

NextEd

Formação Continuada Docente em Robótica Educativa

Augusto César Castro Ribeiro

Rejane Waiandt Schuwartz de Carvalho Faria

UFV
Universidade Federal de Viçosa

NERG
Núcleo de Especialização em Robótica

NAVEGANDO

GATE
Grupo de Atenção à Tecnologia na Educação

FAPEMIG

NextEd

**Formação Continuada Docente em
Robótica Educacional**

Augusto César Castro Ribeiro
Rejane Waiandt Schuwartz de Carvalho Faria

NextEd

Formação Continuada Docente em Robótica Educacional



NAVEGANDO

Navegando Publicações
Uberlândia – Brasil
2026



www.editoranavegando.com
editoranavegando@gmail.com
Uberlândia – MG
Brasil

Direção Editorial: Navegando Publicações
Diagramação: Lurdes Lucena

Copyright © by autor, 2026.

F7244 – RIBEIRO, Augusto César Castro. FARIA, Rejane Waiandt Schuwartz de Carvalho. NextEd - Formação Continuada Docente em Robótica Educacional. Uberlândia: Navegando Publicações, 2026.

ISBN: 978-65-6070-162-5

DOI: 10.29388/978-65-6070-162-5

1. Robótica. 2. Tecnologia. 3. Educação. I. Augusto César Castro Ribeiro. Rejane Waiandt Schuwartz de Carvalho Faria. II. Navegando Publicações. Título.

CDD – 370

Navegando Publicações



NAVEGANDO

www.editoranavegando.com

editoranavegando@gmail.com

Uberlândia - MG

Brasil

Editores

Lurdes Lucena - Esamc - Brasil

Carlos Lucena - UFU - Brasil

José Claudinei Lombardi - Unicamp - Brasil

José Carlos de Souza Araújo - Uninube/UFU - Brasil

Conselho Editorial Multidisciplinar

Pesquisadores Nacionais

Afrânio Mendes Catani - USP - Brasil

Anderson Bretas - IFPA - Brasil

Antônio Alencar Colares - UFOPA - Brasil

Carlos Lucena - UFU - Brasil

Carlos Henrique de Carvalho - UFU, Brasil

Claudio César Fagiani - Uninube - Brasil

Dermeval Saviani - Unicamp - Brasil

Elmiro Santos Resende - UFU - Brasil

Fabiane Santana Previtali - UFU - Brasil

Gilberto Luiz Alves - UFMS - Brasil

Inez Stampa - PUCRJ - Brasil

José Carlos de Souza Araújo - Uninube/UFU - Brasil

José Claudinei Lombardi - Unicamp - Brasil

Laísza Dahmer Pereira - UFF - Brasil

Lúcia Diana Rocha Magalhães - UESB - Brasil

Marcelo Caetano Faureira da Silva - UFU - Brasil

Maria Regina Martins Jacomeli - Unicamp, Brasil

Maria J. A. Rosário - UFPA - Brasil

Newton Antonio Pacelli Bryan - Unicamp, Brasil

Paulino José Oso - Unioeste - Brasil

Ricardo Antunes - Unicamp, Brasil

Robson Luiz de França - UFU, Brasil

Tatiana Dahmer Pereira - UFF - Brasil

Valeria Lucília Forti - UERJ - Brasil

Yolanda Guerra - UFRJ - Brasil

Pesquisadores Internacionais

Alberto L. Balakowsky - Universidad de Buenos Aires - Argentina

Alecia Maria de Castro Martins - (ISMIT), Coimbra - Portugal

Alexander Steffens - Lee University - EUA

Angela A. Fernandez - Univ. Aut. de St. Domingo - Rep. Dominicana

Antonio Vidal Ortega - Pont. Un. Cat. M. y M. - Rep. Dominicana

Armando Martínez Rosales - Universidad Popular de Cesar - Colômbia

Artemis Torres Valenzuela - Universidad San Carlos de Guatemala - Guatemala

Carolina Cusano - Universidad de Buenos Aires - Argentina

Christian Cwik - Universität Graz - Austria

Christian Hauser - Universidad de Talca - Chile

Daniel Schugrensky - Arizona State University - EUA

Elizet Payne Igeles - Universidad de Costa Rica - Costa Rica

Elsa Capson - Université de Nantes / Univ. de la Réunion - France

Elvira Abadillo Mosell - Vanderbilt University - EUA

Fernando Camacho Pedraza - Univ. Autónoma de Madrid - Espanha

José María Alvarado - Universidad de Cartagena - Colômbia

Heleen Veenendaal Delgado - Univ. Autónoma de Coahuila - México

Isabel Giegge - Universidade de Coimbra - Portugal

Isis Sánchez - Universidad del Magdalena - Colômbia

Johanna von Gersdorff, Instituto Mora - México

Lionel Muñoz Paz - Universidad Central de Venezuela - Venezuela

Jorge Enrique Elián-Caro - Universidad del Magdalena - Colômbia

José Jesús Borjón Nieto - El Colegio de Vex Cruz - México

José Luis de los Reyes - Universidad Autónoma de Madrid - Espanha

Juan Marchena Fernández - Universidad Pablo de Olavide - Espanha

Juan Paz y Miño Cepeda, Pont. Univ. Católica del Ecuador - Ecuador

Leiber Dimes Vasquez - Universidad de La Guajira - Colômbia

Marvin Barahona - Universidad Nacional Autónoma de Honduras - Honduras

Michael Zentke - Universität zu Köln - Alemanha

Miguel Pérez - Universidade Nova Lisboa - Portugal

Pilar Cagiao Vila - Universidad de Santiago de Compostela - Espanha

Raul Roman Romero - Univ. Nacional de Colombia - Colômbia

Roberto González Arana - Universidad del Norte - Colômbia

Ronny Vales Hurtado - Universidad de Costa Rica - Costa Rica

Rosana de Motos Silveira Santos - Universidad de Granada - Espanha

Rosario Márquez Macías, Universidad de Huelva - Espanha

Sergio Guerra Vilhoy - Universidad de la Habana - Cuba

Silvia Mancini - Université de Lausanne - Suíça

Tezeta Medina - Universidade do Minho - Portugal

Taizun MacCovey - University of London - Inglaterra

Victor-Jacinto Flecha - Univ. Cat. N. Señora de la Asunción - Paraguai

Yael Condori Núñez - Instituto de Historia de Cuba - Cuba

Agradecimentos

Agradecemos à FAPEMIG pelo financiamento dos projetos “Educação Matemática e Robótica Social: potencialidades e desafios no contexto da Educação Básica” (FAPEMIG/APQ-04493-23) e “Formação de professores para transformação da Educação Básica por meio da Robótica Educacional” (FAPEMIG/APQ-06673-24), cujo apoio foi fundamental para o desenvolvimento das ações de pesquisa e formação apresentadas neste livro. Agradecemos aos membros do Grupo de Atenção às Tecnologias na Educação (GATE) e do Núcleo de Especialização em Robótica (NERo), ambos da Universidade Federal de Viçosa, pelo apoio, acolhimento, parceria e pelas valiosas contribuições ao longo desta trajetória. As trocas de experiências, os diálogos e o compromisso coletivo com a pesquisa, a formação e a inovação educacional foram fundamentais para a construção deste trabalho. Nosso sincero reconhecimento pela dedicação, colaboração e constante incentivo ao desenvolvimento acadêmico e científico.



Prefácio

A obra "NextEd: Formação Continuada Docente em Robótica Educacional", de autoria dos colegas Augusto César Castro Ribeiro e Rejane Waiandt Schuwartz de Carvalho Faria, vinculados à Universidade Federal de Viçosa (UFV), nos convida a refletir sobre o momento atual, mas já com um olhar no futuro. As transformações tecnológicas profundas, onde a escola é convocada a preparar cidadãos para um mundo em constante mutação, privilegiando o pensamento crítico e a resolução de problemas, compõem o contexto de produção deste livro. A obra idealizada por Augusto, experiente professor de matemática da Educação Básica e mestre em Educação, e por Rejane, formadora de professores de matemática e pesquisadora atenta, emerge não apenas como um manual técnico sobre o uso da Robótica na sala de aula, mas como uma proposta pedagógica robusta e necessária para o fortalecimento da Educação Básica brasileira. Mais do que um manual, sobretudo, a obra possibilita múltiplas leituras, pois é material didático, é resultado de pesquisa científica, é livro que dialoga com professores e outros profissionais de Educação.

O material destaca-se pelo seu elevado potencial formativo, especialmente por dirigir-se ao professor da Educação Básica que, embora muitas vezes sem formação técnica prévia na área, deseja integrar a Robótica em suas práticas. A pertinência do tema é inquestionável. A Robótica Educacional é apresentada como um recurso interdisciplinar capaz de promover a aprendizagem ativa e preencher lacunas históricas no ensino de Computação, Matemática e Física, sobretudo, em tempos de implementação da Base Nacional Comum Curricular (BNCC) - Computação em todas as redes de ensino do Brasil.

A organização do livro revela um cuidadoso planejamento didático dos autores. A transição entre a teoria fundamental, discutindo o que é um robô e a perspectiva de Seymour Papert, e a prática, com a plataforma Arduino e o simulador Tinkercad, ocorre de forma fluida e acessível.

Outro cuidado que os autores tiveram se relaciona à estrutura e organização do texto, uma vez que os capítulos são apresentados de forma a reduzir uma possível "frustração" comum no ensino tradicional (baseado na entrega de conteúdo e exemplos, apenas), optando por uma abordagem prazerosa e visual. Do ponto de vista didático-pedagógico, as propostas práticas, como o projeto "Piscar LED" e o "Semáforo", são acompanhadas por seções de "Explorando a Matemática", que conectam a experimentação a conceitos abstratos como a continuidade de funções.

Outro ponto de destaque se refere ao alinhamento com Políticas Educacionais vigentes, como já mencionado anteriormente, pois o material está em estreita harmonia com as demandas da BNCC, ao fomentar competências como a cultura digital, o pensamento científico e a autonomia. Mas, mais do que atender às demandas apontadas pela legislação educacional, ao utilizar

ferramentas de código aberto e simuladores gratuitos, os autores garantem a viabilidade da proposta em diferentes contextos socioeconômicos. Portanto, a proposta do livro transcende o que está posto no currículo e pretende ser um produto educacional que democratiza o conhecimento científico.

Não poderia deixar de mencionar, ainda, que um diferencial desta obra reside na sua origem. Sua narrativa é construída a partir do diálogo entre a Computação e a Matemática, fruto de uma pesquisa de mestrado que entende a Robótica como meio, e não apenas fim. Esse aspecto contribui para se pensar outras propostas pedagógicas, de pesquisa ou de caráter extensionista na perspectiva STEAM, uma abordagem educacional interdisciplinar que integra cinco áreas do conhecimento: Science (Ciência), Technology (Tecnologia), Engineering (Engenharia), Arts (Artes) e Mathematics (Matemática), a qual tem promovido espaços de produção de conhecimento bastante promissores nos mais diversos níveis e contextos educacionais. Embora essa discussão não tenha sido objeto de análise do trabalho realizado por Augusto e Rejane nesta obra, fica evidente o potencial de sua proposta para projetos futuros e correlacionados.

Esse livro, portanto, é um convite à inovação pedagógica. Pela forma didática que desenvolve o tema, o rigor científico e a sensibilidade docente, recomendo com entusiasmo a leitura e utilização deste material por professores, gestores e todos os profissionais comprometidos com uma educação que seja, de fato, transformadora. Trata-se de uma leitura indispensável para quem busca o avanço e o fortalecimento de uma escola inovadora, de qualidade e alinhada às demandas atuais.

Profª. Dra. Silvana Claudia dos Santos
Docente da Universidade Federal de Viçosa
Coordenadora do Grupo de Atenção às Tecnologias na Educação

Apresentação

É com imensa satisfação que apresentamos o NextEd - Formação Continuada Docente em Robótica Educacional. Neste livro está disponibilizado um material didático desenvolvido para levar ao professor o conhecimento necessário para iniciar seus estudos em Robótica Educacional e fazer parte da implantação dos artefatos robóticos e tecnológicos em sala de aula.

Somos Augusto César Castro Ribeiro e Rejane Waiandt Schuwartz de Carvalho Faria, mestrando e orientadora, ambos vinculados ao Programa de Pós-Graduação em Educação da Universidade Federal de Viçosa. Em 2024, iniciamos a pesquisa “Robótica Educacional na Formação Continuada de Professores: uma perspectiva da Educação Matemática” por entendermos que a Robótica Educacional pode ser rica em colaborar com o ensino da Computação, além de ser de grande valor para o ensino de matemática.

Vivemos em uma era em que a tecnologia permeia todos os aspectos da vida cotidiana, transformando a forma como nos comunicamos, aprendemos e trabalhamos. Nesse contexto, a escola desempenha um papel fundamental: preparar os alunos para um mundo em constante mudança, onde a criatividade, o pensamento crítico e a resolução de problemas são habilidades tão importantes quanto a leitura e a escrita.

Nesse cenário, a Robótica Educacional se mostra como uma poderosa aliada. Mais do que apenas montar robôs ou programar comandos, ela é uma ferramenta pedagógica interdisciplinar que promove a aprendizagem ativa e significativa. Ela estimula a curiosidade dos alunos, ao mesmo tempo em que desenvolve competências interdisciplinares e o trabalho em equipe.

Este material foi pensado especialmente para você, professor da Educação Básica, que deseja explorar o universo da Robótica Educacional, mesmo sem ter formação técnica na área. Nos propomos a explorar conceitos, apresentar ferramentas acessíveis e oferecer propostas práticas para que você se sinta confortável para incorporar a Robótica na sua sala de aula.

Ao longo dos capítulos, abordaremos conceitos sobre a Robótica, a utilização do Arduino, os componentes eletrônicos que podem ser utilizados, a utilização do Tinkercad, como programar para realizar as tarefas e como utilizar o Arduino IDE para ver o código funcionando na prática, além de sugestões de atividades que podem ser adaptadas ao contexto da sua realidade educacional.

Desta forma, esperamos que este material contribua para o seu enriquecimento profissional e para a expansão da Robótica no contexto educacional. Seja bem-vindo a esta jornada de descobertas, onde tecnologia e educação caminham juntas para formar cidadãos mais críticos, criativos e preparados para o futuro.

Os Autores

Sobre os autores

Sou Augusto César Castro Ribeiro, professor de Matemática nos Anos Finais do Ensino Fundamental na Escola Estadual Raul de Leoni e na Escola Municipal Padre Francisco José da Silva, ambas em Viçosa-MG. Primeiramente, me graduei em Engenharia de Computação pelo CEFET-MG, mas sempre tive o sonho de ensinar matemática e de educar. Assim, ingressei posteriormente na UFV e me formei em Licenciatura em Matemática. Em minha trajetória tive a oportunidade de lecionar em diferentes redes de ensino, trabalhando em escolas particulares, cursinhos preparatórios, escolas estaduais e municipais mineiras. Durante a minha primeira graduação confesso que uma das grandes frustrações que tive foi a maneira pesada que a programação era abordada e ensinada, muitas vezes sem a oportunidade de ser introduzida de maneira clara de forma que pudesse visualizar resultados naquilo que era feito. Dessa maneira, ao ingressar no mestrado pude ver na Robótica uma oportunidade de levar uma maneira mais prazerosa de iniciar no mundo da programação.



Sou Rejane Waiandt Schuwartz de Carvalho Faria, graduada em Licenciatura em Matemática pelo Instituto Federal Fluminense (IFF, 2009). Mestre e Doutora em Educação Matemática pela Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP, 2012, 2016). Professora Associada da Universidade Federal de Viçosa (UFV) no Departamento de Matemática (DMA) e no Programa de Pós-Graduação em Educação (PPGE). Membro do Grupo de Atenção às Tecnologias na Educação (GATE/UFV), do Núcleo de Especialização em

Robótica (NERO/UFV). Minha trajetória docente foi iniciada na Educação Básica, nos Anos Finais do Ensino Fundamental e no Ensino Médio, nas modalidades Regular e EJA, nas redes pública e privada. Atuando no Ensino Superior, me mantenho próxima da Educação Básica, lecionando em disciplinas como Estágio Curricular Supervisionado e Prática Pedagógica na Licenciatura em Matemática, além de ter atuado como Coordenadora da Área de Matemática do Programa Institucional de Bolsas de Iniciação à Docência (PIBID/UFV).

Nas fotos estão nossas famílias. Na primeira, Augusto está acompanhado de sua esposa, Rebeca, e de sua filha, Amanda. No segundo registro, Rejane está com seu esposo, Luiz Carlos Junior, e seus filhos, Levi e Matias. Dedicamos esta obra a eles.

Sumário

Capítulo 1: Conhecendo a robótica.....	15
Capítulo 2: Conhecendo o Arduino	19
Capítulo 3: Conhecendo os componentes.....	24
Placa de ensaio	24
LED.....	25
Resistores.....	27
Potenciômetro	28
Buzzer ou Piezo	30
Botão	31
Capacitor.....	32
Micro Servo (Servo Motor)	33
Motor de Engrenagem	34
LED RGB	35
Diferentes Tipos de Sensores	36
Interruptor Deslizante.....	36
Capítulo 4: E agora, por onde começar?	39
Ambientando-se.....	39
Utilizando a Ferramenta.....	41

Capítulo 5: Conhecendo a programação	47
Capítulo 6: Arduino IDE: Como fazer o código funcionar?	52
Capítulo 7: Atividades	59
ATIVIDADE 1: PISCAR LED	59
Componentes necessários:.....	59
Simulando no tinkercad:.....	59
Montando o circuito:.....	59
Montando o código:.....	60
EXPLORANDO A MATEMÁTICA NO ENSINO FUNDAMENTAL.....	63
EXPLORANDO A MATEMÁTICA NO ENSINO MÉDIO.....	64
ATIVIDADE 2: SEMÁFORO	65
Componentes necessários:.....	65
Simulando no tinkercad:.....	65
Montando o circuito:.....	65
Montando o código:.....	68
EXPLORANDO A MATEMÁTICA NO ENSINO FUNDAMENTAL.....	71
EXPLORANDO A MATEMÁTICA NO ENSINO MÉDIO.....	72
ATIVIDADE 3: CONTROLANDO A INTENSIDADE LUMINOSA.....	74

Componentes necessários:.....	74
Simulando no tinkercad:	74
Montando o circuito:	74
Montando o código:	77
EXPLORANDO A MATEMÁTICA NO ENSINO	
FUNDAMENTAL	80
EXPLORANDO A MATEMÁTICA NO ENSINO MÉDIO	81
ATIVIDADE 4: ALARME.....	83
Componentes necessários:.....	83
Simulando no tinkercad:	83
Montando o circuito:	83
Montando o código:	84
EXEMPLO 1: Observando o funcionamento do piezo (buzzer)	
e os diferentes tons gerados.....	87
EXEMPLO 2: Alarme de incêndio	88
EXEMPLO 3: Alarme de ambulância	89
EXEMPLO 4: Alarme com Seno completo	91
EXPLORANDO A MATEMÁTICA NO ENSINO	
FUNDAMENTAL	93
EXPLORANDO A MATEMÁTICA NO ENSINO MÉDIO	94
REFERÊNCIAS.....	96

Capítulo 1

Conhecendo a Robótica

A Robótica Educacional transforma a sala de aula em um espaço criativo e investigativo. Mais do que montar robôs, é uma ferramenta para desenvolver pensamento crítico, colaboração e resolver desafios reais com tecnologia. Desta forma, neste capítulo iremos compreender um pouco mais sobre a Robótica e suas aplicações.



Capítulo 1:

Conhecendo a robótica

Durante a implementação do trabalho com robótica no ambiente escolar, é natural surgirem diversas dúvidas por parte dos estudantes envolvidos no processo. Nesse sentido, é importante conversarmos sobre questões que envolvem robótica educacional e que frequentemente emergem no contexto da sala de aula. Os questionamentos, além de serem comuns entre os alunos, também podem refletir incertezas dos próprios professores que estão iniciando sua trajetória nesse campo. Assim, buscamos oferecer subsídios teóricos e práticos que contribuam para uma compreensão mais ampla e segura do uso da robótica como ferramenta pedagógica.

Segundo Campos (2019), um robô é uma máquina mecânica programável que executa tarefas de forma autônoma ou semiautônoma, ou seja, com pouca ou nenhuma intervenção humana. Esses dispositivos combinam componentes físicos, como sensores, motores e estrutura (hardware), com programação que controla suas ações (software). Essa combinação permite que eles interajam de maneira inteligente com o ambiente, realizando atividades que podem ser similares, complementares ou até substitutivas a algumas tarefas humanas. É importante destacar que um robô precisa existir fisicamente, diferentemente de programas de computador ou inteligência artificial que são puramente virtuais.

A robótica, por sua vez, é a área que se dedica ao estudo e construção desses robôs. As tecnologias relacionadas à robótica estão presentes de diversas formas no cotidiano, desde robôs aspiradores, que limpam residências de forma autônoma, até sistemas industriais mais complexos, como os braços robóticos utilizados em montadoras de carro responsáveis por tarefas como soldagem, pintura e montagem de alta precisão. Além disso, a robótica tem provocado impactos significativos na sociedade, promovendo inovação e impulsionando processos de reinvenção em diversos setores, como saúde, agricultura, logística e educação (Gomes et al., 2010).

No contexto educacional, a robótica emerge como uma possibilidade pedagógica inovadora, ao permitir que estudantes se envolvam com conhecimentos multidisciplinares dessa ciência, como programação, mecânica e eletrônica. Papert (1980) defende que, ao se envolverem com esses conhecimentos, os alunos desenvolvem competências fundamentais, como a resolução de problemas e o pensamento crítico, habilidades indispensáveis para a aprendizagem. Dessa forma, a robótica educacional vai além de desenvolver conhecimentos tecnológicos, ela propõe auxiliar os estudantes no desenvolvimento de habilidades relacionadas às disciplinas escolares.

Diante desse breve panorama sobre a robótica, podemos refletir sobre como a integração planejada e consciente dessa tecnologia em nossas práticas docentes pode ampliar as possibilidades de aprendizagem. Apesar de nova e desafiadora, quando bem articulada aos objetivos educacionais, a robótica não apenas pode tornar as aulas mais dinâmicas, mas criar espaços mais propícios para o

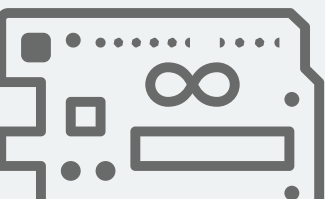
desenvolvimento integral dos estudantes, com práticas que aproximam conceitos abstratos a aplicações concretas, e desafios reais que estimulam a colaboração, criatividade e autonomia dos estudantes.



Capítulo 2

Conhecendo o Arduino

O Arduino é o cérebro por trás de projetos interativos. Acessível, intuitivo e de código aberto, ele conecta programação e eletrônica de forma prática, permitindo controlar sensores, motores e luzes com criatividade. Com uma vasta comunidade e infinitas possibilidades, é a porta de entrada ideal para aprender Robótica na escola. Neste capítulo, vamos conhecer um pouco mais sobre o Arduino e como utilizá-lo



Capítulo 2:

Conhecendo o Arduino

O [Arduino](#) é uma plataforma eletrônica de código aberto, que possui uma interface intuitiva e fácil de utilizar, baseada em hardware e software. Pode ser utilizada por qualquer pessoa que tenha interesse em fazer projetos interativos. Por ser uma plataforma de código e hardwares abertos, o Arduino se tornou uma excelente opção para quem deseja iniciar no ambiente de desenvolvimento. Trata-se de uma opção de valor financeiro acessível, que se mostra eficiente, capaz de facilitar processos difíceis como o uso de microprocessadores e microcontroladores.

O Arduino pode ser entendido como o cérebro de um robô ou de um projeto de eletrônica. No Arduino, você encontrará uma placa de circuito que permitirá a você criar os seus próprios programas. Basicamente, o Arduino funciona como uma comunicação entre o computador e os componentes. Assim sendo, podemos programar no computador e ligar os componentes necessários (como LEDs, buzzers, motores e sensores) no Arduino para fazer o que você aspira em seu circuito. Deste modo, por meio do código programado, você pode controlar os componentes.

A plataforma é utilizada em milhares de projetos de robótica, indo desde o uso em objetos cotidianos até instrumentos científicos complexos. O Arduino possui ainda uma comunidade mundial de criadores - estudantes, amadores, artistas, programadores e

profissionais - que se reúne em torno desta plataforma de código aberto, cujas contribuições se somam a uma quantidade incrível de conhecimento acessível que pode ser de grande ajuda dos novatos aos especialistas.

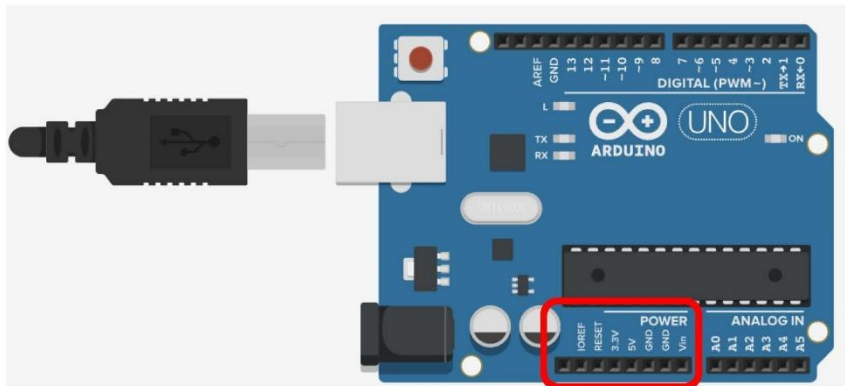
Por ser de código aberto, o Arduino possui ainda projetos abertos para o público, mantendo a tradição passada pelos idealizadores do projeto. Assim, se você deseja aprender ainda mais como utilizar o Arduino, você pode consultar a página oficial e procurar códigos diferentes para se inspirar. Sobretudo, nos tutoriais disponíveis no [Arduino Project Hub](#).

Neste material que elaboramos, você encontrará um resumo de como utilizar o Arduino com um enfoque na utilização em sala de aula. Esclarecemos que não temos a pretensão de abarcar todos os tópicos e componentes possíveis de se utilizar no Arduino. Abordamos ferramentas e especificidades pertinentes ao objetivo da dissertação de mestrado que este material integra. Nestes termos, passamos a apresentar a placa Arduino.

A placa Arduino Uno R3 é composta por 32 portas, tendo cada uma a sua função. Além disso, a placa possui uma fonte de alimentação via USB e uma fonte de alimentação externa em que se recomenda uma alimentação entre 7 e 12 volts.



O Arduino conta com portas, em sua parte inferior, as quais são consideradas conectores de alimentação que serão utilizados para alimentar e aterrar os componentes.



Conectores de alimentação

Dessa forma, o pino 3.3 V e o pino 5V fornecem ao componente a referida tensão para alimentação de circuitos externos, enquanto os pinos GND são pinos de aterramento.

O Arduino conta ainda com mais 14 entradas/saídas digitais e 6 entradas/saídas analógicas, além de mais um pino de aterramento (GND). Essas entradas operam com uma alimentação interna de 5 volts, além de um resistor de pull-up interno que pode ser habilitado por software.



São nessas 20 portas que ocorre a maior parte do seu trabalho nos circuitos do Arduino. É muito importante, portanto, que você saiba diferenciar as portas digitais das portas analógicas.

As **portas digitais** são mais simples e trabalham apenas com dois estados de funcionamento (ligado e desligado), podendo incluir componentes eletrônicos como LEDs e botões, sensores de presença, buzzer, LEDs RGB, displays, sensores de distância, sensores de toque e servo motor.

Já as **portas analógicas** possuem uma faixa contínua de valores a serem trabalhados, permitindo detectar diferentes intensidades e valores de funcionamento. Neste caso, são componentes que possuem uma variação gradual, como para controlar a intensidade luminosa. São exemplos de componentes analógicos: potenciômetros, sensores de luz, sensores de temperatura, potenciômetros deslizantes, joysticks, microfones, controle de volume de áudio, dentre outros.

Capítulo 3

Conhecendo os Componentes

Antes de colocar a mão na massa com o Arduino, é essencial entender os principais elementos que dão vida aos nossos projetos. Neste capítulo, exploramos os componentes básicos — como protoboards, LEDs, resistores e fios — revelando suas funções e curiosidades. Com esse conhecimento, você estará pronto para montar circuitos de forma segura, prática e criativa.



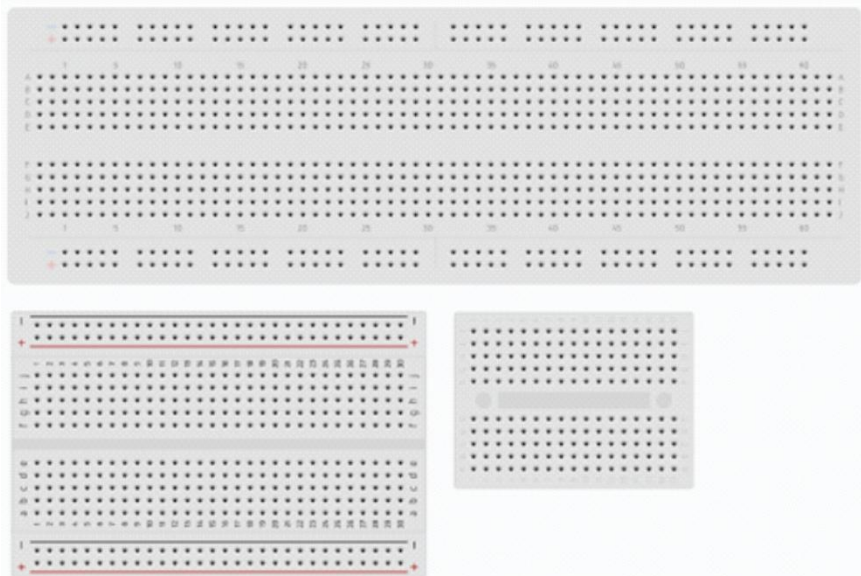
Capítulo 3:

Conhecendo os componentes

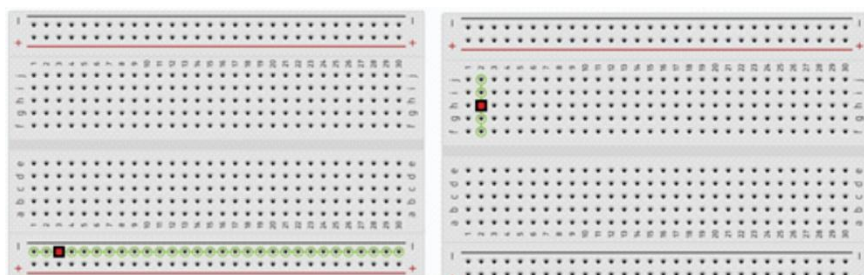
Para trabalharmos com o Arduino, precisamos conhecer um pouco mais sobre os componentes que serão utilizados. Vamos nos ater aos componentes considerados mais importantes e usuais para projetos em robótica educacional, além de nos aprofundarmos nos componentes que serão utilizados nos projetos.

Placa de ensaio

A placa de ensaio, também conhecida como *protoboard* ou *breadboard*, é uma ferramenta muito útil na robótica, ela é utilizada para ligarmos os componentes e pode apresentar diferentes tamanhos.



As placas de ensaio apresentam trilhas (linhas) na horizontal e vertical, de forma que a conexão e alimentação em uma trilha irão alimentar todos os pinos nela presentes. As colunas internas (enumeradas de um a 30 no exemplo) são trilhas verticais, enquanto as linhas enumeradas com os símbolos + e - são trilhas horizontais e devem ser utilizadas para aterrar todos os componentes (que necessitam de aterramento) ou alimentar vários componentes que necessitem de alimentação em um mesmo ponto na placa Arduino. Na nossa proposta, utilizaremos somente o polo negativo da placa de ensaio para fazer aterramentos em um mesmo ponto do Arduino, visto que os componentes que serão trabalhados devem ser aterrados na porta GND.



As placas de ensaio, portanto, são consideradas uma matriz de orifícios que servem para organizar melhor os componentes que serão utilizados e facilitar a experimentação e aprendizado, visto que facilitam diversificar e testar componentes sem danificá-los.

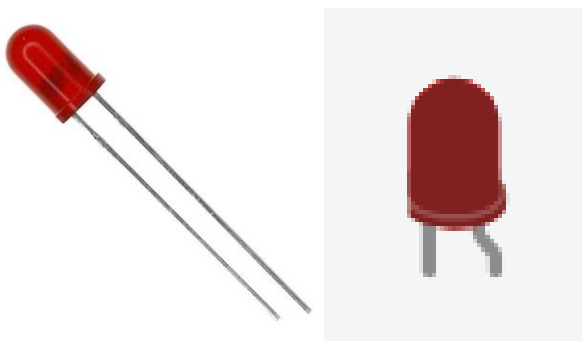
LED

Os LEDs são diodos emissores de luz que desempenham essa função quando em contato com uma corrente elétrica. São

componentes baratos, eficientes e duráveis, que consomem pouca energia, ao trabalharem em uma voltagem de 5v e com baixa potência. Assim, geram pouco calor quando comparados com outras fontes de luz, sendo excelentes componentes na hora de começar a trabalhar com a robótica e para fazer diferentes testes antes de utilizar componentes mais caros.

Os LEDs podem ser utilizados para comunicação visual, indicadores de status, projetos que envolvam iluminação, detecção e sensoramento, decoração de projetos, entre outros. Os LEDs possuem duas pernas, uma maior chamada de ânodo, e outra menor chamada de cátodo.

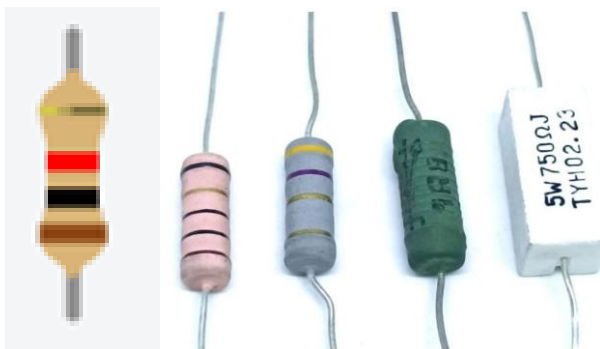
O ânodo é o terminal positivo (+) do dispositivo eletrônico. É por ele que a corrente elétrica entra e, por isso, deve estar conectado ao terminal positivo da fonte de alimentação. Já o cátodo é o terminal negativo (-) do dispositivo eletrônico, por onde a corrente elétrica sai e, por isso, deve ser conectado ao terminal negativo da fonte de alimentação.



Resistores

Os resistores são componentes que controlam o fluxo de corrente elétrica, eles fazem isso oferecendo uma resistência ao fluxo, podemos imaginar como se fossem barreiras que controlam o tráfego. Para elucidar, imagine que você está em uma rodovia que possui 8 pistas para o tráfego de carro com uma movimentação muito grande. Para controlar o tráfego, o governo constrói um posto policial que permite a passagem somente de um carro por vez, fazendo com que a polícia consiga visualizar melhor os carros que estão passando na pista. De maneira análoga, ocorre na robótica, e esse redutor de fluxo para uma pista é o resistor.

Os resistores, portanto, controlam a corrente elétrica em um circuito, fazendo com que os componentes do circuito não queimem por sobrecarga elétrica, protegendo assim o seu circuito eletrônico.



Vale ressaltar que existem diferentes tipos de resistores, que irão oferecer diferentes tipos de resistência. Cada componente possui a necessidade de um diferente tipo de resistor. Ao longo das atividades,

vamos elucidar como verificar a resistência dos resistores que utilizaremos.

Potenciômetro

O potenciômetro é um componente eletrônico analógico que irá controlar manualmente a resistência em um circuito. Ele será muito útil quando aliado a outro componente. Podemos pensar em um potenciômetro como um botão de volume em um dispositivo de som e, ao girar o botão, ajustamos a resistência e, conseqüentemente, ajustamos o volume do som.

O potenciômetro precisa de uma alimentação própria para funcionar e, por isso, ele possui três terminais, sendo o primeiro terminal que deverá ser ligado a uma fonte de alimentação (no caso do Arduino, iremos ligar no terminal de 5 volts nos conectores de alimentação), o último terminal será um terminal de aterramento, já o terminal que se encontra no meio será um terminal frequentemente chamado de “limpador” ou “cursor” que irá retornar para a placa Arduino a quantidade de resistência gerada.

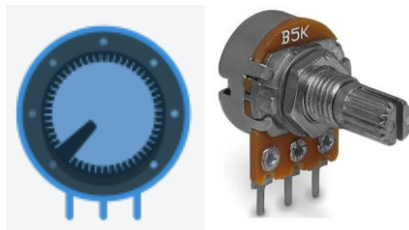
Existem duas formas de utilizar o potenciômetro no Arduino. Como vimos anteriormente, a placa Arduino fornece um resistor de pull up e, por isso, você pode controlar a resistência que irá passar no componente a partir do retorno enviado para a placa Arduino pelo terminal de limpador. Outra forma de utilizar o potenciômetro é conectar diretamente ao terminal do limpador o componente, lembrando que será ainda assim necessário o uso de uma resistência no componente para que ele não queime, visto que, quando o

potenciômetro se encontra no mínimo, ele pode não oferecer resistência ao componente, fazendo com que ele queime.

Existem diferentes tipos de potenciômetros, como:

- **Potenciômetro Linear:** É um dos modelos de componente mais utilizado. Nele, a resistência varia de maneira linear conforme você gira o botão. Isso significa que a mudança de resistência é constante ao longo de todo o movimento do botão.
- **Potenciômetro Logarítmico (ou Áudio):** A resistência, neste tipo de potenciômetro, varia de maneira logarítmica. Eles são frequentemente utilizados em controles de volume de áudio porque correspondem melhor à percepção humana de mudanças de volume.
- **Potenciômetro Rotativo:** Este é o tipo mais comum de potenciômetro e funciona girando um eixo para ajustar a resistência. Eles podem ser lineares ou logarítmicos.
- **Potenciômetro Deslizante (ou Fader):** Em vez de girar, você move um cursor ao longo de uma pista reta. Eles são frequentemente utilizados em mesas de som e equipamentos de mixagem de áudio.

Existem ainda outros tipos de resistores utilizados em projetos mais avançados para obter uma maior precisão no circuito. Em nossos projetos iremos utilizar potenciômetros do tipo rotativo linear que, como citado, são os mais utilizados.

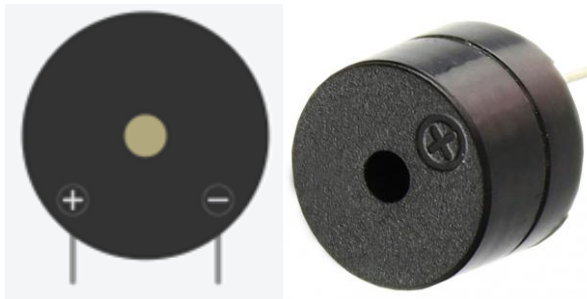


Buzzer ou Piezo

É um pequeno componente capaz de gerar diferentes tipos de sons, geralmente tem um formato circular, ao possuir um pequeno disco laminar que vibra dentro dele para emitir som quando recebe energia elétrica.

Os buzzers são muito utilizados para criação de alarmes sonoros, brinquedos, campainhas, notificações, entre outros. Eles conseguem gerar diferentes tonalidades de sons que podem ser controladas via frequência enviada por comando para o componente.

O componente é simples, possuindo apenas dois terminais, um positivo (+) que deve ser alimentado e um negativo (-) que deve ser aterrado.

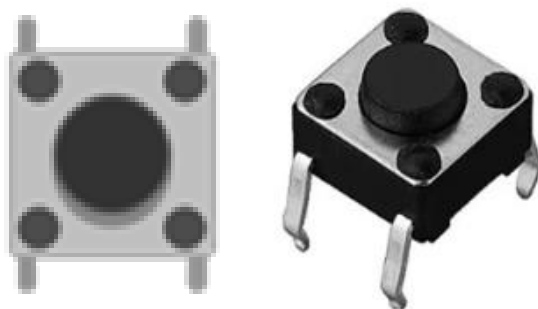


Existem ainda outros componentes utilizados que não irão compor o escopo de exercícios contidos neste material, ainda assim cabe falarmos um pouco mais sobre eles aqui para que você possa, caso queira futuramente, explorar diferentes tipos de possibilidades com seu aluno em sala de aula:

Botão

O botão é um componente eletrônico simples que atua como um interruptor momentâneo em um circuito elétrico. Sua principal função é permitir ou interromper a passagem de corrente elétrica quando pressionado. No botão, há pequenos contatos metálicos que se conectam ao serem pressionados, fechando o circuito e permitindo a passagem da corrente elétrica. Quando o botão é solto, esses contatos se separam, interrompendo o fluxo de eletricidade.

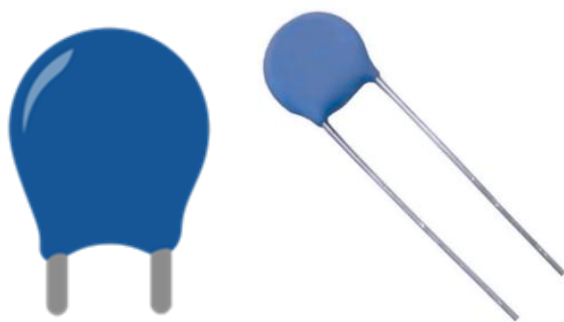
Para utilizar um botão em um circuito com um microcontrolador, como o Arduino, a conexão é feita ligando um dos lados do botão à fonte de alimentação. O outro lado é conectado a um pino de entrada do microcontrolador e, simultaneamente, ao aterramento, por meio de um resistor de valor elevado, que mantém o pino em estado BAIXO (LOW) enquanto o botão não é pressionado. Quando o botão é pressionado, o pino se conecta à alimentação, registrando um sinal ALTO (HIGH).



Nos simuladores, como o Tinkercad, é possível testar o funcionamento do botão clicando sobre ele para simular sua ativação. Para manter o botão pressionado no ambiente virtual, basta segurar a tecla Shift e clicar no botão.

Capacitor

O capacitor é um componente eletrônico utilizado para armazenar temporariamente energia elétrica. Ele é composto por duas placas condutoras separadas por um material isolante, chamado dielétrico. Quando conectado a uma fonte de energia, o capacitor acumula cargas elétricas em suas placas. Quando necessário, essa energia pode ser liberada rapidamente para alimentar um circuito ou estabilizar variações de tensão.



Os capacitores possuem dois terminais e podem ser conectados em série ou em paralelo com outros componentes, dependendo da aplicação desejada. No caso dos capacitores cerâmicos, eles não possuem polaridade, o que significa que podem ser conectados em qualquer posição sem afetar seu funcionamento. Devido à sua

capacidade de armazenar e liberar energia rapidamente, os capacitores são frequentemente utilizados para estabilizar sinais elétricos, armazenar e liberar energia, eliminar ruídos em circuitos e até mesmo ajudar no funcionamento de motores elétricos.

Micro Servo (Servo Motor)

O micro servo, também conhecido como servo motor, é um dispositivo compacto que combina um motor de corrente contínua (CC), uma placa controladora e uma caixa de engrenagens. Ele é utilizado para movimentar algo de forma precisa e controlada.

O funcionamento do servomotor é baseado em sinais de pulso modulado por largura de pulso (PWM). Esses sinais indicam a posição desejada que o motor deve atingir. O motor gira para essa posição específica, de acordo com a duração do pulso que recebe.

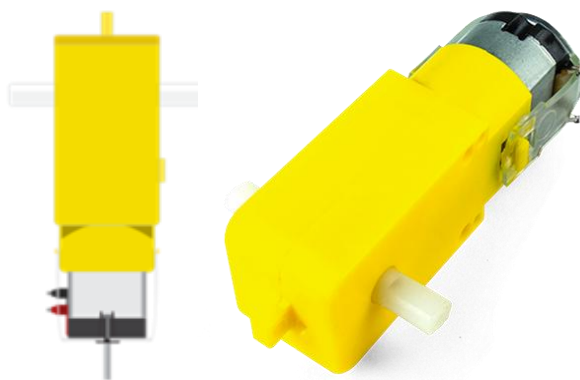
O servo motor possui três fios: o fio de alimentação (Power), que deve ser ligado ao pino 5V do Arduino; o fio de terra (Ground), que deve ser conectado a um pino GND do Arduino; e o fio de sinal (Signal), que deve ser ligado a um pino de saída PWM no Arduino.



Após realizar as conexões, é necessário programar o Arduino para enviar sinais PWM ao pino de sinal do servo, o que pode ser feito utilizando uma biblioteca “Servo”, por exemplo. Assim, é possível controlar o movimento do servomotor, permitindo que ele gire para a posição desejada.

Motor de Engrenagem

O motor de engrenagem é um componente em projetos de robótica, especialmente quando se deseja movimentar partes do robô, como rodas ou braços mecânicos. Ele combina um motor elétrico de corrente contínua (motor CC) com uma caixa de engrenagens.

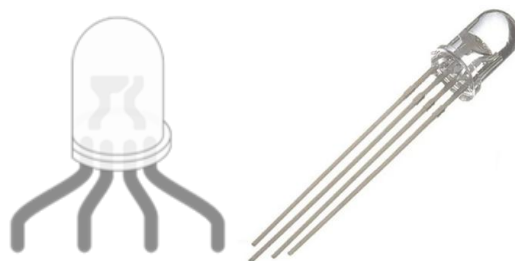


Quando o motor CC é alimentado por corrente contínua, é criado um campo magnético que atrai e repele os ímãs no interior, fazendo o motor girar. A caixa de engrenagens é responsável por reduzir a velocidade de rotação do motor e aumentar seu torque, tornando o movimento mais forte e controlado, ideal para movimentar rodas e robôs.

Para utilizá-lo, basta conectar seus dois fios condutores à fonte de energia: o fio positivo deve ser ligado à alimentação e o fio negativo ao aterramento. Dessa forma, o motor começará a girar. Invertendo a polaridade, o sentido de rotação também se inverte.

LED RGB

O LED RGB é um componente formado por três LEDs (vermelho, verde e azul) encapsulados em um único invólucro. Combinando diferentes intensidades das cores vermelho, verde e azul, o LED RGB possibilita gerar praticamente qualquer cor de luz visível.



Esse dispositivo possui quatro pinos condutores: um cátodo comum que deve ser ligado ao GND, e três pinos correspondentes às cores, que devem ser conectados a uma tensão mais alta para acender cada LED. Além disso, para o funcionamento correto, é necessário utilizar resistores em série com os pinos de cada cor, para evitar danos por excesso de corrente elétrica.

Diferentes Tipos de Sensores

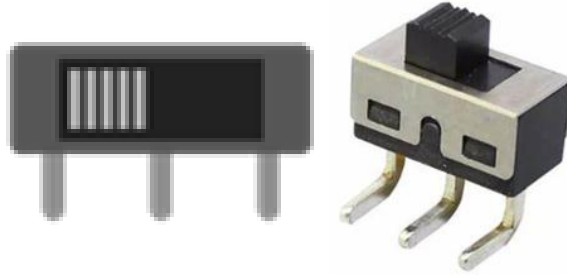
Os sensores são importantes dispositivos de robótica, que possibilitam interações entre o ambiente real e virtual. Eles funcionam captando informações do meio, como luz, distância, temperatura ou som, e as transformam em sinais elétricos que podem ser interpretados por um sistema.

Existem diversos tipos de sensores que podem ser utilizados em projetos educativos, como sensores de luz (que detectam variações de luminosidade), sensores ultrassônicos (que medem distância por meio de ondas sonoras), sensores de temperatura, de toque, entre outros. Cada tipo é escolhido de acordo com a necessidade do projeto.

A conexão dos sensores é geralmente feita com três fios: alimentação (VCC), aterramento (GND) e sinal (Signal). O fio de sinal envia os dados para o microcontrolador, como o Arduino, que então pode tomar decisões com base nessas informações, por exemplo, desviar de obstáculos, seguir uma linha no chão ou acender um LED quando escurece.

Interruptor Deslizante

O interruptor deslizante funciona como um interruptor de alternância em duas posições fixas, permitindo ligar ou desligar circuitos manualmente. Quando o controle deslizante é movido, contatos metálicos internos se unem, permitindo a passagem de corrente elétrica.



Esse tipo de interruptor possui três pinos condutores e, para utilizá-lo, é necessário ligar o pino do meio (Common) à entrada do componente que será controlado (como um motor ou LED). Em seguida, conectar um dos pinos laterais ao positivo da alimentação (Power) e o outro ao negativo ou GND (Ground). A ordem de conexão desses dois pinos define em qual posição o dispositivo estará ligado ou desligado, dependendo da direção do movimento do controle deslizante.

Capítulo 4

E agora, por onde começar?

Antes de usar o Arduino físico, que tal conhecer o Tinkercad? Uma plataforma online gratuita onde você pode simular circuitos, aprender eletrônica, programar com segurança e explorar as ferramentas essenciais da Robótica Educacional de forma prática e acessível.



Capítulo 4:

E agora, por onde começar?

Ambientando-se

Antes de começar de fato a trabalhar com o Arduino, é importante que você se familiarize com a eletrônica. Para isso, uma empresa chamada Autodesk, criada pelo ex-engenheiro do Google Kai Beckman e pelo co-fundador Mikko Mononen, fundou uma plataforma online gratuita para que você possa fazer diferentes tipos de projetos antes de utilizar o Arduino físico, o Tinkercad.

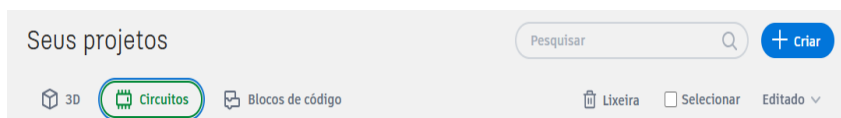
Para utilizar o Tinkercad, é fácil. Primeiramente, precisaremos criar uma conta no site. Para isso, acesse o site <https://www.tinkercad.com/> e clique em inscrever-se no canto superior direito da tela.

Existem três tipos diferentes de conta: a conta de educadores, para aqueles que queiram compartilhar materiais com os alunos em um modelo de “sala de aula”; as contas de alunos, para aqueles que desejam acessar os conteúdos disponibilizados pelos professores; e a conta pessoal, que permite você gerir o seu acervo pessoal de maneira autônoma.

Agora, iremos explorar somente as contas pessoais. Para isso, basta clicar em criar uma conta pessoal, posteriormente escolher a opção entrar com e-mail, colocar o país, data de nascimento, e-mail e senha.

Agora conclua as etapas de cadastro e faça o login no site. Agora, você encontrará a página inicial do Tinkercad com algumas opções. Entre elas, você encontrará um menu localizado à esquerda da tela com algumas opções, dentre as quais iniciaremos explorando a opção projetos (selecione-a no menu).

Na opção projetos, você poderá administrar os projetos em que você tem trabalhado. Cabe ressaltar que o Tinkercad possui projetos não somente na área da robótica, mas também na área de criação de objetos tridimensionais e na área de programação em blocos. Neste trabalho, iremos nos atentar somente à parte de circuitos (selecione esta opção).



 **Início**

 **Classes**

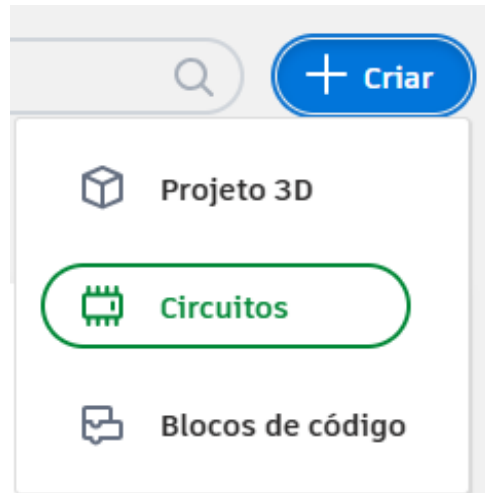
 **Projetos**

 **Coleções**

 **Tutoriais**

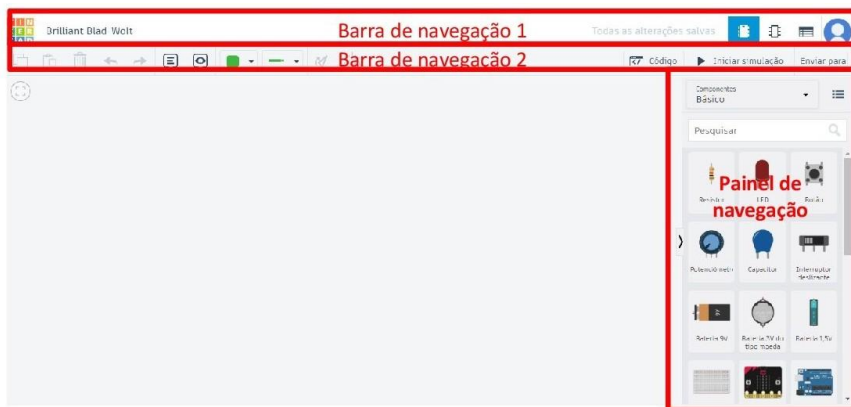
 **Desafios**

Para criar um novo projeto, clique em + criar → circuitos. Agora você está pronto para começar a trabalhar com a robótica.



Utilizando a Ferramenta

Ao abrir a sua página inicial da montagem do circuito, você encontrará duas barras de navegação superiores e um painel de navegação localizado à direita.



A barra de navegação 1 apresenta cinco opções:



- **Opção 1:** representada através do logo do Tinkercad, é responsável por voltar ao menu de controle de projetos pessoais criados.
- **Opção 2:** responsável pela nomeação do arquivo de seu projeto. A ferramenta do Tinkercad apresenta ao desenvolvedor um nome aleatório gerado automaticamente toda vez que um novo projeto é criado, cabe ao usuário renomear o projeto. Para fazer isso, basta clicar em cima do nome gerado e renomear o projeto de acordo com o que você tem feito.
- **Opção 3:** apresenta a visualização do circuito em que você tem trabalhado.
- **Opção 4:** apresenta a visualização esquemática do projeto (opção que será explorada por desenvolvedores experientes em hardware e software).
- **Opção 5:** apresenta uma lista de componentes utilizados durante a realização do projeto.

A barra de navegação 2 apresenta 14 opções:



- **Opção 1:** utilizada para copiar um ou mais componentes selecionados, podendo também ser utilizado o atalho CTRL+C.
- **Opção 2:** utilizada para colar os componentes selecionados, podendo também ser utilizado o atalho CTRL+V.
- **Opção 3:** utilizada para excluir um componente selecionado, podendo ser utilizada também a tecla “delete” do teclado.
- **Opção 4:** desfaz a sua última ação, podendo também ser utilizado o atalho CTRL+Z.
- **Opção 5:** refaz a sua última ação desfeita, podendo também ser utilizado o atalho CTRL+Y.
- **Opção 6:** cria anotações para lembrar de algo para a posteridade, ou para compartilhar alguma informação com a pessoa que for utilizar o seu circuito.
- **Opção 7:** esconde ou mostra as notas do desenvolvedor.
- **Opção 8:** modifica a cor do fio selecionado.
- **Opção 9:** modifica o tipo de fio utilizado.
- **Opção 10:** rotaciona o componente selecionado, podendo também ser utilizada a tecla R do teclado. Essa opção irá te auxiliar algumas vezes a manter o circuito mais organizado.
- **Opção 11:** espelha o componente.
- **Opção 12:** irá mostrar o código utilizado na programação do circuito. É aqui que iremos trabalhar com os códigos do Arduino para conseguirmos fazer diferentes tipos de circuitos funcionarem adequadamente. Ao selecionarmos essa opção, podemos ver que existem três tipos de modos de edição. Isso se deve ao fato de que o Tinkercad possui a opção de trabalhar em blocos. Em nosso projeto,

iremos trabalhar somente com texto, assim após adicionarmos os componentes iremos precisar selecionar a opção texto presente no modo de edição.

- **Opção 13:** inicia a simulação do código escrito, é neste momento que verificaremos se existe algum erro no código escrito.
- **Opção 14:** será utilizada para compartilhar o seu projeto com outras pessoas, exportá-lo ou salvar a imagem do projeto.

O painel de navegação lateral apresenta 4 opções:



- **Opção 1:** utilizada para pesquisar os tipos de componentes. Em nosso trabalho, vamos manter a opção de trabalharmos somente com componentes básicos.
- **Opção 2:** alterna o modo que serão visualizados os componentes, alternando entre uma visualização compacta (padrão) e uma visualização detalhada (que irá apresentar detalhes relevantes sobre cada componente).
- **Opção 3:** utilizada para pesquisar algum componente pelo nome.
- **Opção 4:** utilizada para selecionar o componente desejado clicando em cima dele e arrastando o mouse para colocá-lo na tela.

Outro ponto relevante é o fato de que, ao colocarmos um componente na tela, ele irá abrir uma janela com as opções do componente. Por padrão, todo componente pode ser renomeado. Em alguns casos, teremos outras opções, como alteração de cor, para o caso dos LEDs, e alteração da resistência, para resistores.

Além disso, ao clicarmos na interrogação localizada no canto superior direito da janela aberta junto ao componente, poderemos ver uma documentação do componente com uma breve descrição sobre ele, funcionamento, conexão, utilização, uma introdução ao componente e mais informações. Essa ferramenta será muito útil para que você possa continuar os seus estudos acerca da robótica educacional futuramente.

Capítulo 5

Conhecendo a Programação

A programação é o coração da Robótica: é por meio dela que os circuitos ganham vida. Neste capítulo, você vai conhecer os principais comandos do Arduino, aprender a consultar a documentação oficial e entender como cada instrução funciona nos projetos. Um guia prático para programar com autonomia e confiança.



Capítulo 5:

Conhecendo a programação

O principal motor da robótica é a programação, é a partir de códigos que todo o funcionamento do circuito ocorre. Posteriormente, neste trabalho, iremos apresentar a utilização destes códigos para realização de diferentes atividades. Nesta seção, você poderá voltar para entender melhor o que cada comando faz no seu projeto.

Vale ressaltar que, por ser uma plataforma de código aberto, o Arduino apresenta uma excelente documentação disponível em seu site. Para consultarmos a documentação do comando e um exemplo de sua utilização, basta pesquisarmos no Google o comando desejado acrescido do termo “Arduino documentation”. Geralmente, isso é suficiente para que o primeiro link não patrocinado que apareça seja a documentação do comando disponível no site do Arduino. Caso ainda tenha dúvida se essa é a documentação oficial, basta verificar se o site que a disponibiliza é o “docs.Arduino.cc”.

Observe também que a documentação apresentada está em inglês. Uma dica para aqueles que não dominam o inglês é que, ao clicarmos com o botão direito do mouse sobre a tela do site aberto, encontraremos uma opção “traduzir para o português”.

Ressaltamos que os comandos, que estão explicados neste material, são aqueles que compõem as atividades propostas.

→ **pinMode(Nº ou LED_BUILTIN, OUTPUT ou INPUT);** : Localizada no setup do código, deverá ser utilizada para

configurar a porta (a numeração deverá ser colocada antes da vírgula) como entrada ou saída. Caso você queira utilizar o LED existente na placa Arduino, você pode utilizar o comando *LED_BUILTIN* ao invés do número da porta de entrada. Além disso, temos a opção *INPUT* (geralmente utilizada quando há coleta de dados advindas do componente) ou *OUTPUT*, que irá indicar o tipo de elemento que será conectado à porta, como entrada ou saída. No nosso caso, utilizaremos a opção *OUTPUT* (que deverá ser colocada após a vírgula) para configurarmos a porta como saída, visto que não haverá coleta de informações advindas do componente.

→ **digitalWrite (Nº, HIGH ou LOW);** : Localizada no loop do código, configura a tensão que a porta irá gerar, de forma que a posição HIGH irá receber energia de 5V, enquanto a opção LOW não irá receber energia.

→ **Delay (Nº);** : Gera um tempo de espera no código de acordo com o valor informado em milissegundos.

→ **Serial.begin (9600);** : Configura a taxa de transmissão de bits do componente. Como padrão, vamos utilizar 9600. Devemos ter cuidado, pois ao iniciarmos o monitor serial (para verificarmos os valores recebidos pelo código), isso irá alterar a taxa de bits recebidas pelo piezo (buzzer), o que afetará o seu desempenho e o som gerado por ele.

→ **int VARIÁVEL;** : Declaração de variáveis do tipo inteiro que será utilizada para armazenar informações quando necessárias.

→ **analogRead (PORTA);** : Utilizada para leitura da porta, no nosso caso, no potenciômetro.

➔ **map (valor, deMenor, deMaior, paraMenor, paraMaior); :**

Transforma um valor que varia entre o valor deMenor até o valor deMaior, para um valor que varia de paraMenor até paraMaior. Ele é utilizado para remapear um valor, muitas vezes o valor aceito por um componente não será o mesmo que será gerado pelo outro componente, por isso devemos remapear os valores antes de utilizar (Obs.: O potenciômetro gera valores entre 0 e 1023, enquanto as lâmpadas de LED irão aceitar melhor valores entre 0 e 255).

➔ **analogWrite (Nº, VALOR); :** Análogo ao digitalWrite e

localizado no loop do código, irá fazer com que a porta de numeração desejada receba uma tensão. Porém, dessa vez, o valor irá variar de acordo com o *VALOR* passado.

➔ **Serial.println (VARIÁVEL); :** Irá escrever na tela do

computador o valor lido pela variável para ser feito um melhor controle.

➔ **for (inicialização; condição; incremento) { } :** O laço *for* é

um dos comandos de repetição mais importantes das linguagens de programação, podendo apresentar diferentes sintaxes. No Arduino, a sintaxe utilizada será esta apresentada, onde na inicialização declaramos uma variável do tipo inteiro que apresentará o primeiro valor a ser executado; no segundo momento apresentaremos a condição de parada do laço de repetição, e em terceiro o incremento em que cada execução fará na variável de inicialização, assim o comando `for (int x=0;x<180;x+1){}` irá fazer uma repetição de um em um (x+1), iniciando pelo valor de zero (int x=0) até que atinja um valor de 179

($x < 180$). Vale ressaltar que o valor de incremento $x+1$ deve ser substituído por $x++$.

→ **tone (Nº; FREQUÊNCIA); :** o comando `tone` fará com que o piezo (buzzer) ligado à porta de entrada do comando execute um tom relacionado à frequência desejada.

→ **Declaração de variáveis globais e locais:** Dependendo do local onde a variável será declarada, ela poderá ser observada em todo o código (chamada de variável global) ou somente em parte do código (chamada de variável local). Para tornar mais simples a mudança dos códigos, vamos sugerir que algumas variáveis sejam declaradas de maneira global. Assim, ao alterarmos os valores das variáveis globais no código, os valores serão alterados em todos os códigos, não precisando ser modificados em cada parte dele.

Capítulo 6

Arduino IDE: Como fazer o código funcionar?

Antes de usar o Arduino, é preciso instalar e configurar a IDE. Aqui você aprenderá a fazer isso, escolher a placa correta (Arduino Uno) e montar seu primeiro projeto: fazer um LED piscar com comandos básicos como `pinMode`, `digitalWrite` e `delay`.

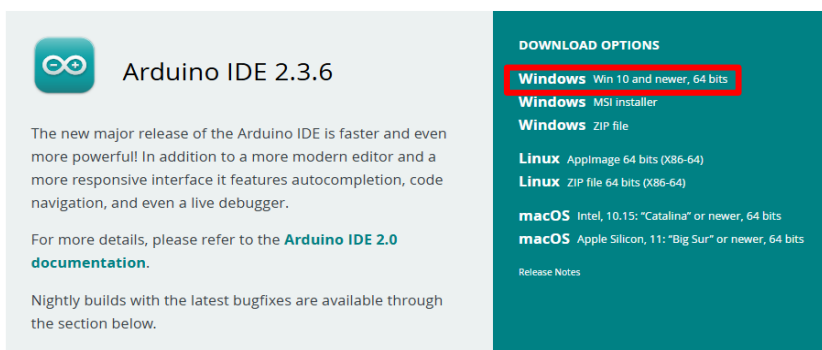


Capítulo 6:

Arduino IDE: Como fazer o código funcionar?

Para o código funcionar no Arduino, precisamos instalar em nosso computador um ambiente que fará a ligação entre o computador e a placa de Arduino. Para isso, iremos fazer o download do Ambiente de Desenvolvimento Integrado do Arduino, chamado Arduino IDE. Para isso, acesse o site: <https://www.Arduino.cc/en/software/>.

No menu à direita, escolha a melhor opção para download de acordo com o sistema operacional da máquina que você está utilizando. Se você estiver utilizando o Windows 10 ou superior, basta clicar na primeira opção:



 **Arduino IDE 2.3.6**

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

DOWNLOAD OPTIONS

- Windows** Win 10 and newer, 64 bits
- Windows** MSI installer
- Windows** ZIP file
- Linux** AppImage 64 bits (X86-64)
- Linux** ZIP file 64 bits (X86-64)
- macOS** Intel, 10.15: "Catalina" or newer, 64 bits
- macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

Caso queira fazer alguma doação para o projeto do Arduino, agora existe uma opção para ser feita a doação, caso contrário, basta clicar na opção Just Download para baixar o arquivo:

\$3	\$5	\$10	\$25	\$50	Other
-----	-----	------	------	------	-------

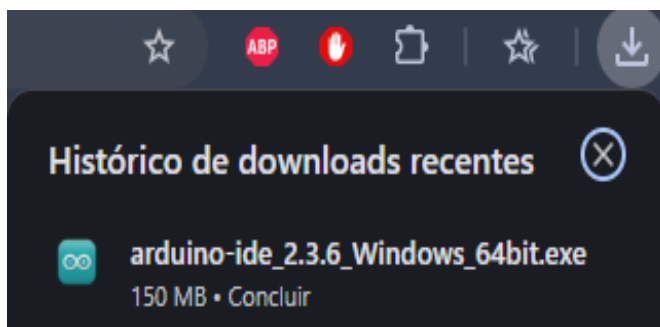
CONTRIBUTE AND DOWNLOAD

or

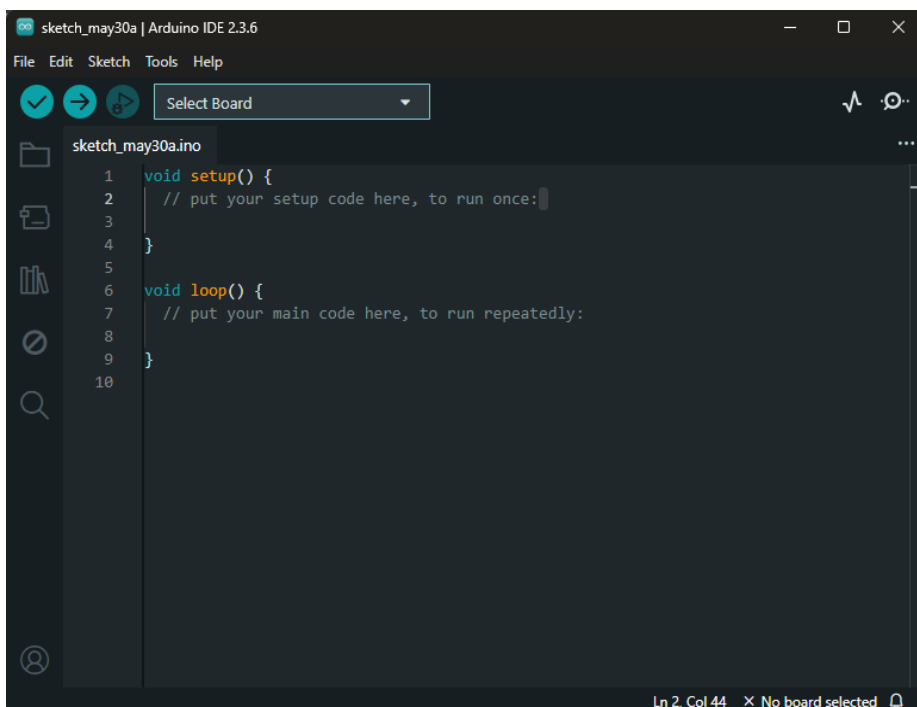
JUST DOWNLOAD



Novamente, o site te dá a opção de se cadastrar para receber notícias ou não. Caso não tenha interesse em receber notícias, basta clicar em Just Download novamente que o arquivo será baixado automaticamente para o seu computador.

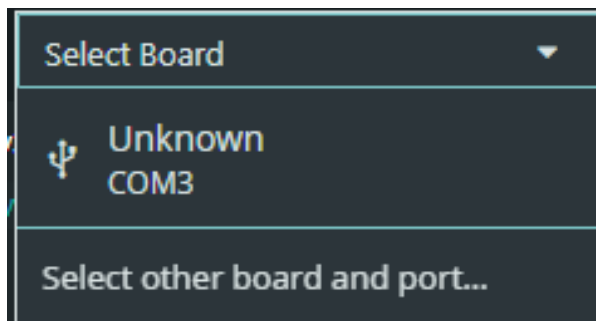


Clique no link baixado e siga o passo a passo para terminar de instalar o Arduino IDE. Para este trabalho, utilizamos o Arduino-ide v2.3.6. Após a instalação, ao executar o Arduino IDE, você irá se deparar com a seguinte tela:

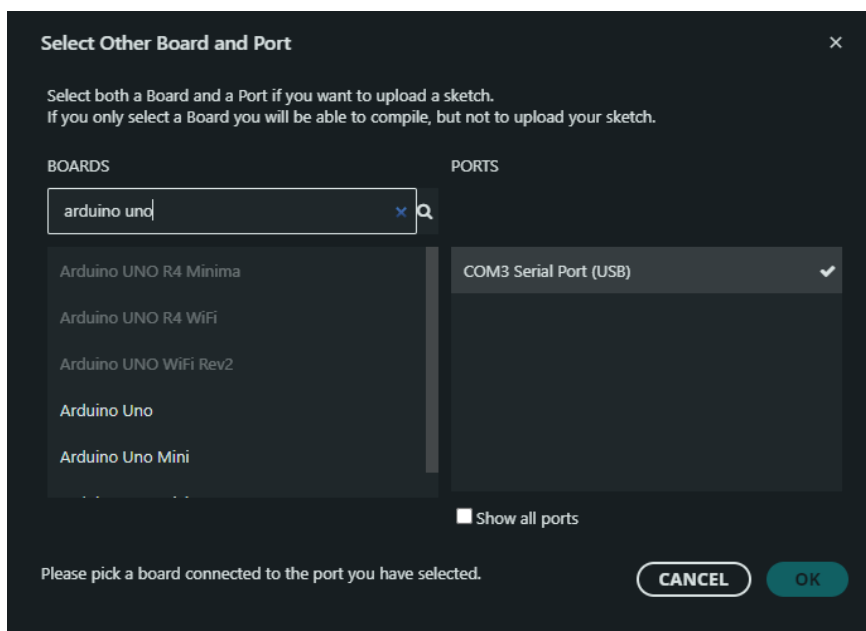


No menu superior, você encontrará cinco opções. Toda vez que você for trabalhar com o Arduino, você deverá demonstrar ao aplicativo qual placa você está utilizando. Assim, ao conectar o Arduino ao computador, você deverá selecionar a opção Arduino Uno para a IDE entender com qual componente você está trabalhando.

Para isso, clique em select board e depois em unknown COM3:



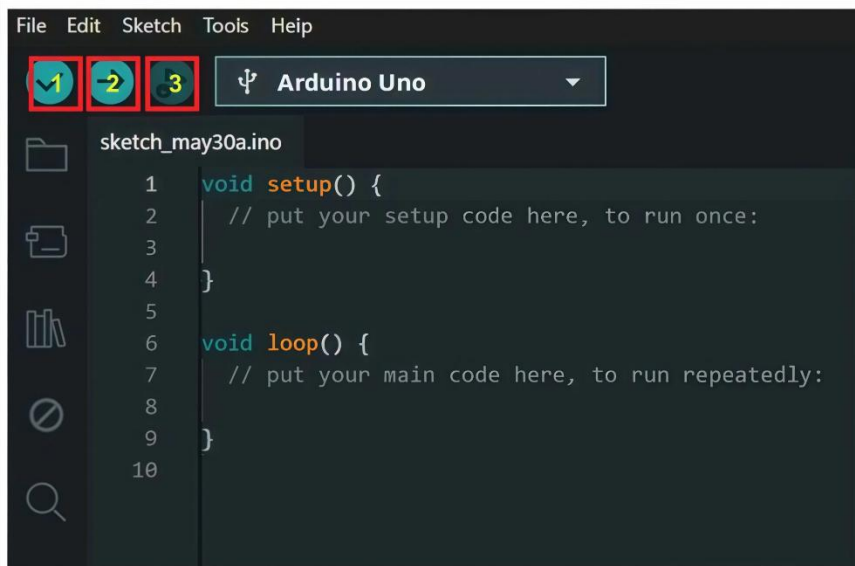
Na nova janela aberta, clique em boards e pesquise por Arduino Uno:



Selecione a opção Arduino Uno (obs.: futuramente, caso você adquira uma placa de Arduino, é importante se atentar a qual placa você

está utilizando, no nosso caso utilizaremos somente a placa Arduino Uno). Após selecionado, clique em OK e sua placa estará selecionada.

Com a placa selecionada, é importante entendermos melhor um pouco sobre as opções do Arduino IDE.



A **Opção 1** será o local onde você poderá verificar o código, observando se o código possui erros antes de subir o código para a placa Arduino.

A **Opção 2** permitirá que você faça o upload do código para a placa Arduino.

A **Opção 3** não é suportada pelo Arduino Uno, então ela permanecerá inativa enquanto você estiver utilizando tal placa.

As opções do menu lateral não serão utilizadas agora.

Agora, para você conseguir levar o código feito no Tinkercad ao Arduino para vê-lo funcionando, primeiramente você deverá

montar o sistema. Após montado o sistema, copie o código feito no Tinkercad e cole no menu principal.

Verifique se o código possui erro, caso não encontre nenhum erro, clique na opção de upload com a placa conectada no computador, dessa maneira, caso não exista nenhum erro de ligação ou código, o seu código irá funcionar.

Capítulo 7

Atividades

Após a apresentação dos fundamentos teóricos sobre a utilização do Arduino e do Tinkercad, este capítulo propõe um conjunto de atividades práticas nessas plataformas e com componentes físicos, além de sugestões de atividades didáticas que podem contribuir para o trabalho com o conteúdo de funções no ensino de Matemática.



Capítulo 7

Atividades

ATIVIDADE 1: PISCAR LED

Componentes necessários:

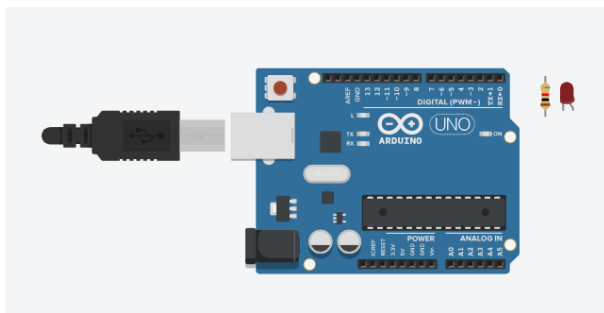
- Uma lâmpada de LED.
- Uma placa Arduino Uno R3.
- Fios e resistores.

Simulando no tinkercad:

- Abra o site do simulador Tinkercad (<https://tinkercad.com/>), faça login (caso ainda não tenha, o login deverá ser criado) e posteriormente crie um novo circuito (clique em Projetos -> Circuitos -> Criar -> Circuitos).
- Renomeie o circuito para o nome Projeto_01_Piscar_LED.

Montando o circuito:

- Adicione à tela os componentes necessários para a realização do projeto.



- Configure a resistência para ser a adequada ao dispositivo utilizado, para isso basta clicar no componente resistor e alterar o valor da resistência (para o LED vamos utilizar uma resistência de 220Ω).
- Posicione o resistor e a lâmpada de LED próximos à porta GND do Arduino.
- Agora vamos ligar os fios à placa Arduino Uno R3, devemos agora ligar a perna catódica do LED ao *GND* do Arduino clicando no terminal catódico e no *GND*.
- Alimente a lâmpada de LED:
 - Ligue o ânodo da lâmpada de LED ao resistor.
 - Ligue o resistor à porta 13 do Arduino.

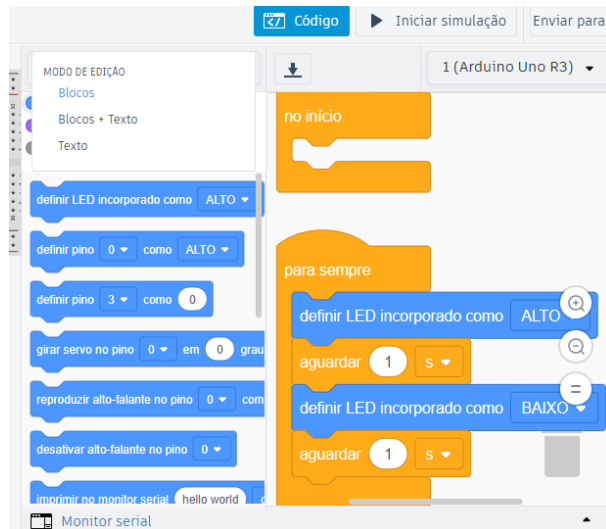
Montando o código:

Códigos utilizados nesta atividade:

- `pinMode(Nº ou LED_BUILTIN ,OUTPUT ou INPUT);`
- `digitalWrite(Nº, HIGH ou LOW);`
- `delay(Nº);`

Com isso em mente, vamos observar como fazer o LED piscar.

- Clique na opção código e no menu de seleção onde está escrito blocos, clique e selecione a opção texto



- Observe que existe um código padrão gerado pelo próprio aplicativo.

```
// C++ code
//
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

- Rodando este código, você poderá observar que o LED do Arduino já irá piscar. Este código é um código padrão gerado para verificar se o Arduino está corretamente conectado.
- Para ligarmos o LED, basta fazermos com que a porta onde o LED será alimentado se torne uma porta do tipo OUTPUT. Para fazermos da porta de número 13 uma porta de saída, basta executarmos o comando *pinMode (13, OUTPUT)*; no setup.
- Utilize o delay para configurar o tempo que você deseja que a lâmpada fique ligada.
- Link de acesso da atividade executada: https://www.tinkercad.com/things/am1hsTO80ur-projeto-1-piscar-led?sharecode=L7-wHsza5LZl_KuieQAkNxCF6J-G83vMBwgnRucyB_U.
- Não consulte o código antes de tentar finalizar o seu código, seja individualmente ou com ajuda.

EXPLORANDO A MATEMÁTICA NO ENSINO FUNDAMENTAL

1) Sabendo que o LED pisca uma vez a cada dois segundos, complete a tabela indicando quantas vezes ele irá piscar de acordo com o tempo que se passou.

Piscadas	Tempo Total (s)

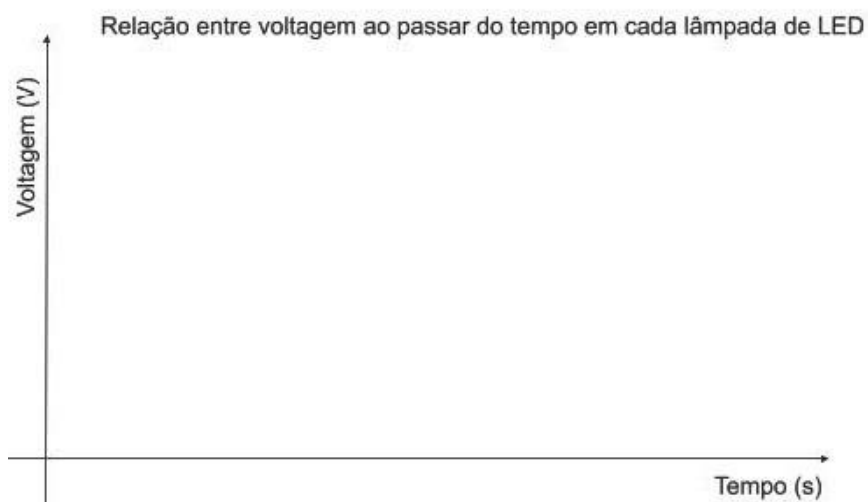
2) Você consegue relacionar ambos os valores que estão mudando?

3) Com base na palavra “variável”, a partir de uma dedução lógica, quais seriam as duas variáveis existentes no primeiro exercício?

4) Se mudarmos o valor do LED para que ele pisque uma vez a cada 3 segundos, quantas vezes ele terá piscado em 21 segundos? Quantos segundos são necessários para ele piscar 21 vezes?

EXPLORANDO A MATEMÁTICA NO ENSINO MÉDIO

1) Faça um gráfico representando a relação entre a voltagem recebida na lâmpada de LED ao passar do tempo



2) Neste gráfico, você consegue observar saltos na(s) linha(s) traçada(s) ou a(s) linha(s) se ligam de maneira contínua?

3) Compare a sua resposta e gráfico com o dos colegas e juntos tentem definir, a partir de uma dedução lógica, o significado do conceito continuidade de uma função.

ATIVIDADE 2: SEMÁFORO

Componentes necessários:

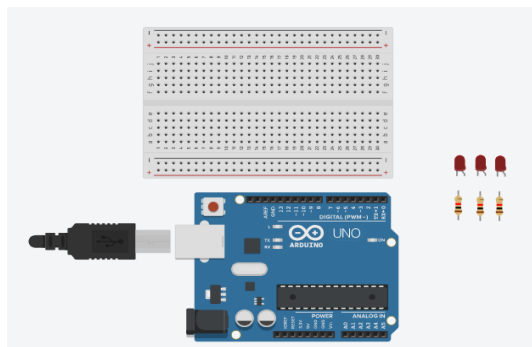
- Três lâmpadas de LED (de preferência uma unidade vermelha, uma amarela e uma verde).
- Placa de ensaio pequena (Protoboard).
- Uma placa Arduino Uno R3.
- Três resistores.
- Fios.

Simulando no tinkercad:

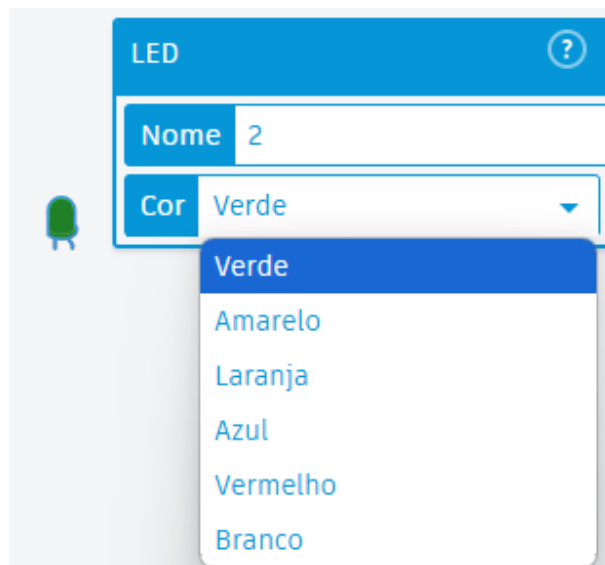
- Abra o site do simulador Tinkercad (<https://tinkercad.com/>), faça login (caso ainda não tenha, o login deverá ser criado) e posteriormente crie um novo circuito (clique em Projetos -> Circuitos -> Criar -> Circuitos).
- Renomeie o circuito para Projeto_02_Semaforo.

Montando o circuito:

- Adicione à tela os componentes necessários para a realização do projeto:

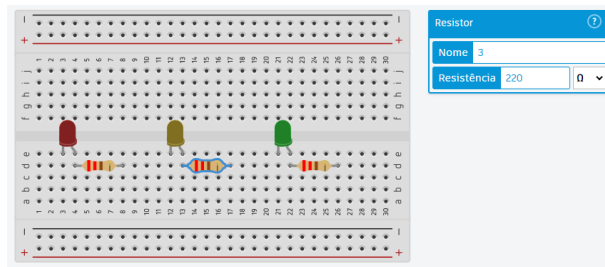


- Modifique a cor dos LEDs de forma que cada uma das três unidades de LED tenha a cor desejada, para isso basta clicar no LED desejado e selecionar a cor.

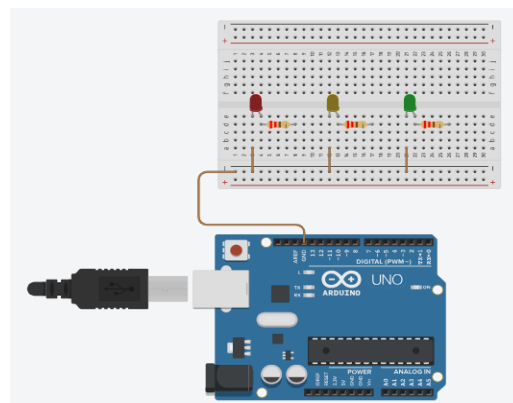


- Posicione as lâmpadas de LED na placa de ensaio, posicionando a perna catódica vermelha na posição e3, a amarela na posição e12 e a verde na posição e21.
- Configure a resistência para ser a adequada ao dispositivo utilizado, para isso basta clicar no componente resistor e alterar o valor da resistência (para o LED vamos utilizar uma resistência de 220Ω).
- Posicione os resistores na placa de ensaio, ligando o terminal 1 ao ânodo de cada um dos LEDs, tome cuidado para que ambos os terminais do resistor não coincidam na mesma coluna, para isso mova o resistor para posição horizontal selecionando-o e

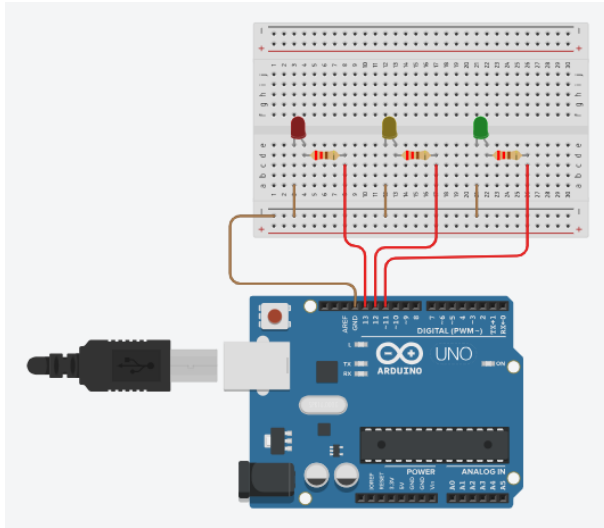
apertando a tecla R no teclado, posteriormente posicione nas posições d4, d13 e d22.



- Agora vamos ligar os fios à placa Arduino Uno R3, dessa forma devemos ligar as pernas catódicas do LED à posição negativa da placa de ensaio e posteriormente ligar essa posição negativa ao *GND* do Arduino. Utilize as posições a3, a12 e a21 para ligar na linha negativa do protoboard.
- Alimente as lâmpadas de LED ligando-as ao Arduino através do resistor. Ligue a porta 13 do Arduino à casa c8, a porta 12 do Arduino à casa c17 e a porta 13 do Arduino à casa c26 do protoboard.



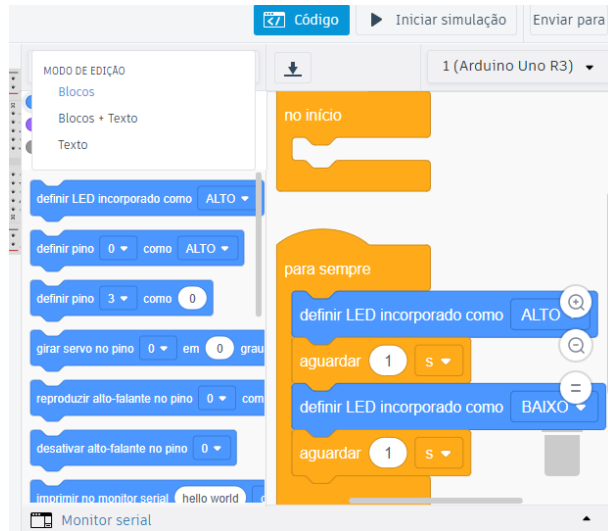
- Caso você tenha seguido todos os passos, é bem provável que o seu circuito se encontre desta maneira:



Montando o código:

Comandos utilizados:

- `pinMode(Nº, OUTPUT ou INPUT);`
- `digitalWrite(Nº, HIGH ou LOW);`
- `delay(Nº);`
- Clique na opção código e no menu de seleção onde está escrito blocos, clique e selecione a opção texto:



- Configure as portas selecionadas anteriormente como portas de saída a partir do comando `pinMode(13, OUTPUT);` dentro do `void setup()`.
- Com as portas configuradas, vamos ligar o primeiro LED a partir do comando `digitalWrite(11, HIGH);`.
- Utilize o delay para configurar o tempo que você deseja que a lâmpada verde fique ligada.
- Utilize agora novamente o comando `digitalWrite(nº, HIGH);` para ligar a próxima lâmpada (amarela) e o comando `digitalWrite(nº, LOW);` para desligar a lâmpada inicial (verde).
- Continue configurando de forma que, ao final do código, o ciclo de ligar e apagar de lâmpadas esteja completo, assim você terá conseguido concluir o seu primeiro semáforo. Não se

esqueça de utilizar o delay sempre que desejar que a lâmpada continue acesa por um período.

- Agora é com você, tente finalizar o código, caso você não consiga, você pode acessar. Caso você tenha alguma dúvida, pode consultar o código pronto:
<https://www.tinkercad.com/things/bt3T4vjwAjF-projeto02semaforo?sharecode=GdHzUURm3seac5YekHd9TsKN441gH7DdsBeo17KtlzU> .
- Não consulte o código antes de tentar finalizar o seu código, seja individualmente ou com ajuda.

EXPLORANDO A MATEMÁTICA NO ENSINO FUNDAMENTAL

- 1) Sabendo que o semáforo que fizemos acende por três segundos a lâmpada verde, um segundo a amarela e três segundos a lâmpada vermelha, calcule o tempo total que o programa leva para executar um ciclo completo.
- 2) Após exatamente um minuto funcionando, qual lâmpada estará ligada?
- 3) Se mantivermos o semáforo ligado por 70 segundos, quanto tempo a lâmpada verde terá permanecido ligada? Quantos ciclos completos foram executados?
- 4) Escreva a fração que representa o tempo em que o semáforo verde ficou aceso em relação ao tempo total de cada ciclo?

EXPLORANDO A MATEMÁTICA NO ENSINO MÉDIO

1) Represente graficamente a simulação realizada (registre como achar melhor, utilize o gráfico de uma função, ou gráfico de barras, colunas ou linhas, ou ainda outro qualquer utilizando sua criatividade).

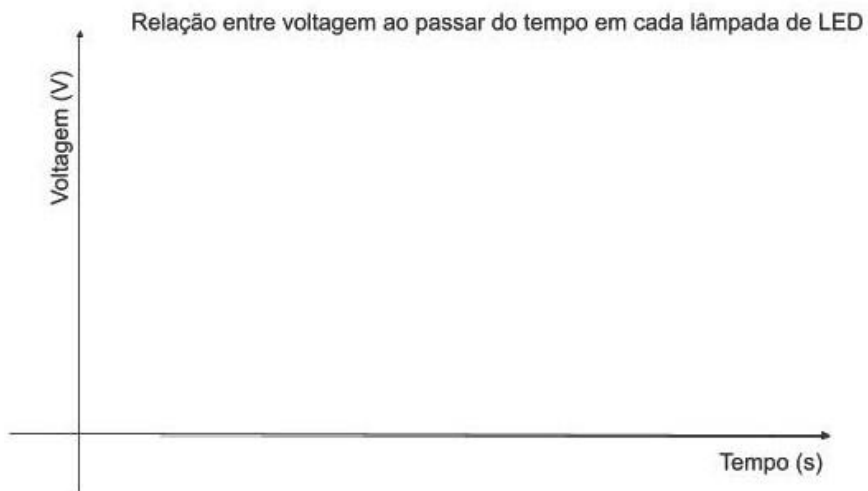


2) Compare o gráfico feito por você com o dos colegas

a) Após comparar o seu gráfico com o dos colegas, você acredita que poderia ter utilizado outro modelo de gráfico?

b) O que você mudaria no seu gráfico?

2) Faça um gráfico representando a relação entre a voltagem recebida em cada uma das lâmpadas de LED ao passar do tempo, utilizando cores diferentes para cada LED



- 4) Este gráfico poderia ser melhor representado por meio de três gráficos separados ou não?
- 5) Quais são as vantagens e desvantagens em representar três gráficos diferentes utilizando as mesmas variáveis?

ATIVIDADE 3: CONTROLANDO A INTENSIDADE LUMINOSA

Componentes necessários:

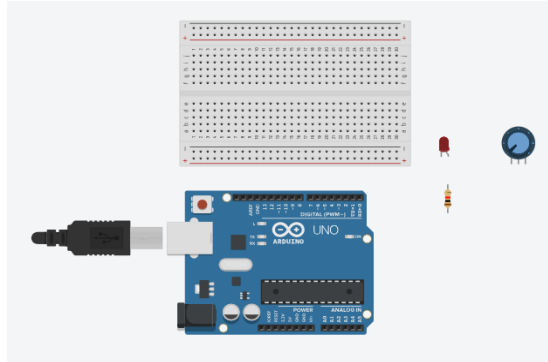
- Uma lâmpada de LED.
- Uma placa de ensaio (Protoboard).
- Uma placa Arduino Uno R3.
- Um potenciômetro.
- Fios e resistores.

Simulando no tinkercad:

- Abra o site do simulador Tinkercad (<https://tinkercad.com/>) faça login (caso ainda não tenha, o login deverá ser criado) e posteriormente crie um novo circuito (clique em Projetos -> Circuitos -> Criar -> Circuitos).
- Renomeie o circuito para
Projeto_03_Controlde_luminosidade.

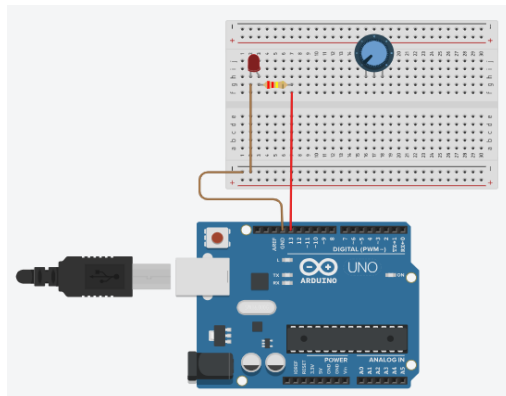
Montando o circuito:

- Adicione à tela os componentes necessários para a realização do projeto



- Posicione a lâmpada de LED na placa de ensaio de forma que a perna catódica fique na posição h2.
- Configure a resistência para ser a adequada ao dispositivo utilizado (para o LED vamos utilizar uma resistência de 220Ω).
- Posicione o resistor na placa de ensaio, ligando o terminal 1 ao ânodo do led, tome cuidado para que ambos os terminais do resistor não coincidam na mesma coluna, dessa forma mova o resistor para posição horizontal selecionando-o e apertando a tecla R no teclado, posteriormente posicione na posição g3 caso esteja seguindo as sugestões de posições.
- Agora vamos ligar os fios à placa Arduino Uno R3, dessa forma devemos ligar a perna catódica do LED à posição negativa da placa de ensaio e posteriormente ligar essa posição negativa ao *GND* do Arduino utilizando a posição g2 para ligar na linha negativa do protoboard.

- Alimente a lâmpada de LED ligando-a ao Arduino através do resistor, ligando a porta 13 do Arduino à casa f7.



- Agora vamos ligar o potenciômetro, colocando o potenciômetro na placa de ensaio e ligando o terminal 1 na posição h16.
- Configure o potenciômetro para que a resistência dele seja a adequada ao dispositivo utilizado (para o potenciômetro vamos utilizar uma resistência de $10\text{ k}\Omega$).
- Alimente o terminal 1 do potenciômetro com os 5V gerados pelo *Arduino*.
- Aterre o terminal 3 do potenciômetro na posição negativa da placa de ensaio, e aterre posteriormente a placa no *GND* do Arduino, utilize a posição g18 da placa de ensaio.
- Alimente o terminal 2 em uma das portas analógicas do Arduino (Obs.: Como o resultado gerado pelo potenciômetro é analógico, devemos ter o cuidado de ligar em portas capazes de ler sinais analógicos, ou seja, uma das portas entre A0 e A5).

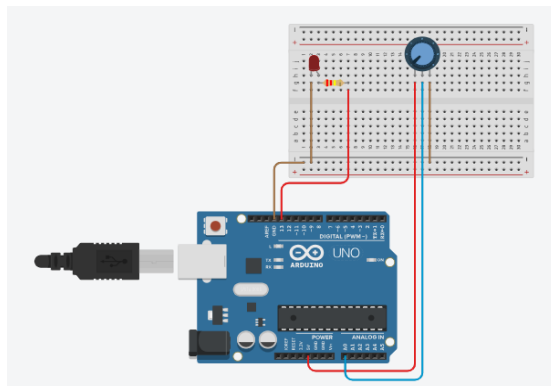
Dessa forma, utilize a porta A0 para fazer a ligação com a porta g17 da placa de ensaio.

- Caso você tenha seguido a sugestão, é bem provável que o seu circuito se encontre desta maneira:

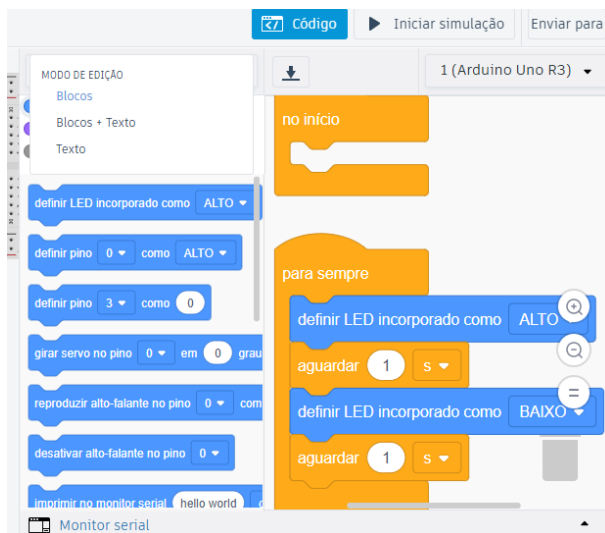
Montando o código:

Códigos utilizados:

- `Serial.begin(9600);`
- `pinMode(Nº, OUTPUT ou INPUT);`
- `int VARIÁVEL;`
- `analogRead(PORTA);`
- `map(valor, deMenor, deMaior, paraMenor, paraMaior)analogWrite(Nº , VALOR);`
- `Serial.println(VARIÁVEL);`



- Clique na opção código e no menu de seleção onde está escrito blocos, clique e selecione a opção texto:



- Inicie a taxa de bits de leitura dos componentes através do `Serial.begin(9600);`
- Configure a porta selecionada anteriormente como porta de saída a partir do comando `pinMode();` dentro do `void setup()`.
- Com a porta configurada, utilize o comando `int qtdLuz=0;` para declarar uma variável do tipo inteiro para receber os valores gerados pelo potenciômetro e inicializar o valor da variável como zero.
- Através do comando `analogRead();` vamos atribuir à variável criada anteriormente o valor lido no potenciômetro utilizando o comando `qtdLuz=analogRead(A0);`

- Modifique os valores gerados pelo potenciômetro através do comando `map()`. (Obs.: O valor modificado deve ser armazenado em uma variável que poderá ser a mesma variável que você utilizou anteriormente).
- Agora é com você, tente finalizar o código, caso você não consiga, você pode acessar. Caso você tenha alguma dúvida, pode consultar o código pronto:
<https://www.tinkercad.com/things/6I03uxcvzdV-projeto03controladeluminosidade?sharecode=L1yWooS8DMIk-gmR3-kDQW2s6be-s-NufGP57ML3Vpo> .
- Não consulte o código antes de tentar finalizar o seu código, seja individualmente ou com ajuda.

EXPLORANDO A MATEMÁTICA NO ENSINO FUNDAMENTAL

1) Avaliando o desenho e observando o que foi feito, indique qual das três lâmpadas está mais forte



Fonte: Elaborado pelo autor com auxílio de inteligência artificial (ChatGPT/OpenAI), 2026.

2) Lúmens é a unidade de medida que identifica a intensidade luminosa em um ambiente. Baseado nisso, um ambiente mais iluminado possui mais ou menos lúmens?

EXPLORANDO A MATEMÁTICA NO ENSINO MÉDIO

1) Registre o resultado obtido mediante um gráfico. Nesta atividade, você pode registrar o seu gráfico da maneira como achar melhor, não existe certo ou errado. Você pode fazer um gráfico em formato de função, um gráfico de barras, colunas, linhas ou até algum tipo de gráfico diferente, utilize a criatividade:

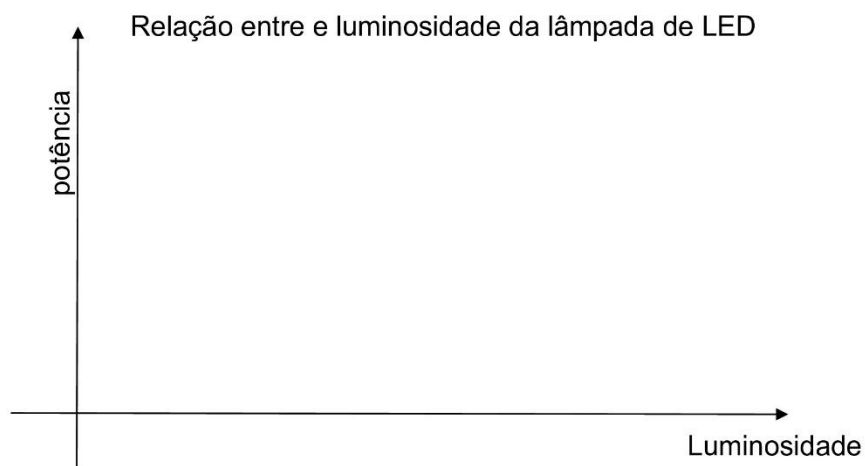


2) Qual o tipo de gráfico você escolheu? Por que você escolheu este tipo de gráfico?

3) Compare o gráfico feito por você com o dos colegas

- a) Após comparar o seu gráfico com o dos colegas, você acredita que poderia ter utilizado outro modelo de gráfico?
- b) O que você mudaria no gráfico feito por você?

4) Complete o gráfico abaixo utilizando um gráfico de linhas que irá comparar



5) Neste gráfico, você consegue observar saltos na(s) linha(s) traçada(s) ou a(s) linha(s) se ligam de maneira contínua?

6) Podemos afirmar, a partir dos conhecimentos adquiridos, que a função obtida é contínua?

ATIVIDADE 4: ALARME

Componentes necessários:

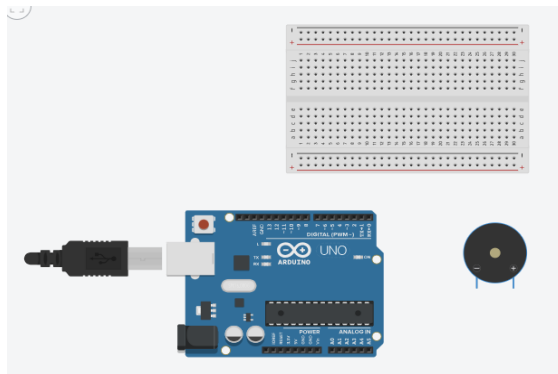
- Uma placa de ensaio (Protoboard).
- Uma placa Arduino Uno R3.
- Um piezo (buzzer).
- Fios.

Simulando no tinkercad:

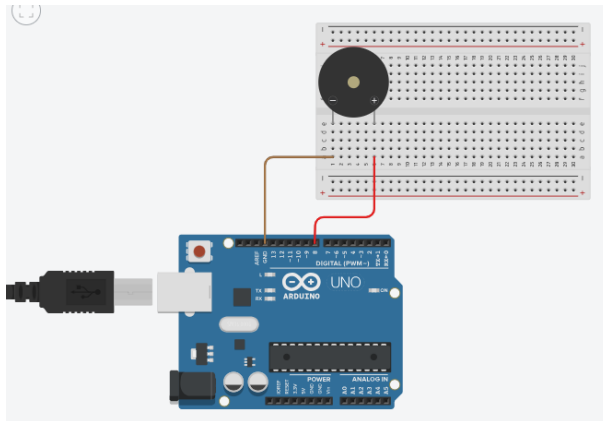
- Abra o site do simulador Tinkercad (<https://tinkercad.com/>) faça login (caso ainda não tenha, o login deverá ser criado) e posteriormente crie um novo circuito (clique em Projetos -> Circuitos -> Criar -> Circuitos).
- Renomeie o circuito para Projeto_04_Alarmes.

Montando o circuito:

- Adicione à tela os componentes necessários para a realização do projeto:



- Posicione o piezo (buzzer) na placa de ensaio, posicionando o terminal negativo na posição e1.
- Agora vamos ligar os fios à placa Arduino Uno R3, aterre o terminal negativo na parte negativa da placa de ensaio ligando a linha horizontal da placa ao *GND* do Arduino.
- Alimente o terminal positivo do Arduino em alguma das portas digitais do Arduino, ligando o terminal positivo à porta 8 do Arduino.
- Caso você tenha seguido o passo a passo, é bem provável que o seu circuito se encontre desta maneira:



Montando o código:

Comandos utilizados:

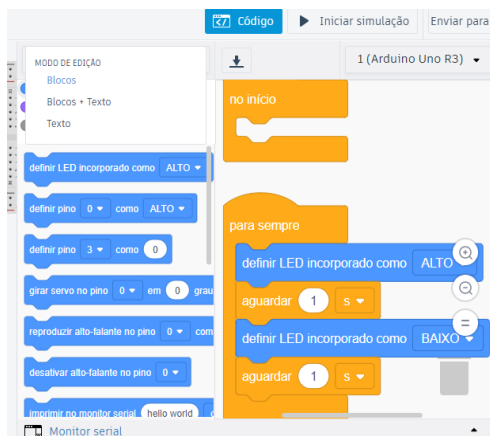
- `for (inicialização; condição; incremento) { }`
- `Serial.begin(9600);`
- `pinMode(Nº, OUTPUT ou INPUT);`

- `int VARIÁVEL;`
- `Serial.println(VARIÁVEL);`
- `tone(Nº;FREQUÊNCIA);`
- **Declaração de variáveis globais e locais**

Vale ressaltar também que o componente piezo (buzzer) possui limitações (por exemplo, ao receber valores negativos), assim alguns valores de referência passados ao piezo (buzzer) podem fazer com que o componente não suporte atualizações dos valores, gerando possíveis travamentos.

Com isso em mente, vamos realizar o código para criarmos os nossos alarmes. É relevante mencionar que existem diferentes tipos de alarmes, e que agora iremos sugerir diferentes tipos de código para podermos visualizar diferentes opções. Cada opção sugerida será disponibilizada posteriormente. Aconselhamos que você tente fazer o seu alarme de uma maneira diferente.

- Clique na opção código e no menu de seleção onde está escrito blocos, clique e selecione a opção texto.



- Vamos começar este código declarando algumas variáveis globais que serão utilizadas por todo o código. Precisaremos aqui de uma variável global que irá modificar o valor base do som gerado pelo piezo (buzzer), sendo que valores do piezo (buzzer) maiores farão com que o som se torne mais agudo. Declare dessa maneira uma variável do tipo inteiro chamada *base* e atribua a ela um valor de 500.
- Configure a porta selecionada anteriormente como porta de saída a partir do comando *pinMode()*; dentro do *void setup()*.
- Com a porta configurada, vamos declarar duas variáveis: uma do tipo inteiro para receber o valor de frequência e outra do tipo float para receber um valor de seno posterior. É importante observar que variáveis do tipo *int* recebem somente valores inteiros, enquanto variáveis do tipo *float* recebem valores com vírgula. Como o seno de um ângulo será um valor entre 0 e 1, geralmente o seu valor não poderá ser expresso de forma inteira, assim teremos que utilizar uma variável do tipo *float* para armazenar o seu valor. Declarar as variáveis sendo: *float valorSeno*; e *int valorFreq*.

A partir daqui, iremos agora separar em diferentes exemplos:

EXEMPLO 1: Observando o funcionamento do piezo (buzzer) e os diferentes tons gerados

- Crie um laço de repetição que iniciará em 0 e terminará em 30, com um incremento de um em um.
- No laço de repetição, faça com que o valor recebido pela *base* seja incrementado em 100, assim o valor recebido pela base será de *base=base+100*;
- Execute o comando `tone` utilizando o n° da porta em que ele foi ligado e a base (que neste caso será a frequência).
- Utilize o `delay` para conseguir ouvir o som de maneira clara, aqui poderá ser utilizado o valor de *delay (500)*;
- Após o laço de repetição, lembre-se de voltar o valor da base para o valor de 100.
- Obs.: Você pode aqui também brincar com esses valores, diminuindo, por exemplo, o incremento da base para 1 e o `delay` para 5 e aumentando o número de repetições, utilize a criatividade.
- Você pode também visualizar no monitor serial os valores encontrados. Para isso, você deverá utilizar o *serialBegin()*; para inicializar o monitor serial e utilizar o comando

`Serial.println(VARIAVEL);` para indicar a variável a ser visualizada.

- Agora é com você, tente finalizar o código, caso você não consiga, você pode acessar. Caso você tenha alguma dúvida, você pode consultar o código pronto: https://www.tinkercad.com/things/2sO94qk7S0Z-projeto04modelo1alarme?sharecode=hRD_AYU5gGNoQ2rRVvu_4Q_kS8MuUzXfKF_78APP2mA .

EXEMPLO 2: Alarme de incêndio

- Crie um laço de repetição que iniciará em 0 e terminará em 180, com um incremento de um em um.
- No laço de repetição, faça com que o valor recebido pelo *valorSeno* seja o seno do ângulo desejado. Para isso, utilizaremos a função $\sin(x)$, onde x é o valor em radianos do ângulo. Obs.: o comando $\text{valorSeno}=(\sin(x*(3.1416/180)))$; fará uma equação que retornará o valor do seno em radianos.
- Agora faça com que o valor da frequência receba o valor do seno gerado multiplicado por 1000 (visto que o valor de seno é sempre um valor entre 0 e 1) acrescido do valor do tom base.
- Execute o comando `tone` utilizando o nº da porta em que ele foi ligado e a base (que neste caso será a frequência).

- Utilize o `delay` para conseguir ouvir o som de maneira clara, aqui o valor utilizado deverá ser de 2, dessa forma utilize o comando `delay (2);`.
- Obs.: Você pode aqui também brincar com esses valores, diminuindo, por exemplo, o incremento da base para 1 e o `delay` para 5 e aumentando o número de repetições, utilize a criatividade.
- Obs.: Você pode aqui também brincar com esses valores, diminuindo e aumentando o `delay` ou a frequência.
- Agora é com você, tente finalizar o código, caso você não consiga, você pode acessar. Caso você tenha alguma dúvida, você pode consultar o código pronto: https://www.tinkercad.com/things/8PC2fR5je3X-projeto04modelo2alarmeincendio?sharecode=91YJ_Kz8ESZpX1szAra6NIV3K0-j3yEuVrg5oniPQ9Q.

EXEMPLO 3: Alarme de ambulância

- Crie um laço de repetição que iniciará em 0 e terminará em 180, com um incremento de um em um.
- No laço de repetição, faça com que o valor recebido pelo *valorSeno* seja o seno do ângulo desejado. Para isso, utilizaremos

a função $\sin(x)$, onde x é o valor em radianos do ângulo. Utilize a equação $valorSeno=(\sin(x*(3.1416/180)))$;

- Agora faça com que o valor da frequência receba o valor do seno gerado multiplicado por 1000 (visto que o valor de seno é sempre um valor entre 0 e 1) acrescido do valor do tom base.
- Execute o comando `tone` utilizando o nº da porta em que ele foi ligado e a base (que neste caso será a frequência).
- Utilize o `delay` para conseguir ouvir de maneira clara, aqui o valor utilizado deverá ser de 20, utilize o comando `delay(20)`;
- Obs.: Você pode aqui também brincar com esses valores, diminuindo e aumentando o `delay` ou a frequência.
- Você pode também visualizar no monitor serial os valores encontrados. Para isso, você deverá utilizar o `serialBegin()`; para inicializar o monitor serial e utilizar o comando `Serial.println(VARIÁVEL)`; para indicar a variável a ser visualizada.
- Agora é com você, tente finalizar o código, caso você não consiga, você pode acessar. Caso você tenha alguma dúvida, você pode consultar o código pronto: <https://www.tinkercad.com/things/eOeUxfa6aQB-projeto04modelo3alarmeambulancia?sharecode=CL2wlZN8MoC9lROI19Xxaq8QJgibPv0C0aRFchm8-4I>.

EXEMPLO 4: Alarme com Seno completo

- Modifique o valor da base para 1000.
- Inicie o monitor serial através do comando `serialBegin()`;
- Crie um laço de repetição que iniciará em 0 e terminará em 360, com um incremento de um em um.
- No laço de repetição, faça com que o valor recebido pelo `valorSeno` seja o seno do ângulo desejado. Para isso, utilizaremos a função $\sin(x)$, onde x é o valor em radianos do ângulo. Utilize a equação $\text{valorSeno} = (\sin(x * (3.1416 / 180)))$;
- Agora faça com que o valor da frequência receba o valor do seno gerado multiplicado por 1000 (visto que o valor de seno é sempre um valor entre 0 e 1) acrescido do valor do tom base.
- Utilize o comando `serial.println(valorFreq)`; para visualizar o gráfico gerado pela frequência utilizada.
- Execute o comando `tone` utilizando o n° da porta em que ele foi ligado e a base (que neste caso será a frequência).
- Utilize o `delay` para conseguir ouvir o som de maneira clara, aqui o valor utilizado deverá ser de 20. Utilize o comando `delay(20)`;
- Obs.: Você pode aqui também brincar com esses valores, diminuindo e aumentando o `delay` ou a frequência.

- Agora é com você, tente finalizar o código, caso você não consiga, você pode acessar. Caso você tenha alguma dúvida, você pode consultar o código pronto:
https://www.tinkercad.com/things/6uIcMPKMvxE-projeto04modelo4senocompleto?sharecode=uIQ_ZrUl5huIzoLN0pIKxhdQBGLSG9YHXZz9GFDsM3c .

EXPLORANDO A MATEMÁTICA NO ENSINO FUNDAMENTAL

- 1) Crie uma sequência utilizando setas ($\uparrow \downarrow \rightarrow \leftarrow \nearrow \nwarrow \searrow \swarrow$) que demonstre o que está acontecendo com o som do alarme ao passar do tempo.
- 2) Sabendo que dentro do programa cada leitura do laço de repetição dura 20 milissegundos e que o laço é repetido 180 vezes, quantos milissegundos dura um ciclo completo da sirene da ambulância?
- 3) Quantos segundos irá durar esse ciclo?
- 4) Em um minuto, quantos ciclos completos terão sido executados?

EXPLORANDO A MATEMÁTICA NO ENSINO MÉDIO

1) Registre o resultado obtido por meio de um gráfico. Nesta atividade, você pode registrar o seu gráfico da maneira como achar melhor, não existe certo ou errado, você pode fazer um gráfico em formato de função, um gráfico de barras, colunas, linhas ou até algum tipo de gráfico diferente, utilize a criatividade.



- 2) Quais são as variáveis que se relacionam neste gráfico?
- 3) Qual é o tipo de gráfico gerado pelo monitor serial?
- 4) Neste gráfico, você consegue observar saltos na(s) linha(s) traçada(s) ou a(s) linha(s) se ligam de maneira contínua?

5) Você tentou utilizar o monitor serial em outras atividades, ou tentou fazer modelos diferentes dos sugeridos? Se sim, quais foram os modelos que você tentou utilizar e quais os resultados obtidos?

REFERÊNCIAS

CAMPOS, F. R. **Currículo, Tecnologias e Robótica na Educação Básica**. São Paulo: PUC, 2011. Disponível em: <https://sapiencia.pucsp.br/bitstream/handle/9619/1/Flavio%20Rodrigues%20Campos.pdf>. Acesso em: 11 abr. 2025

FARIA, Rejane Waiandt Schuwartz de Carvalho; SANTOS, Silvana Claudia dos; BRANDÃO, Alexandre Santos. Experiências pedagógicas com robótica na produção de conhecimentos matemáticos na educação básica. In: SUMARIVA, Guilherme (org.). *Robótica, inovação e educação: conexões para o futuro*. Editora V&V, 2026. p. 148–160.

GOMES, Cristiane Grava; SILVA, Fernando Oliveira da; BOTELHO, Jaqueline da Costa; SOUZA, Aguinaldo Robinson de. A robótica como facilitadora do processo ensino-aprendizagem de Matemática no ensino fundamental. In: PIROLA, Nelson Antonio (org.). **Ensino de ciências e matemática IV: temas de investigação**. São Paulo: Editora UNESP; Cultura Acadêmica, 2010. p. 205–219. Disponível em: <https://books.scielo.org/id/bpkng/pdf/pirola-9788579830815-11.pdf>. Acesso em: 11 abr. 2025.

PAPERT, S. **Mindstorms: Children, Computers, and Powerful Ideas**. Basic Books, 1980.

RIBEIRO, Augusto César Castro; ALMEIDA, Letícia Pereira de; OLIVEIRA, Mayara Bonifácio de; ROMULO, Thaymara Cristina de Souza; FARIA, Rejane Waiandt Schuwartz de Carvalho. *Robótica na escola: práticas inovadoras com matemática na educação básica*. In: SILVA, Cristina Maria da; ALMEIDA, Letícia Pereira de; EUCLIDES, Maria Simone; OLIVEIRA, Mayara Bonifácio de; FARIA, Rejane Waiandt Schuwartz de Carvalho; SILVA, Yuri Leandro Cupertino (orgs.). *Tecendo saberes: sujeitos, práticas e resistências na educação pública*. Viçosa: Editora Navegando, 2025. p. 161-177. Disponível em: <https://www.editoranavegando.com/tecendo-saberes>. Acesso em: 26 Mar. 2026.

RIBEIRO, Augusto César Castro. *Robótica Educacional na Formação Continuada de Professores: uma perspectiva da Educação Matemática*. 2026. Dissertação (Mestrado em Educação) – Universidade Federal de Viçosa, Viçosa - MG, 2026.

Em um mundo cada vez mais conectado e tecnológico, a escola é desafiada a reinventar suas práticas e preparar estudantes para novas formas de aprender, criar e solucionar problemas. É nesse contexto que o livro *NextEd Formação Continuada Docente em Robótica Educacional* surge como um convite à inovação pedagógica, aproximando professores da Educação Básica do universo da Robótica de maneira acessível, prática e interdisciplinar.

O material apresenta conceitos introdutórios sobre Robótica Educacional, Arduino, componentes eletrônicos, programação e ferramentas digitais, além de propostas de atividades que dialogam com a realidade escolar.

Mais do que um manual técnico, esta obra busca fortalecer a formação docente, estimulando práticas criativas, investigativas e colaborativas em sala de aula. Ao integrar tecnologia e educação, os autores propõem caminhos para que professores se tornem protagonistas na construção de experiências de aprendizagem mais significativas, despertando nos estudantes a curiosidade, o pensamento crítico e a autonomia diante dos desafios do século XXI.



FAPEMIG