Laboratório

Sistema Automatizado

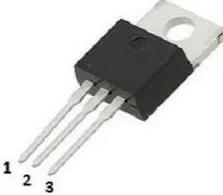


NELIO ALVES GUILHON

2023

COMPONENTES ELETRÔNICOS

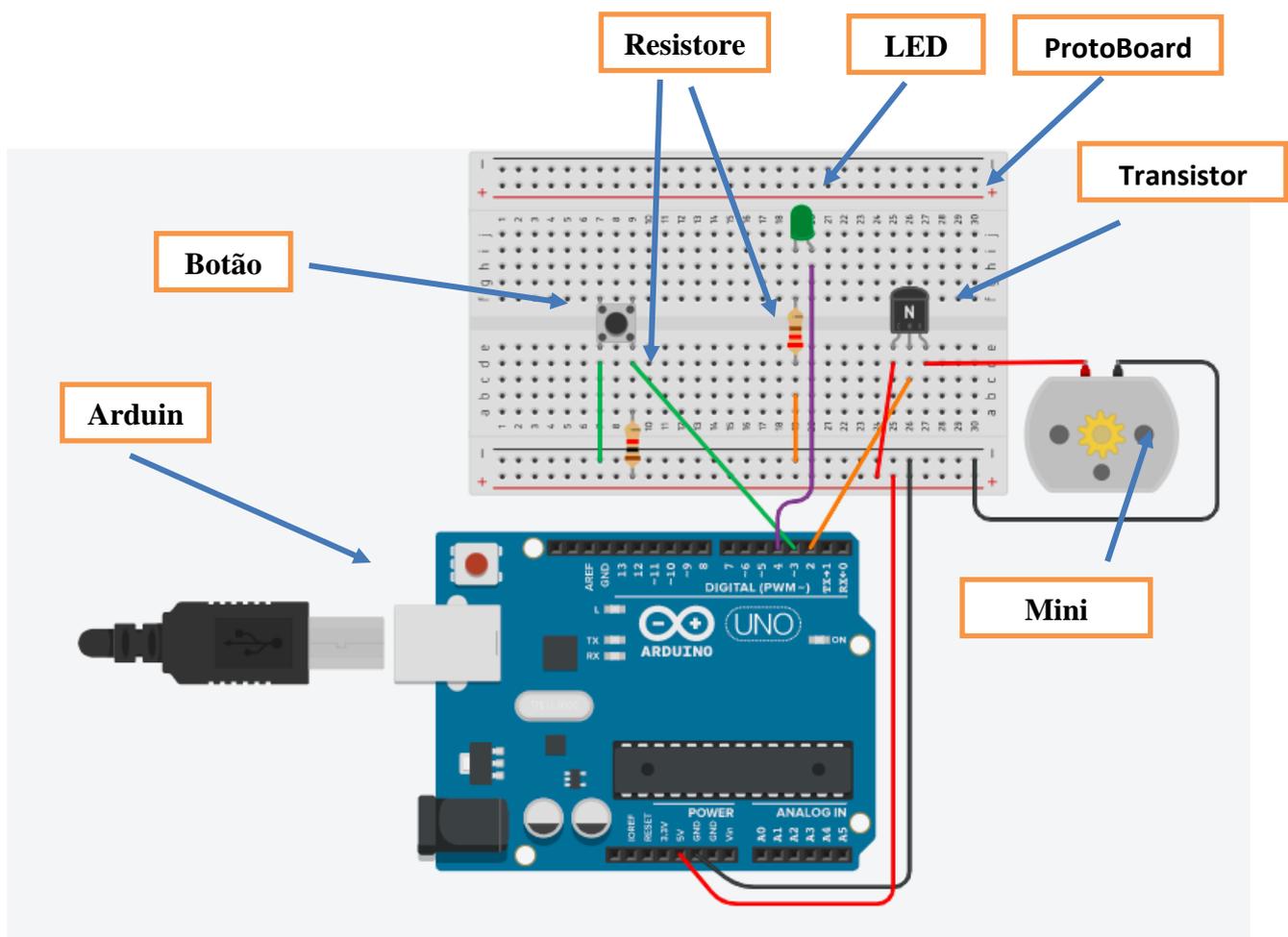
Este manual de atividades se concentra na popular plataforma de código aberto Arduino. Você irá conhecer vários componentes aplicados neste kit para construção de um automatizador controlado por um **Sensor de Umidade de Solo**, que poderá ser usado para as mais diversas experiências, use a sua cabeça e vamos trabalhar!

QTD	DESCRIÇÃO	COMPONENTE ELETRÔNICO
01	Arduino nano	
01	Modulo Sensor de Umidade de Solo	
01	Transistor TIP120	 <p>TIP120 pinout</p> <ol style="list-style-type: none"> 1. Base 2. Coletor 3. Emissor
Vários	Cabinhos Jumper: - Macho – Fêmea - Fêmea – Fêmea - Macho - Macho	
01	Resistores: - 220Ω - 1 k Ω	
01	LED: - Verde	
01	Mini Bomba Submersível p/ Água	

ESQUEMA DE MONTAGEM

Com esse circuito podemos fazer vários projetos, bastando para isso trocar a programação e o **Botão 1** por qualquer outro sensor digital.

No caso do nosso projeto iremos trocar o **Botão 1** por um **Sensor de Umidade de Solo**. A montagem do circuito encontramos na figura abaixo e a codificação em **Bloco** na página 5, enquanto a codificação em **Texto** na página 6.



LISTA DE PEÇAS:

- Arduino
- Micro servo motor
- Botão 1 (Trocar por um Sensor de Umidade de Solo)
- Resistor 1 k Ω
- Resistor 220 Ω
- LED Verde
- Mini Bomba Submersível p/ Água

CODIFICAÇÃO EM BLOCOS



CODIFICAÇÃO EM TEXTO

```
int sensorMao = 0;

void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, INPUT);
  pinMode(4, OUTPUT);

  digitalWrite(2, LOW);
}

void loop()
{
  sensorMao = digitalRead(3);
  if (sensorMao == 0) {
    digitalWrite(4, HIGH);
    digitalWrite(2, HIGH);
    delay(600); // Wait for 600 millisecond(s)
  } else {
    digitalWrite(2, LOW);
    digitalWrite(4, LOW);
  }
}
```

TAREFAS

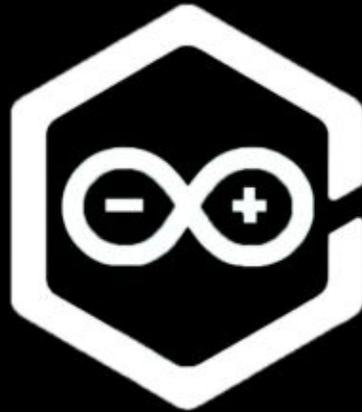
1ª ETAPA

- 1 – Abra a plataforma de **TINKERCAD DE SIMULAÇÃO** em:
<https://www.tinkercad.com/>
- 2 - Monte o circuito da página 4;
- 3 – Faça a programação segundo a página 5 (em bloco); ou
- 4 – Observe como fica a programação em texto resultante da programação em blocos na página 6;
- 5 – Depois de tudo montado clique no botão **Iniciar Simulação**
- 6 – Observe o que aconteceu e faça um relatório.

2ª ETAPA

- 1 – Faça a montagem do circuito na placa **Protoboard**;
- 2 – Troque o Botão 1 pelo **Sensor de Umidade de solo**;
- 3 – Observe a pinagem do sensor:
A0 – DO – GND – VCC;
- 4 – Abra o programa **Arduino IDE**;
- 5 – Depois cópia e cole a codificação em texto do **TINKERCAD** para o programa **Arduino IDE**;
- 6 – Siga a orientação do professor e configure a **PLACA, PROCESADOR e PORTA**;
- 7 – Clique em **Carregar** e acompanhe as mensagens do programa.
- 8 - Observe o que aconteceu e faça um relatório.





RELIMAX
ROBÓTICA LIVRE



COLUN

COLÉGIO UNIVERSITÁRIO

TODOS OS DIREITOS RESERVADOS

Manual de Atividades - Laboratório

Sistema Automatizado

4

NELIO ALVES GUILHON

2023

COMPONENTES ELETRÔNICOS

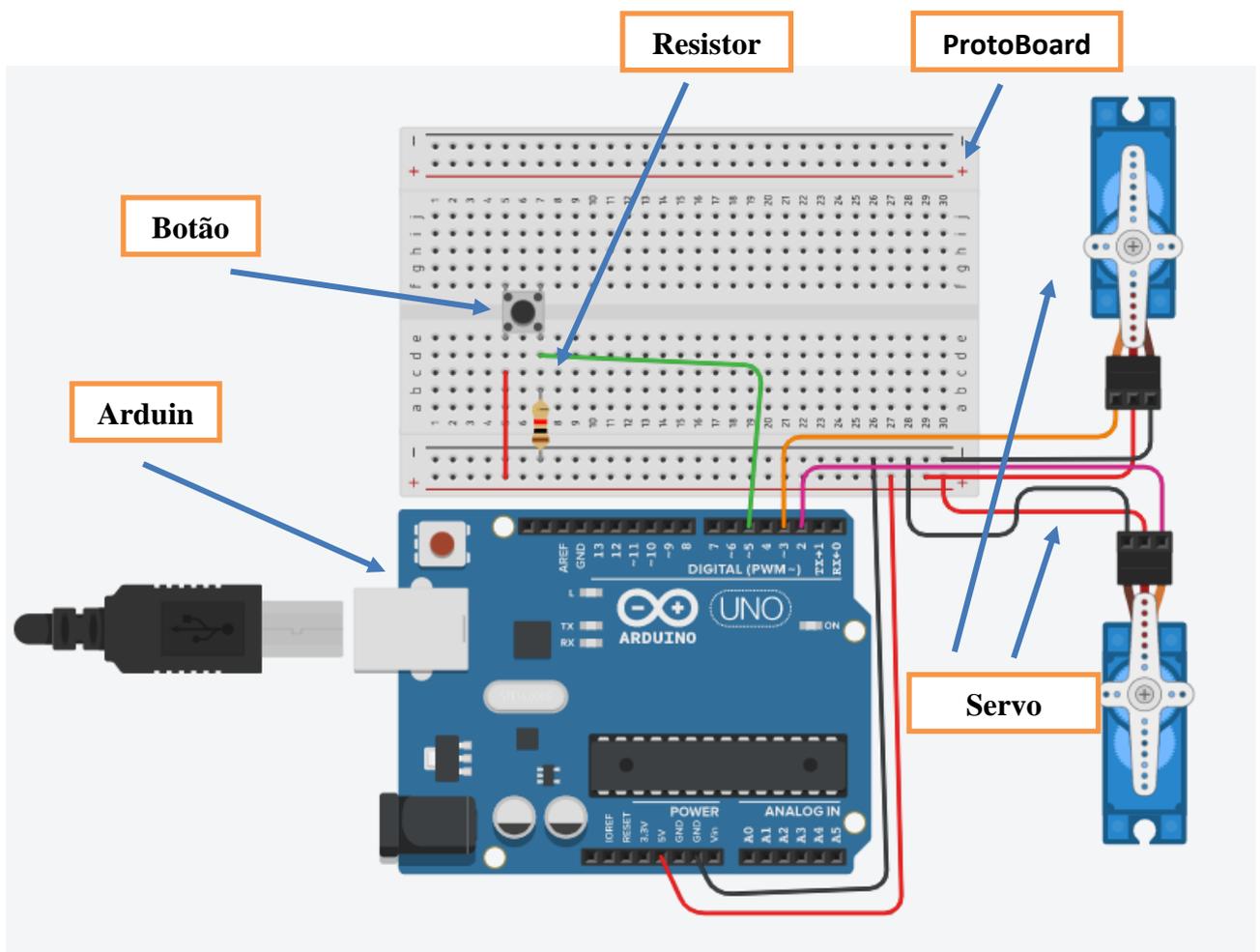
Este manual de atividades se concentra na popular plataforma de código aberto Arduino. Você irá conhecer vários componentes aplicados neste kit para construção de um automatizador controlado por um **Sensor de Som**, que poderá ser usado para as mais diversas experiências, use a sua cabeça e vamos trabalhar!

QTD	DESCRIÇÃO	COMPONENTE ELETRÔNICO
01	Arduino nano	
01	Modulo Sensor de Som	
02	Micro Servo Motor 9g SG90	
Vários	Cabinhos Jumper: - Macho – Fêmea - Fêmea – Fêmea - Macho - Macho	

ESQUEMA DE MONTAGEM

Com esse circuito podemos fazer vários projetos, bastando para isso trocar a programação e o **Botão 1** por qualquer outro sensor digital.

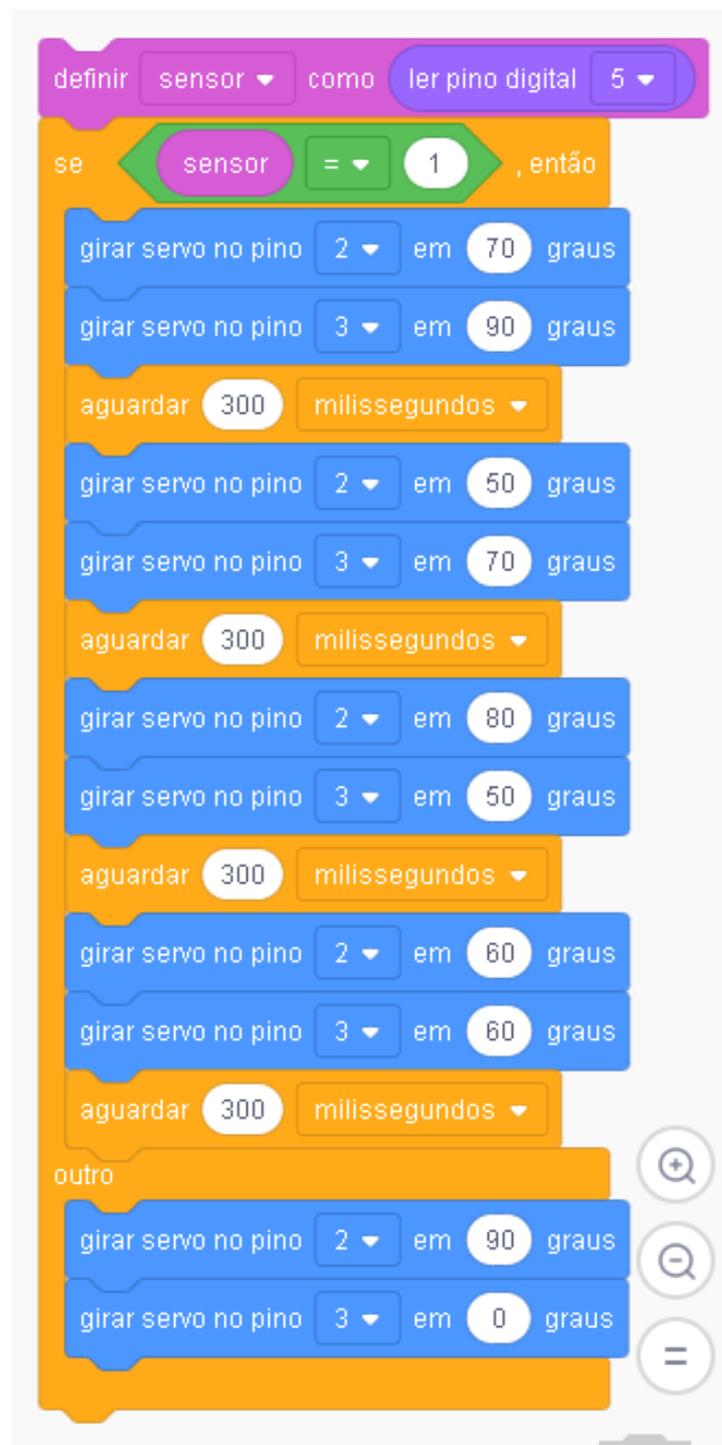
No caso do nosso projeto iremos trocar o **Botão 1** por um **Sensor de Som**. A montagem do circuito encontramos na figura abaixo e a codificação em **Bloco** na página 5, enquanto a codificação em **Texto** na página 6.



LISTA DE PEÇAS:

- Arduino
- 2 Micro servo motor
- Botão 1 (Trocar por um Sensor de Som)
- Resistor 1 kΩ

CODIFICAÇÃO EM BLOCOS



CODIFICAÇÃO EM TEXTO

```
#include <Servo.h>
int sensor = 0;
Servo servo_2;
Servo servo_3;

void setup()
{
  pinMode(5, INPUT);
  servo_2.attach(2, 500, 2500);
  servo_3.attach(3, 500, 2500);
}

void loop()
{
  sensor = digitalRead(5);
  if (sensor == 1) {
    servo_2.write(70);
    servo_3.write(90);
    delay(300); // Wait for 300 millisecond(s)
    servo_2.write(50);
    servo_3.write(70);
    delay(300); // Wait for 300 millisecond(s)
    servo_2.write(80);
    servo_3.write(50);
    delay(300); // Wait for 300 millisecond(s)
    servo_2.write(60);
    servo_3.write(60);
    delay(300); // Wait for 300 millisecond(s)
  } else {
    servo_2.write(90);
    servo_3.write(0);
  }
}
```

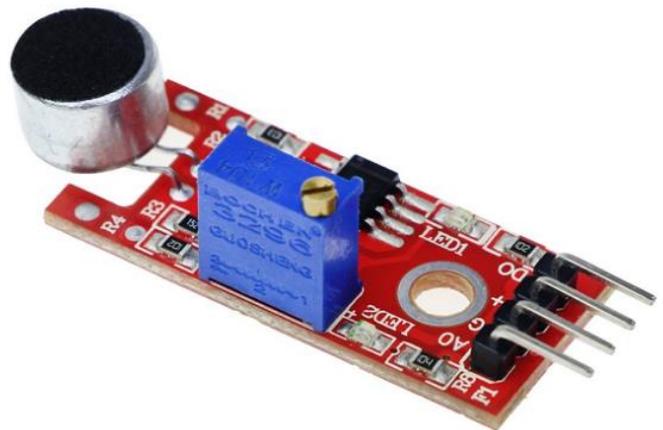
TAREFAS

1ª ETAPA

- 1 – Abra a plataforma de **TINKERCAD DE SIMULAÇÃO** em:
<https://www.tinkercad.com/>
- 2 - Monte o circuito da página 4;
- 3 – Faça a programação segundo a página 5 (em bloco); ou
- 4 – Observe como fica a programação em texto resultante da programação em blocos na página 6;
- 5 – Depois de tudo montado clique no botão **Iniciar Simulação**
- 6 – Observe o que aconteceu e faça um relatório.

2ª ETAPA

- 1 – Faça a montagem do circuito na placa **Protoboard**;
- 2 – Troque o Botão 1 pelo **Sensor de Som**;
- 3 – Observe a pinagem do sensor:
A0 – GND – VCC – DO;
- 4 – Abra o programa **Arduino IDE**;
- 5 – Depois cópia e cole a codificação em texto do **TINKERCAD** para o programa **Arduino IDE**;
- 6 – Siga a orientação do professor e configure a **PLACA, PROCESADOR e PORTA**;
- 7 – Clique em **Carregar** e acompanhe as mensagens do programa.
- 8 - Observe o que aconteceu e faça um relatório.



APENDICE G

DADOS SOCIODEMOGRÁFICOS

Qual o seu nome ou e-mail ?

Qual a sua idade

Qual o seu gênero?

Você cursa que série no COLUN ?

Já tinha participado de alguma capacitação em Robótica?

Você já tinha programado alguma vez?

PRÁTICA INICIAL

1) Você compreendeu o propósito da tarefa?

2) Teve dificuldade em programar o circuito para que ele operasse como um semáforo?

3) Na hora de simular o circuito na plataforma Tinkercad funcionou tudo normalmente?

4) Na hora de montar o circuito na placa de montagem houve problemas?

5) Conseguiu entender o funcionamento dos componentes eletrônicos, sensores e atuadores?

RESULTADOS

6) A prática realizada auxiliou na clareza de novos conceitos de programação, você compreendeu o propósito da tarefa?

7) Os conceitos de programação em bloco abordados foram adequados para a tarefa proposta?

8) O uso da Plataforma de Simulação e a programação em blocos facilitou sua compreensão na programação?

9) Você acredita que é mais simples aprender programação através de uma linguagem textual?

10) Você acredita que é mais simples aprender programação usando os blocos ?

11) Você acredita que é mais simples iniciar um projeto de robótica programando com blocos?

12) O uso da Robótica contribuiu para a sua compreensão nos princípios de

programação?

13) Você considera que a robótica torna a programação mais envolvente e divertida?

14) Você considera que a montagem do semáforo foi relevante para sua aprendizagem?

APENDICE H

PROPOSTA DO MODELO TAM PARA A PESQUISA

O Modelo de Aceitação de Tecnologia pode ser adaptado de maneira aprofundada para oferecer uma compreensão aprimorada sobre como a RE pode ser implementada de forma mais eficaz para otimizar os benefícios educacionais. O quadro de referência proposto para esta pesquisa compreende 14 questões organizadas em cinco grupos, permitindo que os entrevistados tenham clareza sobre como suas respostas contribuem para o estudo.

- 1) Facilidade de Uso Percebida (FUP) – 4 questões – Escala de Likert
- 2) Utilidade percebida (PU) – 4 questões – Escala de Likert
- 3) Atitude comportamental (AC) – 2 questões – Escala de Likert
- 4) Intenção de Uso (IU) – 2 questões – Escala de Likert
- 5) Uso Real do Sistema (URS). – 2 questões – Escala de Likert

Quadro 1: Instrumento de pesquisa

DIMENSÃO/CONCEITOS	VARIÁVEIS
Facilidade de Uso Percebida (FUP)	<p>FUP1 Aprender a programar o Arduino é fácil?</p> <p>FUP2 Tenho a capacidade de me tornar um especialista em Arduino</p> <p>FUP3 Com o Arduino, aprimoro minhas habilidades de programação</p> <p>FUP4 Eu sei facilmente como proceder para usar o Arduino?</p>
Utilidade Percebida (UP)	<p>UP1 Utilizando o Arduino, amplio meu poder de raciocínio</p> <p>UP2 Penso que o Arduino contribui positivamente para a aprendizagem de programação</p> <p>UP3 O conhecimento em robótica amplia as chances de obter melhores oportunidades no mercado de trabalho.</p> <p>UP4 Ao utilizar a robótica, minha performance apresenta melhorias.</p>
Atitude Comportamental (AC)	<p>AC1 Utilizar o Arduino é uma ótima ideia.</p> <p>AC2 Me agrada a ideia de incorporar a robótica em meus estudos</p>
Intenção de Uso (IU)	<p>ITU1 Eu pretendo utilizar o Arduino sempre que possível</p> <p>ITU2 Eu tenho a intenção de aumentar o uso do arduino</p>
Uso Real do Sistema (URS)	<p>US1 Uso o Arduino raramente</p> <p>US2 Durante a oficina, utilizava frequentemente a plataforma Tinkercad</p>

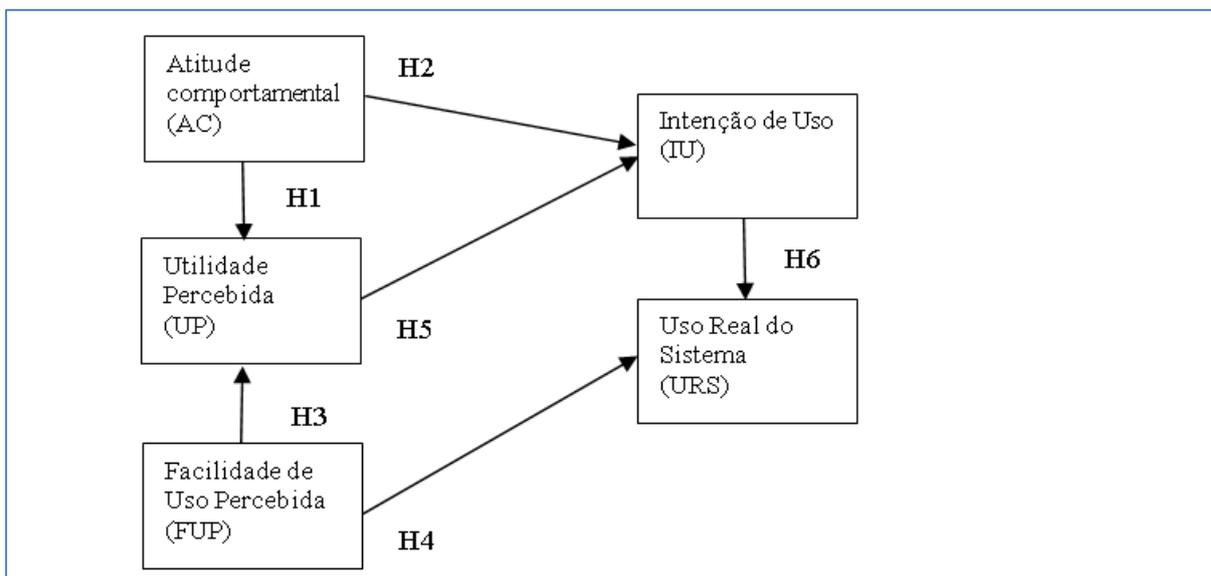
Fonte: Adaptado de Gomes (2022).

No modelo ilustrado na (Figura 1), é possível visualizar as dimensões propostas. O desígnio central deste modelo conceitual é elucidar a conexão entre diversos fatores que podem impactar na aceitação da RE pelos usuários. Desta forma, incorporou-se o construtor “Atitude Comportamental” que diz respeito à avaliação pessoal, seja positiva ou negativa, em relação a um determinado comportamento (HEDLER *et al.*, 2016).

O modelo desenvolvido incorpora uma nova dimensão, além daquelas já presentes no modelo de Davis (1989), levando à formulação de seis hipóteses de trabalho “H1”, “H2”, “H3”, “H4”, “H5” e “H6”, A partir desses insights, é proposto um modelo que é fundamentado nas seguintes suposições:

1. **H1** – Atitude Comportamental afeta Utilidade Percebida da RE.
2. **H2** – Atitude Comportamental afeta a Intenção de Uso da RE.
3. **H3** – Facilidade de Uso Percebida tem efeito Utilidade Percebida da RE
4. **H4** – Facilidade de Uso Percebida tem efeito sobre o Uso Real da RE.
5. **H5** – A Utilidade Percebida tem ação sobre influencia a Intenção de Uso da RE.
6. **H6** – A Intenção de Uso tem impacto sobre Uso Real da RE.

Figura 1 – MODELO DE INVESTIGAÇÃO.



Fonte: Adaptado de Davis (1989).

Em relação às hipóteses listadas, espera-se entender melhor como as atitudes e percepções dos usuários influenciam diretamente a aceitação e a utilização da RE. As hipóteses sugerem umnexo causal entre a atitude comportamental e diferentes dimensões da

aceitação da tecnologia, tais como a utilidade percebida e a intenção de uso. A facilidade percebida de uso é vista como um fator crucial que pode afetar tanto a utilidade percebida quanto o uso real da RE. Em suma, essas hipóteses buscam oferecer uma estrutura para investigar e compreender os fatores determinantes na adoção e efetiva utilização da RE como ferramenta.