



Desenvolvimento de currículo STEM para educação de eletrônica e robótica no ensino secundário

Lorena Pereira de Carlos Jeranoski

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para
obtenção do Grau de Mestre em Engenharia Industrial - Ramo Engenharia Mecânica.

Trabalho orientado por:

Professor PhD Paulo Leitão

Professor MSc Valfredo Pilla Júnior

Bragança

2022-2023



Desenvolvimento de currículo STEM para educação de eletrônica e robótica no ensino secundário

Lorena Pereira de Carlos Jeranoski

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para
obtenção do Grau de Mestre em Engenharia Industrial - Ramo Engenharia Mecânica.

Trabalho orientado por:

Professor PhD Paulo Leitão

Professor MSc Valfredo Pilla Júnior

Bragança

2022-2023

Agradecimentos

Agradeço principalmente a Deus por me dar o dom da vida, e nunca me abandonar na minha caminhada. Agradeço a minha mãe Rosinês e ao meu pai Jefferson, por me criarem em um ambiente cheio de amor para ser uma mulher independente e batalhadora. Agradeço a minha irmã Luana, por sempre me lembrar o quão forte eu posso ser. Agradeço ao meu afilhado Leonardo, por me motivar a ser sempre melhor por ele. Agradeço aos meus avôs e avós por me mostrarem que a vida vale a pena. Agradeço ao meu padrinho Geraldo Leonel por, durante toda a minha vida, me apoiar mesmo de longe.

Agradeço a todos os meus amigos por aplaudirem minhas vitórias e me apoiarem nas minhas derrotas. E aos que estiveram ativamente nessa jornada comigo, Pedro, Vitória, Leonardo, Ingrid, João, Lucas e Daniel. Vocês foram essenciais para que eu tivesse forças para não desistir, eu rezo todos os dias pela vida de cada um.

Agradeço a Professora Rute Gaspar por ter acreditado no meu trabalho. Hoje eu escrevo essa dissertação porque tive o prazer de encontrar uma mulher com vontade de inovar e abraçar meu projeto. Agradeço ao Agrupamento de Escolas Miguel Torga e ao Agrupamento de Escolas Emídio Garcia por me acolherem para implementar meu projeto. Agradeço ao meu orientador Professor Doutor Paulo Leitão por me guiar nesse processo.

Não posso deixar de agradecer a cantina do IPB, a qual tive o prazer de ser colaboradora, e com isso ter um alimento de qualidade todos os dias nas refeições. Isso possibilita a tantos alunos continuar estudando, e eu serei eternamente grata.

Resumo

As competências tecnológicas são cada vez mais necessárias, reforçadas pela transformação digital que a Indústria 4.0 está a promover. Nesse contexto, a inclusão da educação Science, Technology and Mathematics (STEM) na educação básica torna-se uma importante aliada, visando que a criança desenvolva capacidades que posteriormente serão importantes para sua vivência. Como resolução de problemas, protagonismo, autonomia, pensamento computacional, criatividade, inovação e trabalho em equipe. A necessidade deste projeto é dada pela falta de materiais estruturados para a educação STEM no ensino médio. Foi realizado um projeto de Investigação em Design Educacional (IDE) para criar uma série de quatro projetos de eletrônica e robótica, com o objetivo de introduzir iniciantes nesse campo e promover o ensino STEM. Os projetos incluíram guias de instruções claras e combinaram métodos tradicionais e interativos. Módulos adicionais chamados "Gotas de Conhecimento" foram desenvolvidos para aprofundar os conhecimentos dos alunos interessados. Avaliações foram realizadas para medir a eficácia dos métodos de ensino, com a participação de alunos do 9º e 12º ano. A avaliação de dois docentes foi feita através de um formulário *online* e foi utilizada para alimentar o ciclo da IDE, possibilitando a melhoria dos materiais a cada projeto. Os dados de 18 alunos do 9º ano e 29 alunos do 12º ano foram adquiridos através da plataforma Kahoot!, com o intuito de tornar a aprendizagem mais interativo e envolvente. Os resultados foram encorajadores, onde tanto as turmas apresentaram uma melhora significativa na compreensão e aplicação dos conceitos abordados no material didático, com um aumento de 26.4% na compreensão do primeiro projeto aplicado em comparação com o último, como também se mostraram mais comprometidos e motivados em sua aprendizagem, conforme evidenciado por sua

maior participação nas discussões e atividades em sala de aula. Os professores também expressaram satisfação com o material, atendendo às necessidades de diversos alunos. Em conclusão, a abordagem utilizada mostrou-se eficaz e envolvente para a aprendizagem dos alunos.

Palavras-chave: Educação STEM; Educação tecnológica.

Abstract

Technological skills are increasingly necessary, reinforced by the digital transformation that Industry 4.0 is promoting. In this context, the inclusion of STEM education in basic education becomes an important ally, aiming for the child to develop skills that will later be important for their experience. Such as problem solving, protagonism, autonomy, computational thinking, creativity, innovation and teamwork. The need for this project is given by the lack of structured materials for STEM education in high school. An Educational design Project (EDP) was undertaken to create a series of four electronics and robotics projects, with the aim of introducing beginners to this field and promoting STEM teaching. Projects included clear how-to guides and combined traditional and interactive methods. Additional modules called "Knowledge Drops" were developed to deepen the knowledge of interested students. Assessments were carried out to measure the effectiveness of teaching methods, with the participation of 9th and 12th grade students. The evaluation of two professors was carried out using an *online* form and was used to feed the EDP cycle, enabling the improvement of materials in each project. Data from 18 9th graders and 29 12th graders were collected through the Kahoot! platform, with the aim of making learning more interactive and engaging. The results were encouraging, where both the classes showed a significant improvement in the understanding and application of the concepts addressed in the didactic material, with an increase of 26.4% in the understanding of the first project applied in comparison with the last one, as well as being more engaged and motivated in their learning, as evidenced by their increased participation in classroom discussions and activities. Teachers also

expressed satisfaction with the material, meeting the needs of diverse students. In conclusion, the approach used proved to be effective and engaging for student learning.

Keywords: STEM education; Technological education.

Conteúdo

Agradecimentos	v
Resumo	vi
Abstract	viii
1 Introdução	1
1.1 Contexto e motivação	1
1.2 Objetivos	2
1.3 Estrutura do documento	3
2 Revisão da literatura	5
2.1 Educação tecnológica	5
2.1.1 Educação STEM	6
2.1.2 Robótica educacional	7
2.2 Forma de aprendizagem do indivíduo	8
2.2.1 Níveis de desenvolvimento	9
2.2.2 Importância da brincadeira na aprendizagem	10
2.2.3 Papel da escola	11
2.3 Métodos de ensino	11
2.3.1 Instrucionismo	12
2.3.2 Construcionismo	12
2.4 investigação em design educacional	15

2.4.1	Ciclos, fases e outras características	15
2.4.2	Avaliações formativas	16
2.4.3	Metodologia adotada no projeto	17
3	Fase 1: Análise e exploração	19
3.1	Análise e exploração de métodos instrucionistas	21
3.2	Análise e exploração de métodos construcionistas	23
4	Fase 2: Desenvolvimento e prototipagem	27
4.1	Guião de atividades	28
4.1.1	Introdução	28
4.1.2	Simulação	29
4.1.3	Sistema físico	30
4.1.4	Atividades colaborativas	31
4.1.5	Atividades interativas	32
4.1.6	Desafio	32
4.2	Gotas de conhecimento	33
4.3	Materiais de avaliação e <i>feedback</i>	35
4.3.1	Avaliação dos discentes	35
4.3.2	<i>Feedback</i> dos docentes	37
4.3.3	<i>Feedback</i> dos discentes	39
5	Fase 3: Implementação e avaliação	42
5.1	Resultados e avaliações 9 ^o ano	44
5.1.1	Projeto 1 - Blink	45
5.1.2	Projeto 2 - Campanha	47
5.1.3	Projeto 3 - Sensor de distância	49
5.1.4	<i>Feedback</i> dos alunos	50
5.2	Avaliação e resultados 12 ^o ano	51
5.2.1	<i>Feedback</i> dos alunos	52

5.3	<i>Feedback</i> docentes	54
6	Conclusões e trabalhos futuros	55
A	Projeto 1 - Blink	A2
B	Módulo 1 - Introdução ao <i>Hardware</i> e <i>Software</i>	B1
C	Módulo 2 - Comentários e variáveis no código	C1
D	Módulo 3 - Grandezas digitais e analógicas	D1
E	Módulo 4 - Funções	E1
F	Módulo 5 - <i>Breadboard</i>	F1
G	Módulo 6 - Resistências elétricas	G1
H	Módulo 7 - LED	H1
I	Projeto 2 - Campanha	I1
J	Módulo 8 - <i>Push button</i> e <i>Buzzer</i>	J1
K	Módulo 9 - Monitor serial	K1
L	Projeto 3 - Sensor de aproximação	L1
M	Questionario qualitativo alunos	M1
N	Questionario quantitativo alunos	N1

Lista de Tabelas

2.1	Comparação dos métodos de ensino utilizados	14
3.1	Cronograma de análise e exploração	20
5.1	Calendário de aplicação dos projetos	44
5.2	Resultados da aplicação do Projeto 1 - 9º ano	45
5.3	Resultados da aplicação do Projeto 2 - 9º ano	47
5.4	Resultados da aplicação do Projeto 3 - 9º ano	49
5.5	Resultados da aplicação do Projeto 1 - 12º ano	52

Lista de Figuras

1.1	Descrição das partes da estrutura de cada projeto	3
2.1	Níveis de desenvolvimento. Adaptada de [28]	10
2.2	Iterações de ciclos sistemáticos [42]	16
2.3	Camadas de avaliação formativa [42]	17
2.4	Estrutura geral utilizada na implementação da IDE	18
3.1	Exemplo atividade "Blink"	22
3.2	Exemplo atividade "Sistema de segurança"	23
3.3	Estrutura e conteúdo de cada projeto do material	24
4.1	Página introdutória do guião	28
4.2	Página de instruções para simulação	29
4.3	Página de explicação de montagem do sistema físico	30
4.4	Exemplo de atividades colaborativas inclusas no guião	31
4.5	Exemplo de atividades interativas "Pense sempre além"	32
4.6	Modelo de desafio proposto ao fim do guião	32
4.7	Lista das gotas de conhecimento correspondentes ao projeto	33
4.8	Exemplo de um módulo de gota de conhecimento	34
4.9	Pergunta prática	36
4.10	Pergunta teórica	36
4.11	Pergunta de motivação	36
4.12	Perguntas formulário <i>feedback</i> docentes (a)	37

4.13	Perguntas formulário <i>feedback</i> docentes (b)	38
4.14	Perguntas formulário <i>feedback</i> docentes (c)	38
4.15	Perguntas formulário <i>feedback</i> docentes (d)	39
4.16	Perguntas formulário <i>feedback</i> discentes (a)	40
4.17	Perguntas formulário <i>feedback</i> discentes (b)	40
4.18	Perguntas formulário <i>feedback</i> discentes (c)	41
5.1	Divulgação do Projeto no site da Escola Emídio Garcia	43
5.2	Gráfico correspondente a taxa de acerto por pergunta - Projeto 1	46
5.3	Gráfico correspondente a taxa de acerto por pergunta - Projeto 2	48
5.4	Evolução de taxa de acerto dos projetos	50
5.5	Avaliação dos <i>feedbacks</i> dos alunos 9º ano	50
5.6	Avaliação dos <i>feedbacks</i> dos alunos 12º ano	53

Acrónimos

EDP Educational design Project. viii

IDE Investigação em Design Educacional. vi, xvii, 3, 4, 18, 19, 22, 24, 26, 27, 37, 41, 46–49, 51, 52

NDP Nível de Desenvolvimento Potencial. 10, 13

NDR Nível de Desenvolvimento Real. 9

PADDE Plano de Ação Para o Desenvolvimento Digital da Escola. 42

QI Quociente de Inteligência. 10

STEM Science, Technology and Mathematics. vi–ix, 1, 2, 5, 6, 8, 18, 19, 44, 51, 54, 55

TIC Tecnologias da informação e comunicação. 3, 43, 44, 57

Capítulo 1

Introdução

A presente dissertação foi desenvolvida com o objetivo de concluir o Mestrado em Engenharia Industrial com Ramo em Engenharia Mecânica. Para isso, foi proposto o tema Desenvolvimento de currículo STEM para educação de eletrônica e robótica no ensino secundário. Neste capítulo é feita uma apresentação do contexto e da motivação que se insere esta dissertação. Ao final do capítulo são apresentados os objetivos e a estrutura do documento.

1.1 Contexto e motivação

Todo desenvolvimento na tecnologia dá novas ferramentas para que o homem possa se apropriar da natureza e tenha a possibilidade de transformá-la conforme seus interesses. O desenvolvimento massivo da tecnologia impacta principalmente na educação, onde se espera desenvolver conhecimentos e *skills* que caminhem junto com a necessidade que essa evolução apresenta [1].

As novas profissões necessitam uma força de trabalho que envolvem tópicos como criatividade, inovação, pensamento crítico, resolução lógica de problemas e alfabetização tecnológica [2], fatores que impulsionam a procura pela educação STEM. Além disso, à medida que a tecnologia avança e se torna mais acessível, aumenta a necessidade por alunos que estejam comprometidos a desenvolver capacidades STEM.

Esse contexto resultou em um aumento drástico pela procura da robótica educacional nas últimas duas décadas [3], pois é capaz de aumentar a motivação, colaboração e persistência quando os alunos se deparam com um cenário de aprendizagem complexo [4]. Além de ser uma ferramenta poderosa para o ensino de pensamento computacional [5]. Por esse motivo, os responsáveis pelos currículos escolares estão cada vez mais interessados em incorporar abordagens STEM na educação formal [6].

Porém, atualmente coexistem diferentes conceitos de currículo STEM [7], e apenas alguns estudos teóricos possuem estruturas para o ensino [8]. Além disso, muitas vezes as diretrizes didáticas não são claras o suficiente para os professores [9], o que dificulta a implementação, já que a abordagem STEM implica diversos desafios para os professores. A conexão entre conhecimentos STEM e problemas da vida real não é uma tarefa fácil, e exige que o professor prepare aulas que permitam aos alunos compreender essa integração [10]. Muitos professores consideram que requer muito tempo e esforço para projetar aulas com caráter STEM [9], e por esse motivo poucas escolas possuem uma educação STEM operacionalizada [11].

Uma forma de resolver esse problema é criar um currículo estruturado de educação STEM, pois os professores são mais propensos a integrar esses conceitos quando os materiais curriculares atendem essa necessidade de forma clara e direta [12].

1.2 Objetivos

Tendo isso apresentado, o objetivo do presente trabalho é desenvolver um novo material educacional usando uma abordagem híbrida que combina construcionismo, aprendizagem baseada em projetos e instrucionismo, permitindo uma experiência de aprendizagem abrangente que auxilie na implementação do currículo STEM nas escolas. O desenvolvimento desse novo material educacional envolveu a combinação dessas três abordagens de forma a maximizar seus benefícios. O material é projetado para ser interativo e envolvente, com foco na aprendizagem prática e na aplicação no mundo real. Ele incorpora uma gama de diferentes atividades de aprendizagem, incluindo tarefas baseadas em projetos,

instrução direta e oportunidades de aprendizagem auto dirigidas.

Para o desenvolvimento deste material será utilizada a metodologia investigação em Design Educacional IDE, que irá estruturar o processo de desenvolvimento.

Para testar a eficácia desse novo material educacional, foi implementado em sala de aula, com uma turma do 9º ano, e duas turmas do 12º ano como parte das aulas regulares de Tecnologias da informação e comunicação (TIC), tendo uma duração total de 12 horas-aula. Os resultados de aprendizagem foram medidos usando testes padronizados por meio de métodos qualitativos e quantitativos, pelos alunos e professores, afim de demonstrar a eficácia do material e para poder alimentar o ciclo da IDE com os *feedbacks*.

Este material consiste em quatro projetos, com dificuldade crescente, que seguem a estrutura mostrada na Figura 1.1.

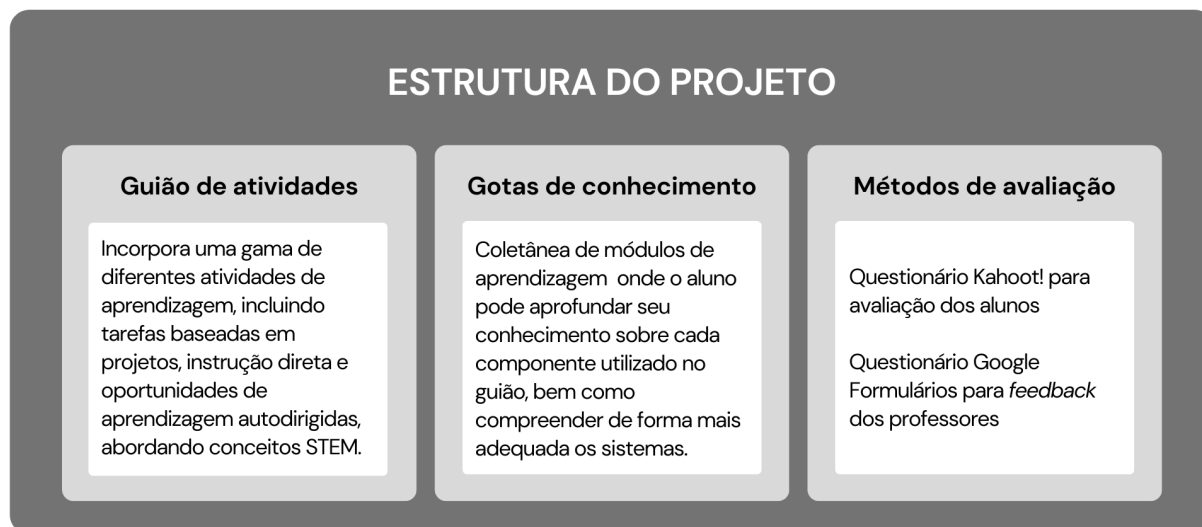


Figura 1.1: Descrição das partes da estrutura de cada projeto

1.3 Estrutura do documento

Este trabalho foi dividido em seis capítulos e 12 apêndices.

No segundo capítulo é feita, de forma detalhada, a revisão bibliográfica que engloba conceitos de educação tecnológica, métodos de ensino e design de investigação educacional.

No terceiro capítulo contém a descrição da implementação da primeira fase da IDE, explicando a investigação preliminar do problema e contexto escolhido.

Já no quarto capítulo é apresentado o desenvolvimento da segunda fase da IDE, detalhando o desenvolvimento e prototipagem dos materiais.

No quinto capítulo apresenta-se a terceira fase da IDE, demonstrando a implementação dos materiais em sala de aula, bem como seus resultados.

No sexto capítulo são descritas as conclusões e recomendações para trabalhos futuros.

E por fim, nos apêndices foram colocados os guiões e as gotas de conhecimento.

Capítulo 2

Revisão da literatura

Este trabalho de mestrado tem como objetivo utilizar de métodos de design educacional, afim de desenvolver um material de suporte para educação STEM dentro do ensino secundário. Para isso, será feita uma revisão da literatura sobre educação tecnológica e educação STEM, assim como robótica educacional. Adicionalmente será revista a forma de aprendizagem do indivíduo e os diferentes métodos de ensino.

2.1 Educação tecnológica

A tecnologia é inerente ao nosso dia a dia, fazendo com que diversas tarefas básicas necessitem do uso da mesma. Cada vez mais a tecnologia influencia a sociedade do conhecimento, sendo de grande importância para inúmeros contextos e finalidades. Dessa forma, é interessante que a tecnologia seja abordada como um aliado da aprendizagem, trazendo essa interação para o ambiente escolar [13].

O ensino tecnológico foi designado como essencial em 2015, onde também foi discutido as necessidades sociais de haver uma formação que permita a resolução de problemas reais no futuro, utilizando ferramentas que estejam à disposição. Assim sendo, o aluno estará acostumado com o uso dessas ferramentas e, quando o problema surgir, terá competência para utilizá-las de forma natural e orgânica [14].

A maior motivação da inclusão de uma educação tecnológica é desenvolver capacidades

que não precisam necessariamente ser aplicadas à tecnologia, como: criar, desenvolve soluções com base em um problema real, utilizar a lógica, ter capacidade de discernimento, resiliência, protagonismo, criatividade, entre outros [15].

Diversas vezes ocorre uma tensão entre as visões da educação tecnológica, a educação científica destinada a produzir a nova geração de cientistas, e a educação científica destinada a suprir o conhecimento e compreensão da ciência para que a nova geração se tornem capazes de tomar decisões democráticas [16]. Porém muitos movimentos e organizações apoiam a combinação dos dois [17].

2.1.1 Educação STEM

Cada vez mais a educação STEM ocupa espaço na literatura de investigação, o que converge para um interesse em incorporar abordagens STEM nas matrizes curriculares da educação formal [17]. Esse movimento se dá pelo fato dessa abordagem ser considerada eficiente para desenvolver a alfabetização científica dos cidadãos, e conseqüentemente aumentar o número de jovens a seguir o ramo da tecnologia no final da escolaridade obrigatória [18].

Estudos indicam que as crianças do ensino fundamental geralmente apresentam um maior interesse no conteúdo STEM, porém se enxerga um padrão de declínio de interesse à medida que a idade aumenta, e para resgatar o interesse se faz necessária a implementação de educação STEM no ensino secundário [19].

A educação STEM exige um nível de integração mais alto do que apenas o tratamento das quatro áreas separadamente. Pode inclusive ser considerada uma nova disciplina, a qual é uma forma de instrução educacional transversal, que tem a capacidade de desenvolver *hard skills* e *soft skills* [20].

Para a aplicação da educação STEM, pode ser formuladas atividades de robótica que trazem foco em várias, senão todas as quatro áreas STEM. Onde a robótica educacional é discutida como uma ferramenta poderosa para o ensino de pensamento computacional, codificação e engenharia [21].

Em consonância com a educação STEM, existe também a educação STEAM, que inclui os mesmos componentes do STEM, mas acrescenta a letra A para Artes (*Arts*). A educação STEAM incorpora elementos das artes e design ao currículo STEM, reconhecendo a importância da criatividade, imaginação e pensamento crítico no processo de inovação. Ela busca integrar as disciplinas STEM com as artes visuais, música, teatro, dança e outras formas de expressão artística [22].

2.1.2 Robótica educacional

Os primeiros *kits* de robótica voltados para a educação foram criados por Seymour Papert, em 1967, para ajudar crianças e compreenderem o funcionamento de pequenos robôs criados por Grey Walter [23].

Esses robôs se assemelhavam a tartarugas e tinham a função de seguir fontes de luz e, caso colidissem com algum obstáculo, afastavam-se de imediato. O modo de programação era pensado para computadores com menos capacidade de processamento. Com o tempo, os computadores avançaram em tecnologia e permitiram o surgimento de uma tartaruga virtual para a simulação da programação. Desse modo, surgiu a linguagem “Logo”, que ensinava as crianças a programarem os movimentos das tartarugas e que, posteriormente, seria integrado em componentes da Lego (como blocos, engrenagens, motores etc.) para que as crianças continuassem com o processo de aprendizagem e desenvolvessem sistemas programados [23].

Apesar de apresentar uma certa limitação, a Logo mostrou-se uma forma eficaz e bem difundida de transmissão do conhecimento de programação para as crianças [24]. Após essa primeira empreitada, surgiram vários outros métodos de difusão do conhecimento, segundo Gonçalves (2007). Entre eles, estão: o Tijolo Logo Brick, o Red *Brick*, o Lego *Mindstorms*, o *software cricket*, a Placa *GoGo* e afins [23].

Todos os sistemas descritos têm alguns objetivos em comum que tentam ser contemplados pela proposta inserida no trabalho. Com a Robótica Educacional fazendo parte do projeto pedagógico da instituição de ensino que contrata o serviço. Os objetivos são:

- Desenvolvimento do trabalho em grupo;
- Desenvolvimento de capacidade de resolução de problemas;
- Desenvolvimento da autonomia, a fim de capacitar o indivíduo a se posicionar, saber argumentar e participar na tomada de decisões;
- Desenvolvimento de pensamentos lógicos e capacidades matemáticas, trabalho em grupo, planejamento de projetos e noção espacial;
- Promover o desenvolvimento de projetos interdisciplinares, integrando conceitos de diversas áreas.

A robótica educacional se mostrou ser uma ferramenta eficaz de aprendizagem, principalmente quando se junta à aprendizagem baseada em projetos, onde STEM, programação e pensamento computacional podem ser englobados no mesmo projeto [21]

Assim sendo, o indivíduo que aprende, desenvolve capacidades múltiplas capaz de consolidar a aprendizagem e atingir o Nível de Conhecimento Real. Um dos níveis do desenvolvimento intelectual das crianças, como será visto a frente.

2.2 Forma de aprendizagem do indivíduo

O ser humano tem uma capacidade de aprendizagem que fascina diversos cientistas e rege estudos cada vez mais aprofundados sobre o assunto. Eles tentam caracterizar como ocorre essa aprendizagem de forma rápida e eficaz. Paralelo a esses estudos, investigações são feitas para possibilitar que máquinas tenham o mesmo comportamento [25].

Com base nisso, será apresentada, brevemente, a teoria vista por Lev Vygotsky, a qual tem como objetivo tentar explicar como é dado o desenvolvimento de aprendizagem no ser humano.

O conhecimento humano é fortemente moldado pelo meio em que ele está inserido, sendo esse processo influenciado pelo ambiente e pelos indivíduos que convivem nesse ambiente. Isso pode ser chamado de aprendizagem social [26]. A aprendizagem social é

de suma importância para a atualização das informações do indivíduo. Ela é responsável pela constante mudança na base de conhecimento dele. Em vista disso, Vygotsky elaborou uma tese que diz que o desenvolvimento cognitivo do ser humano se dá com as interações que ele faz com o meio social, com a cultura e com os outros membros da cultura.

Para esse autor, a aprendizagem se dá pelo alcance de ir em direção a um estado final melhor do que aquele em que se encontrava anteriormente. A aprendizagem se consolida e se dá por completo quando o indivíduo é colocado diante de uma situação em que ele precisa pedir ajuda, uma situação inesperada ou uma caótica. Essa situação precisa ser externalizada em busca de apoio. Com isso, o indivíduo consegue obter o controle da situação e adquirir novos conhecimentos, a partir de três fases importantes [27]:

- Planeamento: na qual planeia os próximos passos com base nas informações que tem;
- Inibição: nessa parte, as ideias anteriormente pensadas no planeamento são filtradas, para que o indivíduo possa prosseguir com suas ações e para que sejam factível dentro de seu ambiente real;
- Local de controle: o local de controle é a fonte de informações do indivíduo. Pode ser definido como o subsídio que é necessário para regular o pensamento. O local de controle mostra as opções que podem ser plausíveis para o indivíduo com base no que foi pensado no planeamento e na inibição.

2.2.1 Níveis de desenvolvimento

São definidos dois níveis de desenvolvimento da análise da relação entre o “processo de desenvolvimento” e a “capacidade de aprendizagem”, como mostra a Figura 2.1.

- Nível de Desenvolvimento Real (Nível de Desenvolvimento Real (NDR)): Tudo que uma pessoa é capaz de fazer por si própria pertence a esse nível. Ou seja, capacidades que podem ser consideradas adquiridas ou formadas. Esse nível é o parâmetro para

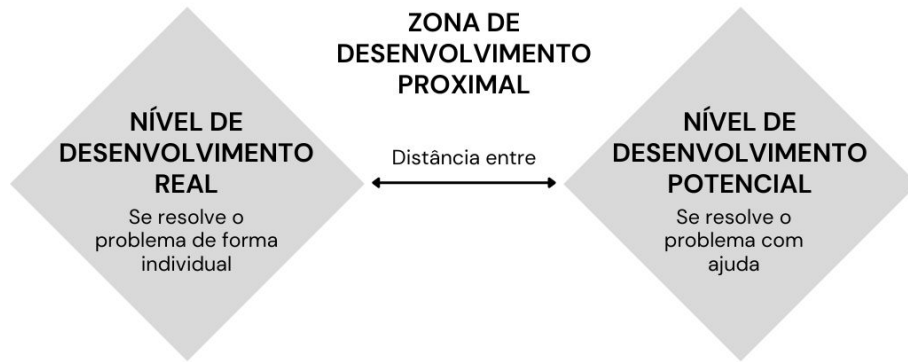


Figura 2.1: Níveis de desenvolvimento. Adaptada de [28]

medir o quociente de inteligência (Quociente de Inteligência (QI)). Os conhecimentos estão consolidados [26];

- Nível de Desenvolvimento Próximo ou Potencial (Nível de Desenvolvimento Potencial (NDP)): Neste nível, estão as atividades e as funções que uma pessoa consegue fazer apenas com ajuda de outros que são considerados mais capazes. São as funções em processo de amadurecimento.

A distância entre os dois níveis e o espaço que existe entre elas, o autor chama de Zona de Desenvolvimento Proximal, que pode ser desenvolvida com soluções de problemas que estejam sob a orientação de um indivíduo que seja mais capacitado para tal função.

2.2.2 Importância da brincadeira na aprendizagem

A brincadeira na vida da criança tem um fator e uma importância inegável, mas para [26], a brincadeira não é uma característica relacionada apenas as crianças, e sim uma construção social que foi apreendida nas relações interpessoais, tornando a brincadeira

uma forma de aprendizagem sociocultural. Esse aprender brincando tem dois lados importantes: o desenvolver de comportamentos lúdicos dos adultos para envolver a criança e o processo de descobrimento e curiosidade das crianças.

As brincadeiras baseiam-se em espaços nos quais podem criar, inovar e experimentar, seguindo regras previamente estabelecidas pelos participantes. Esse ambiente, segundo Vygotsky, cria uma vontade na criança cada vez maior de aprender, superar desafios e aprender sobre seus próprios limites. Todo esse ambiente facilita a criação de Zona de Desenvolvimento Proximal [29].

2.2.3 Papel da escola

Para se criar um ambiente de ensino e aprendizagem, alguns fatores são essenciais e indispensáveis, como o indivíduo que aprende, o que ensina e a interrelação entre essas duas partes. Para a Pedagogia, a avaliação é feita com base no que cada aluno é capaz de desenvolver sozinho. Vygotsky acha isso um erro e diz que o aluno deve ser avaliado com base em sua capacidade de interagir e colaborar com o próximo tomando com base seus conhecimentos prévios, e juntamente a isso saber trabalhar e reagir a informações e a instruções sendo dadas [26]. Esse tipo de avaliação permite que a parte que está aprendendo desenvolva a autonomia de que precisa, para que possa controlar as próprias atividades. É papel da escola promover o ambiente propício para haver o desenvolvimento integral do aluno. Bem como disponibilizar de todas as ferramentas necessárias [30].

2.3 Métodos de ensino

Um dos maiores desafios percebidos para o andamento deste trabalho foi a escolha do método de ensino que será utilizado para nortear os projetos. Foram escolhidos dois pilares que serão seguidos ao longo do trabalho: O construcionismo, o qual tem o aluno como protagonista no processo da construção do conhecimento [31] e a aprendizagem baseada em projetos, uma metodologia ativa de ensino.

2.3.1 Instrucionismo

O instrucionismo é um método de ensino que se concentra na transmissão do conhecimento do professor para o aluno. Nessa abordagem, o professor é visto como a figura de autoridade que transmite conhecimento, e os alunos são receptores passivos de informações [32]. O objetivo do instrucionismo é garantir que os alunos aprendam o que precisam saber por meio da memorização mecânica e da repetição de informações. Este método é baseado na crença de que o professor sabe mais e que os alunos aprendem melhor ouvindo e seguindo as instruções [33].

O instrucionismo teve um impacto significativo na educação, particularmente no ambiente de sala de aula tradicional. Essa abordagem de ensino existe há séculos e tem sido amplamente utilizada em muitos sistemas educacionais em todo o mundo. O foco na memorização e na repetição tem sido benéfico para os alunos que precisam se lembrar de informações rapidamente, como no caso de aprender fatos matemáticos básicos ou fórmulas científicas. No entanto, também foi criticado por ser muito rígido e não permitir criatividade ou pensamento crítico [34].

Um ponto positivo do instrucionismo é que ele fornece um ambiente estruturado para a aprendizagem. Os alunos sabem o que se espera deles e o que precisam aprender. Além disso, o instrucionismo é eficaz para ensinar capacidades fundamentais, como leitura, escrita e aritmética. No entanto, um de seus pontos fracos é que pode não promover o pensamento crítico ou a criatividade. Os alunos podem se esforçar para aplicar o que aprenderam em situações do mundo real e podem não conseguir se adaptar a novos desafios [35].

2.3.2 Construcionismo

Na teoria construtivista de Jean Piaget, a aprendizagem é construída com a experiência. Usando a reflexão da experiência, os indivíduos se tornam capazes de construir o conhecimento e as compreensões do mundo a sua volta [36].

O construcionismo de Seymour Papert, tem como base a teoria construtivista de Piaget, com o qual compartilha a ideia de que a aprendizagem envolve construir estruturas de conhecimento. Além disso, esse autor defende que, para a aprendizagem ser verdadeiramente eficaz, os alunos devem aprender com base na concepção de objetos que sejam perceptíveis para o contexto do mundo real, para públicos reais, aprendendo com suas experimentações criativas [37].

A ênfase dada ao aluno como protagonista contrasta com o “instrucionismo”, que é a prática prevalecente acadêmica, na qual o conhecimento deve ser transmitido e absorvido de forma estática e não volátil. Porém, muitas investigações mostraram que essa forma de transmitir conhecimento não tem sido eficaz, já que os alunos retêm pouco das informações que foram transmitidas e apresentam dificuldade em transmitir o conhecimento adquirido [30].

A fim de aumentar o desenvolvimento individual, o construcionismo propõe que os professores atuem como facilitadores, treinando os alunos a procurarem por conhecimento. Isso é feito substituindo a aula com caráter puramente expositivo por uma aula colaborativa em que, por meio da exposição de problemas, os alunos são convidados a desenvolverem soluções para estes [37].

2.3.2.1 Aprendizagem baseada em projetos

O método construcionista proporciona ao aluno uma “aprendizagem ativa”, a qual permite fazer conexões entre diferentes áreas de conhecimento. A Aprendizagem Baseada em Projetos (NDP) é uma abordagem construcionista que tem o aluno como personagem principal, envolvendo-o em resolução de problemas reais. É estruturado para que os alunos elaborem e criem projetos destinados a abordar a resolução desses problemas que são apresentados pelo professor ou por eles mesmos [38].

A NDP pode ser definida pela utilização de projetos autênticos e realistas, baseados em uma questão, tarefa ou problema altamente motivador e envolvente, para ensinar conteúdos acadêmicos aos alunos no contexto do trabalho

cooperativo para a resolução de problemas. [...] A investigação dos alunos é profundamente integrada à aprendizagem baseada em projetos, e como eles têm, em geral, algum poder de escolha em relação ao projeto do seu grupo e aos métodos a serem usados para desenvolvê-los, eles tendem a ter uma motivação muito maior para trabalhar de forma diligente na solução de problemas.
[39]

Ao investigar um problema e projetar uma solução, os alunos são convidados a adquirir conhecimentos amplos e interconectados de princípios e conceitos-chave. Esse ambiente de desenvolvimento de projetos para resolver um problema estimula o esforço, a persistência e a autocrítica por parte dos alunos, além da atenção, do sentimento de pertencimento e de responsabilidade, da autoconfiança, do gerenciamento de tempo e da comunicação interpessoal. Dessa forma, os alunos podem elaborar suas próprias ideias e soluções, podendo aplicá-las e aprender fazendo [40]. 'Na ótica dos estudantes, tais métodos, por partirem de situações reais ou se aproximarem da realidade, estimulam o estudo constante, a independência e a responsabilidade do aluno' [41].

Tendo exposto os métodos que serão utilizados, a Tabela 2.1 traz a comparação direta dos métodos, com suas vantagens e desvantagens.

Tabela 2.1: Comparação dos métodos de ensino utilizados

Método	Objeto principal	Vantagem	Desvantagem
Instrucionismo	Professor	Amplamente utilizado Ambiente estruturado para aprendizagem	Não encoraja criatividade ou pensamento crítico
Construtivismo	Aluno	Promove aprendizagem ativa Desenvolve pensamento crítico	Desafio para alunos que não conseguem trabalhar de forma independente

2.4 investigação em design educacional

A investigação em design educacional é apropriada para desenvolver soluções baseada em investigação para problemas complexos na prática educacional ou para desenvolver ou validar teorias sobre processos de aprendizagem, ambientes de aprendizagem e afins. Essa estrutura de investigação abrange o estudo sistemático de projetar, desenvolver e avaliar intervenções educacionais, como: programas, processos de aprendizagem, ambientes de aprendizagem, materiais de ensino-aprendizagem, produtos e sistemas [42].

Segundo [43], as abordagens de investigação "tradicionais" dificilmente fornecem prescrições que são úteis para problemas de desenvolvimento em educação. Esse tipo de investigação em design educacional permite que os investigadores ajustem sistematicamente vários aspectos, de modo que cada ajuste necessário é visto como um tipo de experimentação, e por isso afirma que a investigação em design educacional é capaz de desenvolver soluções ótimas para o problema que está em contexto.

A investigação de design é um termo que pode variar dependendo dos objetivos e características da investigação, como: design de experimentos [44], estudos de design [45], investigação baseada em design [46], investigação de desenvolvimento [47] e investigação de engenharia [48].

2.4.1 Ciclos, fases e outras características

Independente do propósito da investigação em design educacional, existe um ciclo sistemático para que o processo de investigação seja efetivo, como demonstrado na Figura 2.2.

Os processos de design educacional devem seguir o processo sistemático de caráter cíclico: as iterações entre as atividades de análise, design e avaliação ocorrem até que seja alcançado o equilíbrio apropriado entre os ideais e o resultado final [49].

As três fases são descritas dessa forma:

- Fase 1 - Análise e exploração: onde ocorre a revisão da literatura, a avaliação das

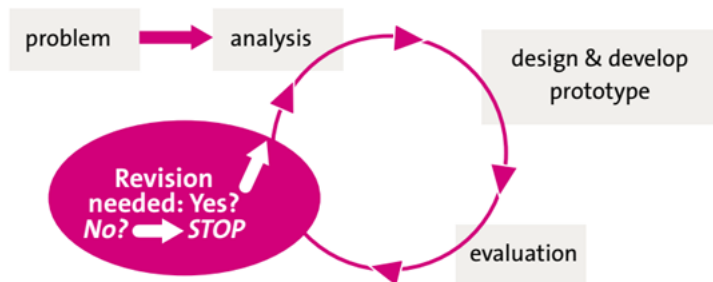


Figura 2.2: Iterações de ciclos sistemáticos [42]

necessidades, análise de contexto e a partir daí o desenvolvimento de uma estrutura conceitual para o estudo;

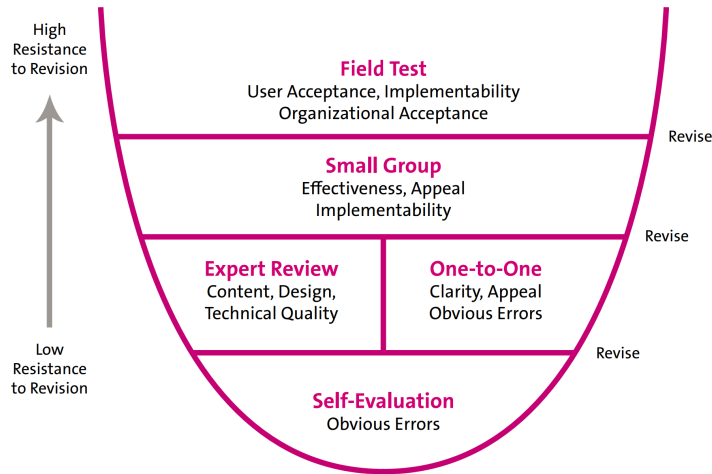
- Fase 2 - Desenvolvimento e prototipagem: nesta fase deve ser feito o desenvolvimento dos *designs* com base na investigação preliminar, os quais serão experimentados e revistos com base em avaliações formativas.
- Fase 3 - Avaliação e reflexão: Avaliações formativas para poder concluir se a solução atende as especificações e se os usuários-alvo estão dispostos a aplicá-lo em seu ensino. Essa fase resulta em recomendações para melhoria da solução.

2.4.2 Avaliações formativas

As avaliações formativas ocupam um espaço de grande importância na investigação em design educacional, pois é a avaliação voltada para a melhoria. Essa avaliação tem várias camadas, como demonstrado na Figura 2.3, desde mais informal nos estágios iniciais (autoavaliação, avaliação individual, revisão de especialistas), até a avaliação em pequenos grupos, afim de testar a praticidade e eficácia, e se possível um teste de campo completo (nível mais elevado) [50].

O projeto de investigação deve ser adaptável para mudar e evoluir em cada fase, levando em conta os *feedbacks* de cada uma das avaliações formativas. Essa capacidade de adaptação permite que a sinergia entre a prática e a investigação possa ser maximizada [42].

Figura 2.3: Camadas de avaliação formativa [42]



2.4.3 Metodologia adotada no projeto

Em conclusão, tanto o instrucionismo quanto o construcionismo têm seus pontos fortes e fracos. Embora o instrucionismo possa ser eficaz para ensinar capacidades fundamentais, o construcionismo pode ser mais adequado para desenvolver o pensamento crítico e as capacidades de resolução de problemas, por isso o material educacional proposto foi desenvolvido usando uma abordagem híbrida que combina construcionismo, metodologia baseada em projetos e instrucionismo. Essa abordagem permite uma experiência de aprendizagem abrangente que incorpora uma variedade de métodos de ensino diferentes, garantindo que todos os alunos tenham a oportunidade de se envolver com o material de uma maneira que atenda às suas necessidades individuais.

A metodologia da IDE citada anteriormente será seguida neste projeto, tendo a primeira fase consistindo na análise e exploração das necessidades dos alunos e dos métodos de ensino, bem como experimentação de atividades. Já a fase de desenvolvimento e prototipagem é dedicada ao desenvolvimento dos materiais com os métodos escolhidos na fase anterior e sua posterior aplicação em sala de aula. Por fim, a última fase analisa os dados das avaliações dos alunos e dos professores após a aplicação dos materiais em sala de aula.

Ao adotar essa metodologia, o projeto tem como objetivo criar um ambiente de aprendizagem inclusivo e envolvente que promova a aquisição de capacidades fundamentais e capacidades de pensamento crítico. A abordagem híbrida permite flexibilidade para atender a diferentes estilos de aprendizagem e preferências, garantindo que cada aluno possa maximizar seu potencial de aprendizagem. O uso de atividades baseadas em projetos aprimora as capacidades de resolução de problemas dos alunos e incentiva a participação ativa e a colaboração.

Em resumo, a metodologia adotada neste projeto combina os pontos fortes do instrucionismo e do construcionismo para criar uma experiência de aprendizagem abrangente e personalizada. Ao incorporar uma variedade de métodos e estratégias de ensino, o projeto visa fornecer aos alunos uma base sólida de capacidades, ao mesmo tempo em que desenvolve suas capacidades de pensamento crítico e competências de resolução de problemas.

Tendo dito isso, a Figura 2.4 mostra a estrutura final da IDE e os procedimentos para implementação do método para o desenvolvimento do currículo STEM.

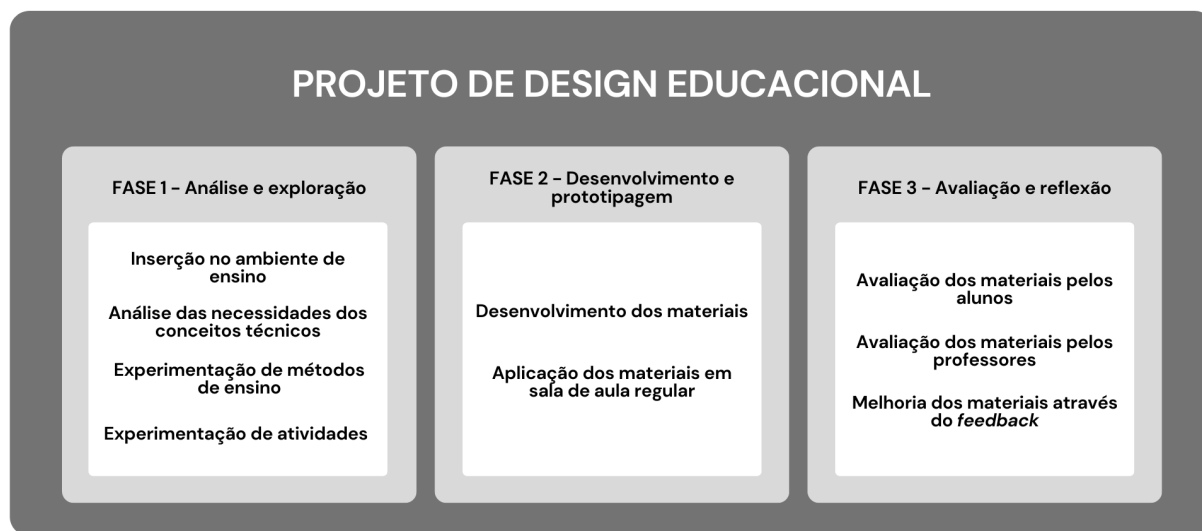


Figura 2.4: Estrutura geral utilizada na implementação da IDE

Capítulo 3

Fase 1: Análise e exploração

Como visto anteriormente, a primeira fase da IDE é dedicada a uma análise e exploração dos conceitos que estão sendo abordados na proposta. No caso do presente trabalho, que tem como objetivo desenvolver um currículo STEM para a educação de robótica e eletrônica no ensino secundário, essa fase foi utilizada para se inserir no contexto de ensino de robótica dentro de uma escola pública em Portugal, e entender quais são as necessidades e expectativas dos alunos quanto ao ensino STEM.

Para isso, escolheu-se participar voluntariamente de um clube de robótica já existente no Agrupamento de Escolas Miguel Torga, onde encontros semanais eram conduzidos destinados a alunos do 9º ao 12º ano. Foi construído um cronograma de 2 (dois) meses de duração total, o qual tinha como objetivo reunir informações de quais conceitos de eletrônica e programação são interessantes e efetivos para a construção do material, além de quais os melhores métodos de abordagem para o ensino.

O cronograma foi validado ao final da primeira participação no clube, feita no dia 02/03/2022, a qual teve um caráter de observação, onde inicialmente foi apresentado o projeto e o intuito geral do trabalho, mas tinha como objetivo entender qual era a dinâmica do ensino da robótica. A escola dispunha de um robô mBot, da marca Makeblock®, o qual segundo site da marca:

"é um robô de codificação STEM para iniciantes, o que torna o ensino e a

aprendizagem da programação de robôs simples e divertidos. Com uma chave de fenda e instruções passo a passo, as crianças podem construir um robô do zero e experimentar as alegrias da criação prática. À medida que avançam, eles aprenderão sobre uma variedade de máquinas robóticas e peças eletrônicas, entenderão os fundamentos da programação baseada em blocos e desenvolverão seu pensamento lógico e capacidades de design [51]."

Esse robô estava sendo utilizado como ferramenta de ensino para os alunos, e foram feitas algumas experiências neste dia, mas como os alunos já estavam avançados com os experimentos que o mBot fornecia, foi exposto pelos alunos e pela professora a necessidade de expandir e aprofundar os conhecimentos. A partir dessa informação foi pensado um cronograma que contivesse alguns experimentos com dificuldade crescente, os quais seriam expostos durante o clube para o desenvolvimento dos alunos. Na Tabela 3.1 encontra-se o cronograma.

Tabela 3.1: Cronograma de análise e exploração

	Data	Atividade	Dificuldade
Primeira parte	09/03/2022	Blink	Iniciante
		LED RGB com botão	Iniciante
	16/03/2022	Sinalizador SOS	Iniciante
		Sensor LDR	Iniciante
	22/03/2022	Workshop	Iniciante
Segunda parte	06/04/2022	Codificador SOS	Intermediário
	20/04/2022	Sistema de segurança	Intermediário
	27/04/2022	Sensor de ré	Intermediário
		Sinaleiro	Intermediário
	04/05/2022	Contador display 7 segmentos	Avançado
	11/05/2022	Display LCD	Avançado

O cronograma foi considerado com a finalidade de implementar e testar alguns métodos de ensino que foram vistos na revisão da literatura, bem como as plataformas sugeridas,

para que ao final dessa fase fosse possível estruturar de forma mais adequada o material que será desenvolvido na Fase 2.

3.1 Análise e exploração de métodos instrucionistas

Na primeira parte do cronograma foram aplicadas atividades consideradas de nível iniciante, com materiais desenvolvidos para serem seguidos pelos alunos através de uma aula expositiva via *slides*. Essas atividades foram preparadas para ter o professor como elemento principal e indispensável, onde ocupa a posição de porta-voz do conhecimento, e o aluno como recetor direto.

As quatro atividades dessa primeira parte seguiram a seguinte estrutura:

- Apresentação do sistema a ser reproduzido;
- Apresentação dos materiais necessários para a atividade;
- Apresentação dos conceitos técnicos abordados na atividade;
- Estrutura do *Hardware* na simulação;
- Programação no *Software* de simulação;
- Teste na simulação;
- Montagem do *Hardware* no sistema real;
- Estrutura do código no *Software* Arduino IDE;
- Teste final no sistema físico.

Como dito anteriormente, os alunos do clube de robótica trabalhavam com um material da Makeblock®, o qual fundamentou o conhecimento de programação baseada em blocos, e por esse motivo, nessa fase inicial não foi necessário fazer a introdução a esta linguagem de programação, o que abriu a oportunidade de explorar mais profundamente a programação escrita em C++.

A seguir, a Figura 3.1 mostra alguns dos *slides* de uma das atividades para exemplificar a estrutura e estética.

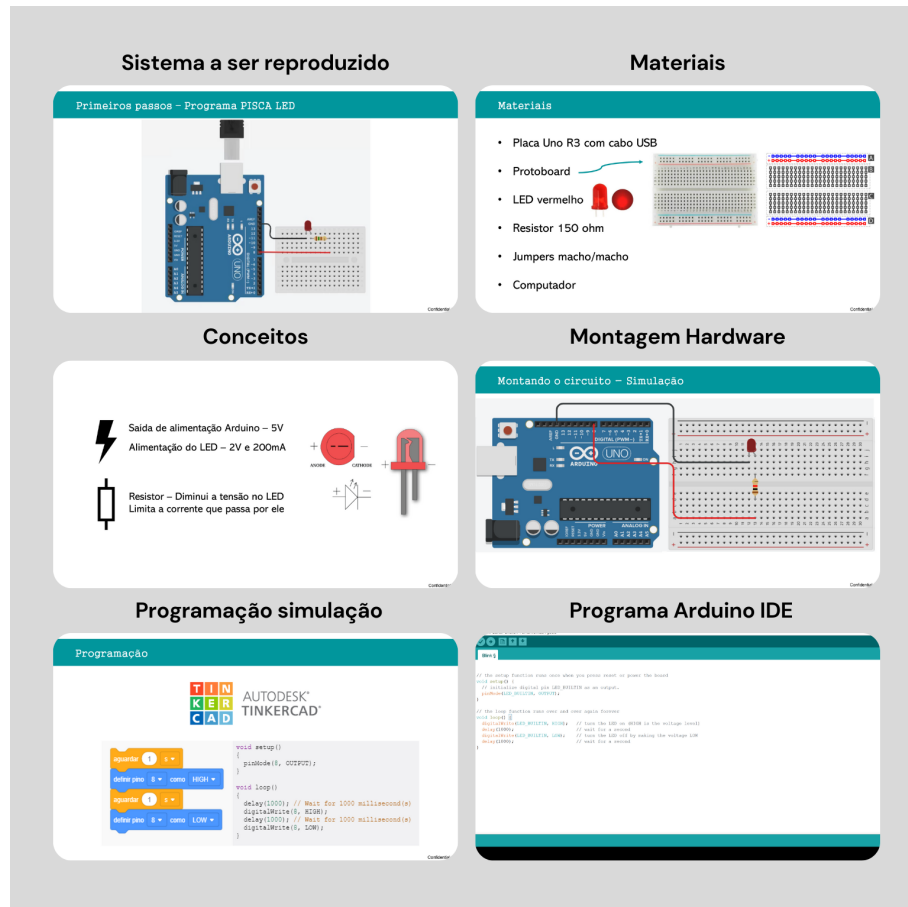


Figura 3.1: Exemplo atividade "Blink"

Finalizadas as atividades da primeira fase no clube de robótica, foi aberto para o corpo docente e discente um *Workshop* de Introdução à Robótica, já previsto no cronograma, que teve a finalidade de validar quais conceitos deveriam ser passados nesta fase inicial para aqueles que ainda não tinham tido contacto com eletrônica e programação anteriormente.

Ao final dessa fase, após a implementação das atividades no clube de robótica e o *workshop* com alunos e professores, pode-se começar a estruturar quais conceitos e técnicas deveriam ser levados em conta para desenvolver os materiais na Fase 2 da IDE.

3.2 Análise e exploração de métodos construcionistas

A segunda parte do cronograma, diferente da primeira que teve um caráter instrucionista, foi dedicada a tornar o aluno ativo em sua aprendizagem, utilizando de técnicas do construcionismo de Seymour Papert, que permitem que o aluno se tornem protagonistas na construção do próprio conhecimento. Nesta etapa os professores e instrutores atuam como facilitadores, e os alunos como elemento central, criando um ambiente colaborativo.

Tendo isso dito, os experimentos foram projetados para testar o impacto de diferentes abordagens construcionistas nos resultados de aprendizagem dos alunos. As atividades no cronograma foram idealizadas para que fossem realizadas com a aprendizagem baseada em projetos, ou seja, foi estruturado para que os alunos elaborassem soluções para os projetos apresentados, com as ferramentas disponibilizadas, sempre indo em direção ao conhecimento da eletrônica e programação.

Em cada uma das atividades propostas, no início foi exposto qual o contexto do problema e qual deveria ser o resultado da solução. Posteriormente quais as competências seriam trabalhadas em cada atividade. A Figura 3.2 mostra um exemplo de uma atividade que foi aplicada na segunda parte do cronograma.

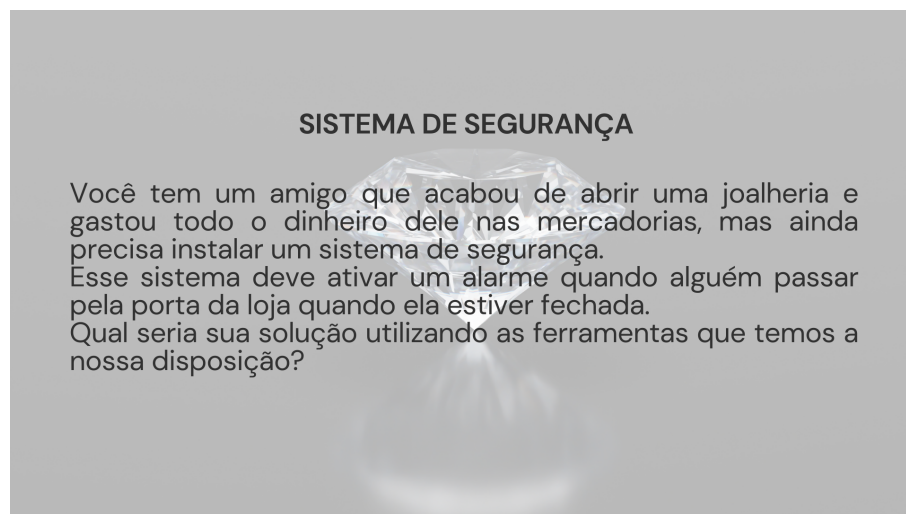


Figura 3.2: Exemplo atividade "Sistema de segurança"

Com essa dinâmica os alunos tiveram a liberdade de investigar informações *online*

sobre as ferramentas disponibilizadas, e desenvolver uma solução que atendesse os requisitos do problema. Durante a aplicação dessas atividades o professor atuou apenas como facilitador quando alguma dúvida surgia, interferindo o mínimo possível no processo de construção de conhecimento, por esse motivo não foi estruturado um material de apoio para esses projetos, apenas o contexto do problema proposto.

Ao final da segunda fase do cronograma, após a implementação de técnicas construcionistas, juntamente com as conclusões da primeira parte do cronograma, pude identificar os métodos construcionistas mais eficazes e incorporá-los ao design do material didático e finalizar a estrutura e conteúdo do currículo que foi desenvolvido na Fase 2 da IDE, conforme a Figura 3.3.

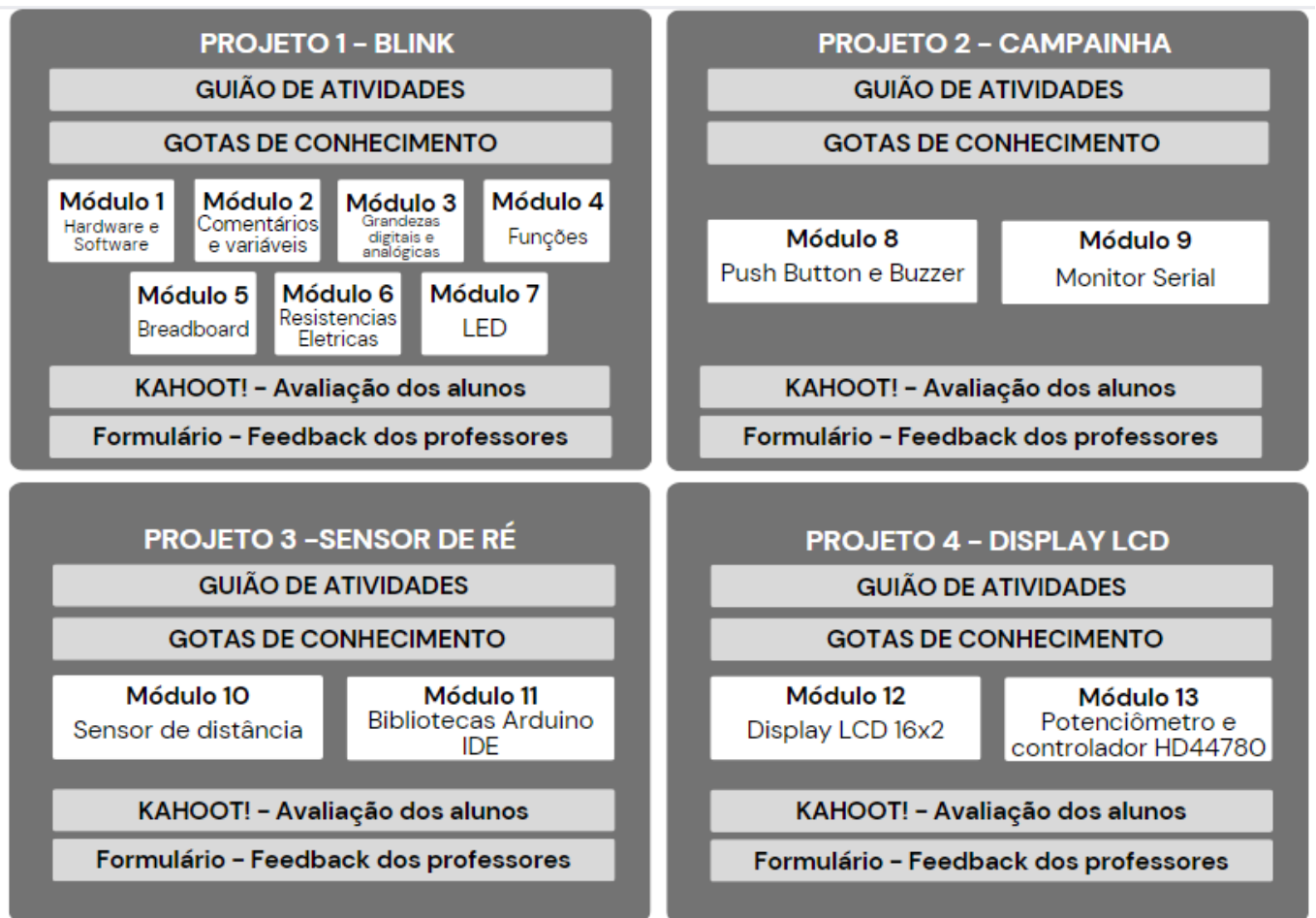


Figura 3.3: Estrutura e conteúdo de cada projeto do material

A estrutura do material consiste em quatro componentes principais:

- Guião de atividades: o guião é um documento que fornece instruções e desafios que envolvem o aluno nas atividades voltadas para a aprendizagem em robótica e eletrónica, criado para ajudá-los a executar as atividades. Foi ofertado ao aluno de forma digital, através da plataforma disponibilizada pela escola.
- Gotas de conhecimento: assim como o guião de atividades, as gotas de conhecimento também são documentos que foram ofertados aos alunos de forma digital. As gotas de conhecimento foram concebidas como parte do processo de adaptação do projeto, para atender alunos com diferentes níveis de desenvolvimento académico. O objetivo dessas gotas é aprimorar a compreensão dos conceitos básicos, garantindo assim que os alunos que possuem familiaridade prévia com o assunto permaneçam comprometidos e motivados. Além disso, as gotas de conhecimento servem para avivar aspetos mais técnicos dos projetos, atendendo aos interessados em se aprofundar no assunto.
- Avaliação dos alunos: após a conclusão de cada projeto, o objetivo principal do trabalho é determinar o sucesso do processo de aprendizagem dos alunos. Para atingir este objetivo, era indispensável criar métodos de avaliação que pudessem aferir o progresso da aprendizagem, pensamento e raciocínio dos alunos. Tendo dito isso, a avaliação dos alunos consiste em um questionário realizado via Kahoot! que permite a verificação do processo de aprendizagem.
- *Feedback* dos professores: os professores também foram envolvidos no processo de avaliação, onde foram convidados a dar seu *feedback* através de um formulário da @Google.

O processo de seleção dos conteúdos de cada projeto envolveu vários critérios, entre eles a disponibilidade de componentes para implementação. A ênfase foi dada aos componentes que são facilmente acessíveis e baratos. Além disso, foram priorizados componentes que podem operar somente na fonte de tensão do @Arduino, considerando seu baixo consumo

de energia. Os componentes escolhidos para os projetos também foram selecionados por sua simplicidade, sem exigir circuitos mais complexos para seu funcionamento. Por fim, a flexibilidade foi considerada no processo de seleção, com componentes adaptáveis para uso em arranjos de circuitos simples e complexos.

Além disso, a curva de aprendizagem da faixa etária alvo também foi considerada. Na fase inicial da IDE, foi reconhecido que era necessário aumentar o nível de dificuldade com etapas facilmente alcançáveis para acomodar os alunos que podem ter dificuldades com conceitos mais avançados. Como resultado, os quatro projetos foram cuidadosamente selecionados para manter um aumento gradual de dificuldade, sem apresentar saltos significativos entre eles. Para isso, cada projeto se baseia no anterior, com o primeiro projeto envolvendo a ativação de um atuador, o segundo integrando um sensor e um atuador, o terceiro apresentando um sensor um pouco mais complexo e o projeto final incluindo um atuador mais avançado. Esse aumento gradual de complexidade ajudou a garantir que os alunos não fossem desmotivados, ao mesmo tempo em que forneceu desafios acessíveis que aprimoraram sua experiência de aprendizagem.

A fase inicial da IDE decorreu na Escola Secundária Miguel Torga situada na cidade de Bragança, em Portugal, onde participaram alunos dos 12 aos 15 anos. Especificamente, os alunos estavam matriculados no 9º ou 12º ano e, como resultado, os projetos foram adaptados para atender aos requisitos e interesses desse grupo demográfico específico, podendo ser adaptados para outras realidades na mesma faixa etária.

A próxima secção será dedicada a explicar detalhadamente o conteúdo de cada parte da estrutura.

Capítulo 4

Fase 2: Desenvolvimento e prototipagem

Finalizada a Fase 1 no clube de robótica, a IDE segue para a Fase 2, que como visto anteriormente é onde ocorre o desenvolvimento e prototipagem do materiais propostos na estrutura vista no capítulo anterior. Nesta fase um novo material educacional foi desenvolvido usando uma abordagem híbrida que combina construcionismo, metodologia baseada em projetos e instrucionismo. Essa abordagem permite uma experiência de aprendizagem abrangente que incorpora uma variedade de métodos de ensino diferentes, garantindo que todos os alunos tenham a oportunidade de se envolver com o material de uma maneira que atenda às suas necessidades individuais.

O material inclui uma variedade de atividades interativas, como construir modelos físicos ou trabalhar com simulações virtuais, que permitem aos alunos construir ativamente sua compreensão do conceito ou ideia apresentadas.

Nas secções seguintes serão abordados os quatro componentes do Projeto 1 - *Blink* (guião de atividades, gotas de conhecimento, avaliação dos professores e *feedback* dos alunos), para exemplificar como foi desenvolvida cada parte da estrutura dos projetos. Será explicado detalhadamente os aspetos de cada componente, bem como seu conteúdo e forma de construção.

4.1 Guião de atividades

4.1.1 Introdução

Nesta parte inicial do projeto é feita a contextualização dos conceitos que serão aprendidos, o nível de dificuldade do projeto, sendo dividido entre iniciante, intermediário e avançado. Bem como apresenta as competências pretendidas de aprendizagem, e a lista de materiais utilizados para o projeto. A Figura 4.1 demonstra como é a estrutura da página de introdução;

Iniciante
 Intermediário
 Avançado

BLINK - PISCANDO MEU PRIMEIRO LED

COMO UTILIZAR OS EXEMPLOS PRONTOS AO NOSSO FAVOR

Neste projeto podemos verificar e aprender alguns conceitos básicos que irão nos acompanhar por toda nossa caminhada pelo mundo da robótica! Vamos aprender a utilizar o LED incorporado na placa e no final levaremos mais adiante e teremos um LED externo piscando no ritmo que escolhermos!

Vamos lá?

COMPETÊNCIAS QUE VOCÊ VAI APRENDER COM ESSE PROJETO:

Hardware	Software
<ul style="list-style-type: none">• Funcionamento LED• Resistência• Funcionamento Protoboard	<ul style="list-style-type: none">• Como carregar o sketch na placa• Criação de variáveis• Construção de laço

Você vai precisar de:

- 1 placa Arduino UNO
- 1 LED vermelho 5mm
- Resistência 150 ohms e ¼ W
- Protoboard (Breadboard)

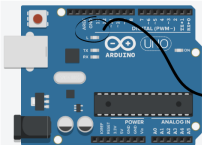
Figura 4.1: Página introdutória do guião

4.1.2 Simulação

Com os estudos preliminares feitos na Fase 1, viu-se a necessidade de implementar a simulação nos projetos, já que os alunos estariam, em sua maioria, tendo o primeiro contacto com a robótica, e projetar soluções computacionais-eletrónicas é uma tarefa complexa pela exigência de programação e cautela quanto aos dispositivos físicos comprometidos na implementação do circuito eletrónico final, risco que desaparece com o uso de simuladores. Dessa forma, após feita a introdução, segue-se para a parte da simulação, onde contém uma série de instruções que permitem que o aluno consiga efetuar a simulação no Tinkercad®, como mostra a Figura 4.2. Essa plataforma permite que a programação seja feita em blocos, facilitando que alunos que nunca tiveram contacto com programação possam realizar o projeto sem maiores dificuldades.


PARTE 1 - Simulação do LED na placa

Você sabia que a placa Arduino já vem com um LED dentro dela? E além disso, podemos controlar como ele acende ou apaga? Vamos aprender juntos!



Posição do LED incorporado na placa!


Vamos primeiro simular no TINKERCAD (<https://www.tinkercad.com/>) para ver como funciona!
Para isso vamos precisar somente da placa Arduino UNO, e colocar o seguinte código para rodar:



O bloco de controle "para sempre" caracteriza nosso *loop*, e o código dentro do *loop* será rodado para sempre nesse caso.

Neste código o LED incorporado está recebendo a informação para ligar, aguardar 1s e desligar.

Pronto! Agora é só iniciar a simulação e você verá o LED incorporado piscando!

 **PENSE SEMPRE ALÉM!**

O que acontece se aumentar o tempo que o LED deve aguardar para acender e apagar? Faça alguns testes para ver a diferença e discuta com os seus colegas os resultados!

Figura 4.2: Página de instruções para simulação

4.1.3 Sistema físico

Com a simulação completa, o aluno é encorajado a seguir para o próximo passo, onde é explicado de forma detalhada como fazer a montagem no sistema físico, bem como sua programação e teste, como mostrado na Figura 4.3;

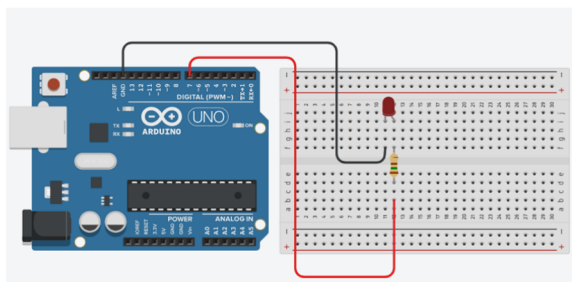
PARTE 3 - Piscar um LED externo

Para esse próximo passo iremos utilizar os materiais descritos no início do projeto! Então pegue seus equipamentos e vamos juntos!

HARDWARE

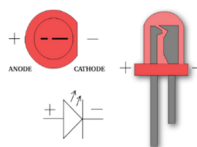
Para incluirmos um LED externo precisamos primeiro aprender alguns conceitos básicos de eletrônica, não queremos queimá-lo, certo?

A figura mostra o esquema de ligação que iremos utilizar para o LED externo:



Para os LEDs acenderem, é necessário passar corrente elétrica por eles, e para isso o LED tem uma maneira correta de ligação, pois ele só acende quando a corrente elétrica corre no sentido certo.

No LED há dois terminais: o polo positivo, também chamado de ânodo (terminal mais comprido) e o polo negativo, também chamado de catodo (terminal mais curto).



Para que funcione corretamente, a corrente elétrica deve passar primeiro pelo polo positivo, porém temos que cuidar para não passar muita corrente (correndo o risco de queimá-los), nem pouca corrente (correndo o risco de não acendê-los), para isso utilizaremos resistências limitadoras. Assim temos nosso diagrama esquemático de ligação do LED:

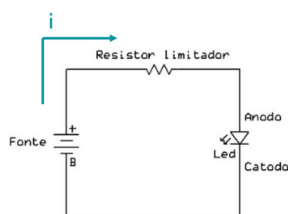


Figura 4.3: Página de explicação de montagem do sistema físico

4.1.4 Atividades colaborativas

Com o intuito de instruir o aluno a aprofundar mais o conhecimento de programação, as atividades colaborativas, chamadas de "Vamos entender juntos", traz explicações detalhadas sobre a estrutura do código, afim de consolidar o conhecimento da programação, como mostrado na Figura 4.4;

Vamos entender juntos a estrutura do código?

Diferente da programação em blocos que fizemos, esse código está escrito na linguagem de programação C++, a linguagem utilizada na plataforma Arduino.

O código base sempre deve ter duas funções, a função *setup* e a função *loop*.

FUNÇÃO SETUP

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
```

A função *setup* roda sempre que o botão reset é acionado, ou quando a placa com o código é ligada. Nela buscamos inicializar as variáveis que serão utilizadas no projeto. No caso do projeto Blink, declaramos que nosso LED incorporado está sendo visto como uma saída.

Para isso foi usado a função `pinMode(LED_BUILTIN, OUTPUT);`

Função que configura o pino

Pino (número ou nome do pino que deseja configurar)

Modo de Entrada ou Saída (INPUT ou OUTPUT)

Dessa forma configuramos nosso LED incorporado (LED_BUILTIN) para ser um atuador (OUTPUT), já que ele deverá acender e apagar conforme o código.

FUNÇÃO LOOP

```
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

A função *loop*, como visto na programação em blocos, é nossa função "para sempre". É onde colocamos o código que será repetido indefinidamente.

Como enviaremos apenas sinal de HIGH e LOW (aceso e apagado, respetivamente), utilizaremos a função `digitalWrite(LED_BUILTIN, HIGH);`

Aciona um pino digital

Pino (número ou nome do pino que deseja acionar)

Valor (HIGH ou LOW)

Figura 4.4: Exemplo de atividades colaborativas inclusas no guião

4.1.5 Atividades interativas

Durante o decorrer da leitura do projeto, o aluno é apresentado a atividade interativas, onde são expostos a perguntas e desafios a serem resolvidos no decorrer do projeto, normalmente exigindo um maior nível de compreensão dos assuntos abordados. Essas atividades são chamadas no projeto de "Pense sempre além", convidando o aluno a ir além no conhecimento que está sendo construído, como mostrado na Figura 4.5;

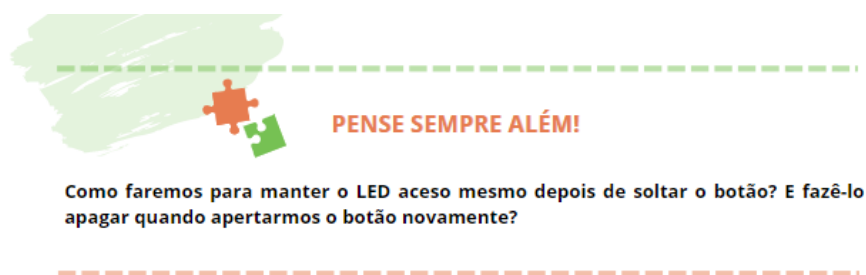


Figura 4.5: Exemplo de atividades interativas "Pense sempre além"

4.1.6 Desafio

Após finalizado o projeto proposto, o aluno é convidado a resolver um desafio com um nível elevado, é apresentado o contexto e o problema a ser resolvido, sem mais instruções, como mostrado na Figura 4.6.

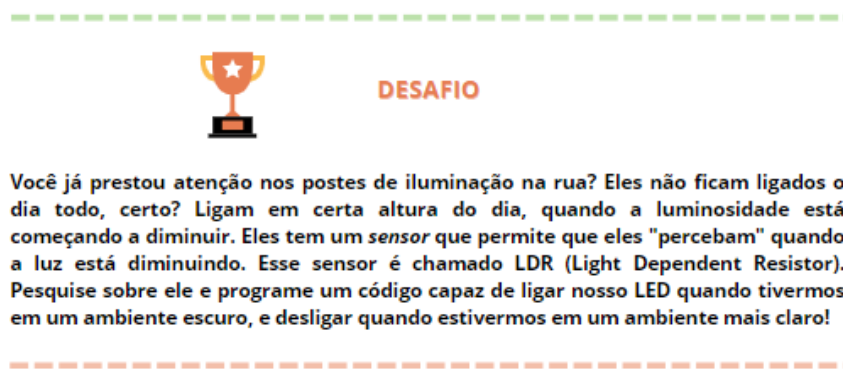


Figura 4.6: Modelo de desafio proposto ao fim do guia

4.2 Gotas de conhecimento

Ao final de cada projeto o aluno conta com o que foi chamado de gotas de conhecimento, uma coletânea de módulos de aprendizagem que reúne os conceitos que foram abordados pelo projeto e são detalhados separadamente, onde o aluno pode aprofundar seu conhecimento sobre cada componente utilizado, bem como compreender de forma mais adequada os sistemas, como mostrado na Figura 4.7;

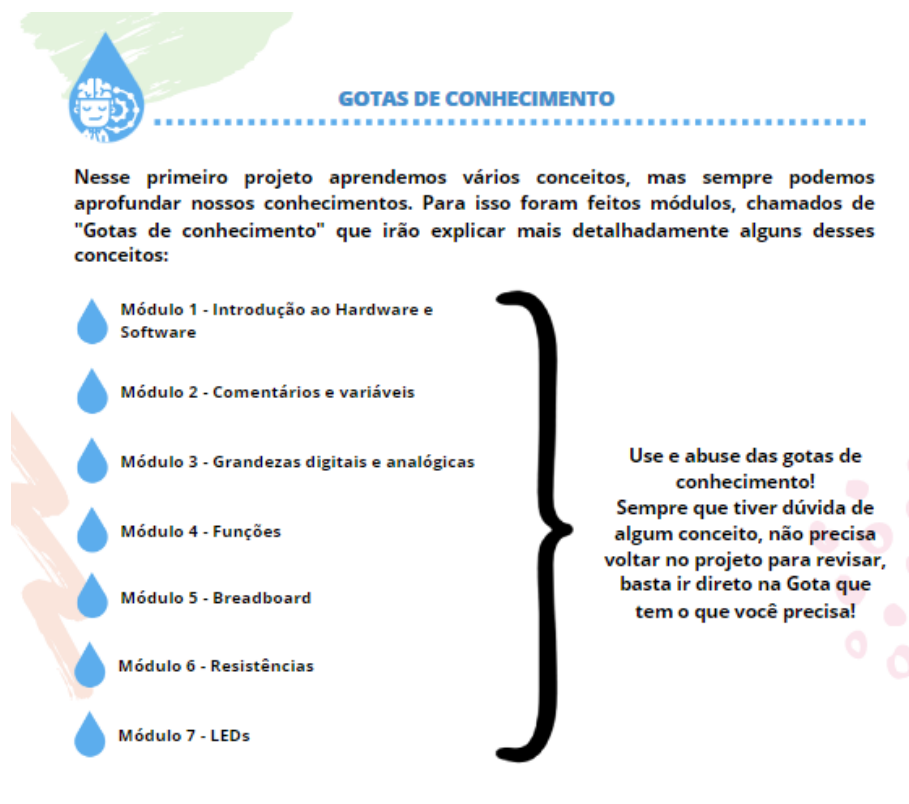


Figura 4.7: Lista das gotas de conhecimento correspondentes ao projeto

A Figura 4.8 mostra um exemplo da construção da gota de conhecimento, onde, como dito anteriormente, traz conceitos aprofundados dos componentes utilizados no projeto.



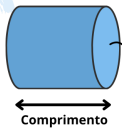
MÓDULO 6

RESISTÊNCIAS ELÉTRICAS

Quando existe uma corrente elétrica em um condutor, os elétrons livres se deslocam, gerando colisão entre si e também nas paredes do condutor. Essas colisões geram uma certa dificuldade para se deslocar, caracterizando a resistência elétrica.

Em resumo: A resistência elétrica é a dificuldade física que o material oferece à passagem de corrente elétrica, e segundo o Sistema Internacional (SI) é medida em ohms (Ω).

Cada condutor oferece uma resistência diferente à passagem de elétrons, que pode ser caracterizada por alguns fatores:



- comprimento do condutor (L) : quanto maior o condutor, maior a resistência.
- área da seção transversal (A): quanto menor a área, maior a resistência.
- resistividade do material (p): capacidade do material se opor a corrente elétrica. Quanto maior a resistividade, maior a resistência elétrica.

A Segunda Lei de Ohm resume esses fatores na seguinte equação:

$$R = p \frac{L}{A}$$

Vamos pensar em alguns exemplos de resistências elétricas no nosso dia a dia?



Secador de cabelo, aquecedor elétrico, ferro de passar roupa.



Percebeu que todos esquentam? Essa transformação de energia elétrica em energia térmica (calor) é chamada de Efeito Joule. Lembra que o deslocamento dos elétrons geram colisão com os átomos do condutor? O aumento da temperatura é gerado através do aumento da agitação dos átomos.



Esse aumento de temperatura deve ser levado em consideração quando formos escolher a resistência que utilizaremos nos nossos projetos.

Mas como saber qual valor de resistência deve ser usado? Pra isso iremos utilizar a Lei de Ohm:

$$\Delta U = R * i \quad \text{Onde } U \text{ é a diferença de potencial, } R \text{ a resistência e } i \text{ a corrente}$$

Deve-se conhecer a diferença de potencial que temos em cima dos componentes para fazer o cálculo da resistência. Iremos utilizar o exemplo do LED vermelho do nosso primeiro projeto.

Para calcularmos a resistência utilizada, precisamos primeiro buscar no manual do LED qual sua corrente e tensão de funcionamento. Esse manual se chama Datasheet. Segue abaixo a tabela de onde retiramos essas informações:

Características Optoeletrônicas	Symbol	Min.	Tip.	Max.	Unid.	Condição de Teste
Intensidade Luminosa	I_v		140		md	$I_f=20mA$ (Nota 1)
Ângulo de Visada	$2\theta_{1/2}$		30		Grdu	(Nota 2)
Comprim. de Onda	λ_d		640	660	nm	$I_f=20mA$ (Nota 3)
Tensão Direta	V_f	2.0	2.5		V	$I_f=20mA$
Corrente Reversa	I_r	—	50		μA	$V_r=8V$

Fonte: Datasheet LED 5mm - VERMELHO DIFUSO SYM-R503-30-D Symtronic

Para o LED vermelho, temos uma tensão direta de 2,0 V com uma corrente de 20mA.

Iremos utilizar esses dados para calcular.

Como o Arduino tem uma fonte de tensão de 5V, e o LED irá utilizar apenas 2V, teremos uma diferença de potencial (deltaU) de 3V. Ficando assim:

$$V = R * i$$

$$(5-2) = R * 0,02$$

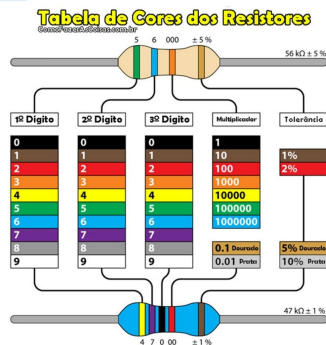
$$R = 150 \text{ ohm}$$

Ou seja, chegamos a conclusão de que precisamos de uma resistência de 150 ohms para fazer nosso LED funcionar. Agora, como encontramos qual LED é o de 150 ohms?

Temos duas formas mais simples:

Testar as resistências uma a uma até encontrar uma de 150 ohms (parece um pouco demorado certo?). Ou conhecer o código de cores das resistências:

CÓDIGO DE CORES DAS RESISTÊNCIAS



Vamos ver alguns exemplos de resistências para implementarmos o código de cores?



Resistência de 4 faixas:
Primeira - Vermelho = 2
Segunda - Roxo = 7
Terceira - Amarela = 10000 (multiplicador)
Quarta - Dourado = 5% (tolerância)
270000 ohms
ou 270 k ohms



Resistência de 4 faixas:
Primeira - Marrom = 1
Segunda - Preto = 0
Terceira - Laranja = 1000 (multiplicador)
Quarta - Dourado = 5% (tolerância)
10000 ohms
ou 10 k ohms

E a nossa resistência de 150 ohms? Como seria?



150 ohms

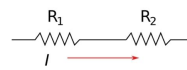
Resistência de 4 faixas:
Primeira - Marrom = 1
Segunda - Verde = 5
Terceira - Marrom = 10 (multiplicador)
Quarta - Dourado = 5% (tolerância)

Porém pode ser que o cálculo da resistência não dê um valor comercial, ou seja, pode ser que precise arredondar para cima para encontrar um valor disponível no mercado.

Mas podemos também fazer associação de resistências:

RESISTÊNCIAS EM SÉRIE

Elas estarão ligadas em série quando uma estiver ligada em seguida da outra, sem nenhum outro componente no meio:



Percebe-se que a corrente que circula nas resistências é a mesma, e a diferença de potencial no sistema será a soma da diferença de potencial em cada uma.

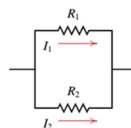
$$\Delta V = \Delta V_1 + \Delta V_2 = (R_1 + R_2) * i$$

$$R_{eq} = R_1 + R_2$$

Ou seja, quando colocamos duas resistências em série elas tem suas resistências somadas.

RESISTÊNCIAS EM PARALELO

Duas resistências estão em paralelo quando seus terminais estão ligados no mesmo ponto. Dessa forma:



Diferente da associação em série, agora a diferença de potencial é a mesma nas duas resistências, e a corrente no sistema é a soma das correntes nas resistências.

$$i = i_1 + i_2 = \left(\frac{1}{R_1} + \frac{1}{R_2}\right) * \Delta V$$

$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} \text{ ou } R_{eq} = (R_1 * R_2) / (R_1 + R_2)$$

Figura 4.8: Exemplo de um módulo de gota de conhecimento

4.3 Materiais de avaliação e *feedback*

Uma vez que o objetivo principal da investigação em design que está sendo discutida nesse trabalho, é entender se o processo de aprendizagem dos alunos foi bem sucedido com o material desenvolvido, será reunida uma gama de dados para entender a evolução da aprendizagem e o progresso no pensamento e raciocínio dos alunos durante o experimento. Para isso, foram adquiridos dados tanto dos alunos quanto dos professores. Nas subsecções a seguir serão discutidos os métodos de avaliação de cada um, bem como as ferramentas utilizadas para tal.

4.3.1 Avaliação dos discentes

Ainda de acordo com o método de investigação escolhido, os alunos foram considerados como um grupo pequeno de teste, afim de entender a efetividade do material e de sua implementação no ambiente escolar.

A avaliação dos alunos foi dada por meio da plataforma Kahoot!, uma plataforma de aprendizagem baseada em jogos que permite criar e administrar questionários, investigações e discussões. Os alunos podem participar dessas atividades acessando-as em seus próprios dispositivos (como *smartphones* ou *laptops*) e conectando-se ao Kahoot! sessão através de um código [52].

Além disso, Kahoot! permite uma competição amigável entre os alunos, o que pode aumentar o envolvimento e a motivação. Ele também permite avaliações fáceis e rápidas da compreensão do aluno, o que pode ajudar os professores a adaptar suas instruções para atender melhor às necessidades de seus alunos [53].

No geral, Kahoot! ajuda a criar um ambiente de aprendizagem interativo e envolvente, que pode ajudar a melhorar a motivação e a compreensão do aluno. Por esses motivos essa plataforma foi a escolhida para recolher os dados dos alunos.

Durante o processo de avaliação foram feitas perguntas de cunho prático, como mostrado na Figura 4.9, de conceitos teóricos, como mostrado na Figura 4.10 e perguntas que

visavam entender a motivação e comprometimento do aluno, como visto na Figura 4.11.

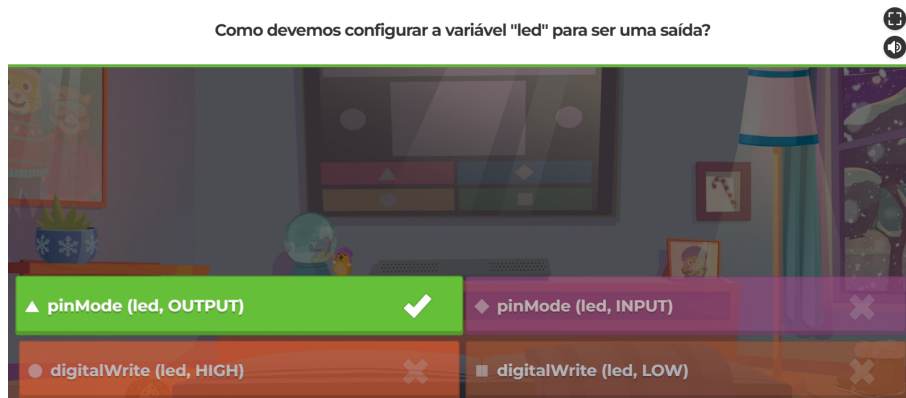


Figura 4.9: Pergunta prática

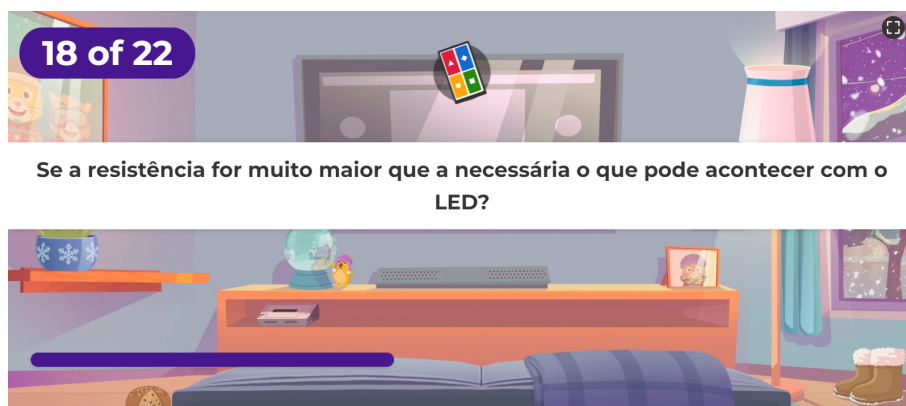


Figura 4.10: Pergunta teórica

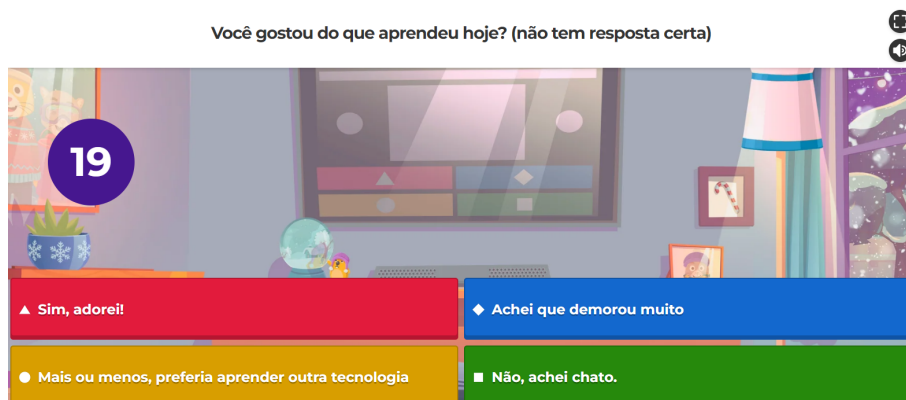


Figura 4.11: Pergunta de motivação

No Capítulo 5, os resultados serão discutidos, afim de explorar a eficácia do material didático na promoção da aprendizagem e o impacto dos diferentes métodos utilizados na conceção do material.

4.3.2 *Feedback* dos docentes

Com o objetivo de estar em consonância com a teoria da IDE, o professor foi visto como um especialista, agregando na avaliação formativa com suas opiniões sobre o conteúdo, o delineamento e a qualidade técnica do material. Para isso, ao final da implementação de cada projeto, foi enviado ao professor responsável pela turma um formulário *online*, com o intuito de receber informações valiosas sobre o processo de ensino e aprendizagem do aluno naquele projeto em si. Para isso foram elaboradas perguntas qualitativas e quantitativas para guiar o professor a uma reflexão de como foi a implementação do projeto. Foram blocos de perguntas voltadas à experiência de aprendizagem dos alunos, a experiência do ensino aos alunos, como se imagina a integração dos materiais na aula curricular, e para finalizar um *feedback* geral de melhorias. As Figuras 4.12, 4.13, 4.14 e 4.15 a seguir mostram as perguntas aplicadas no formulário.

Two screenshots of a feedback questionnaire for teachers. Each screenshot shows a question followed by a 5-point Likert scale from 'Discordo totalmente' to 'Concordo totalmente'.

Top screenshot:

Você notou alguma mudança na atitude dos alunos em relação ao aprendizado *
após a implementação dos projetos de robótica?

1 2 3 4 5

Discordo totalmente Concordo totalmente

Bottom screenshot:

As atividades de robótica melhoraram o pensamento crítico e as habilidades de *
resolução de problemas dos alunos

1 2 3 4 5

Discordo totalmente Concordo totalmente

Figura 4.12: Perguntas formulário *feedback* docentes (a)

Os projetos de robótica ajudaram a envolver os alunos que antes estavam desinteressados nas disciplinas STEM *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Como você classificaria o nível de motivação dos alunos durante as atividades de robótica em comparação com outros temas? *

1 2 3 4 5

Muito desmotivados Completamente motivados

Figura 4.13: Perguntas formulário *feedback* docentes (b)

O conteúdo e as atividades dos projetos foram desafiadores e estimulantes para os alunos *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Na sua opinião, quais foram os pontos fortes dos projetos? *

Your answer _____

Houve algum ponto negativo nos projetos? Em caso afirmativo, explique. *

Your answer _____

Figura 4.14: Perguntas formulário *feedback* docentes (c)

The image shows two sections of a feedback form, each with a green border. The first section contains the question: "Os projetos propostos se alinham com o currículo e os objetivos de aprendizagem" followed by a red asterisk. Below the question is a 5-point Likert scale with radio buttons, labeled "1" through "5" above the buttons. The scale is anchored with "Discordo totalmente" on the left and "Concordo totalmente" on the right. The second section contains the question: "Você recomendaria a aplicação destes projetos para outros professores?" followed by a red asterisk. Below the question is a 5-point Likert scale with radio buttons, labeled "1" through "5" above the buttons. The scale is anchored with "Não recomendaria" on the left and "Com certeza recomendaria" on the right.

Figura 4.15: Perguntas formulário *feedback* docentes (d)

4.3.3 *Feedback* dos discentes

A fim de alinhar-se com a teoria do design educacional (IDE), os alunos que participaram da implementação dos projetos foram solicitados a fornecer *feedback* por meio de um formulário online que utilizava escala Likert. Desta forma, contribuindo para a avaliação formativa ao compartilhar suas opiniões sobre o conteúdo, o design e a qualidade técnica do material. Essas perguntas foram elaboradas para orientar o aluno a refletir sobre a implementação do projeto, abordando aspectos como a experiência de aprendizagem, o nível de dificuldade dos materiais, bem como a motivação de continuar a aprender eletrônica e robótica.

As Figuras 4.16, 4.17 e 4.18 a seguir mostram as perguntas aplicadas no formulário.

	Muito ruim	Ruim	Neutro	Bom	Muito bom
Qual era seu nível de conhecimento prévio em robótica e programação?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Suas habilidades e conhecimentos em robótica mudaram desde o início das atividades?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Os projetos atenderam às suas expectativas?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 4.16: Perguntas formulário *feedback* discentes (a)

Você achou as atividades desafiadoras o suficiente?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Os materiais das atividades (guião e módulos) foram úteis e fáceis de entender?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Você sentiu que as atividades ajudaram a desenvolver novas habilidades ou mesmo aprimorar as já existentes?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 4.17: Perguntas formulário *feedback* discentes (b)

Você achou benéfico o trabalho ser em grupo?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O método de avaliação (Kahoot!) foi adequado?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Você achou o tempo dos projetos adequados?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Você recomendaria essas atividades para outros colegas?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 4.18: Perguntas formulário *feedback* discentes (c)

No Capítulo 5, os resultados serão discutidos e analisados, mostrando como esses *feedbacks* alimentaram o ciclo da IDE, possibilitando a melhoria dos materiais dos projetos seguintes, levando em conta os dados recolhidos.

Capítulo 5

Fase 3: Implementação e avaliação

Após o desenvolvimento do material didático descrito anteriormente, foi criado o projeto "*Arduino Project for Students*" em parceria com a Escola Secundária Emídio Garcia, especialmente para a aplicação dos materiais desenvolvidos. Este projeto foi alinhado com o Plano de Ação Para o Desenvolvimento Digital da Escola (PADDE), também chamado de programa "C@pacita-te", que tem como objetivo "potenciar os processos de inovação através do digital, nas escolas e adequá-las aos contextos e desafios atuais da nossa sociedade"[54]. O poster de divulgação do projeto foi colocado no *site* da escola, como mostrado na Figura 5.1. Com este projeto foi possível iniciar a fase de testes dentro da escola com duas turmas: 9º ano e 12º ano, com alunos de 12 a 15 anos em média, respetivamente. O objetivo da fase de teste era determinar se o processo de aprendizagem foi bem-sucedido com o material desenvolvido.

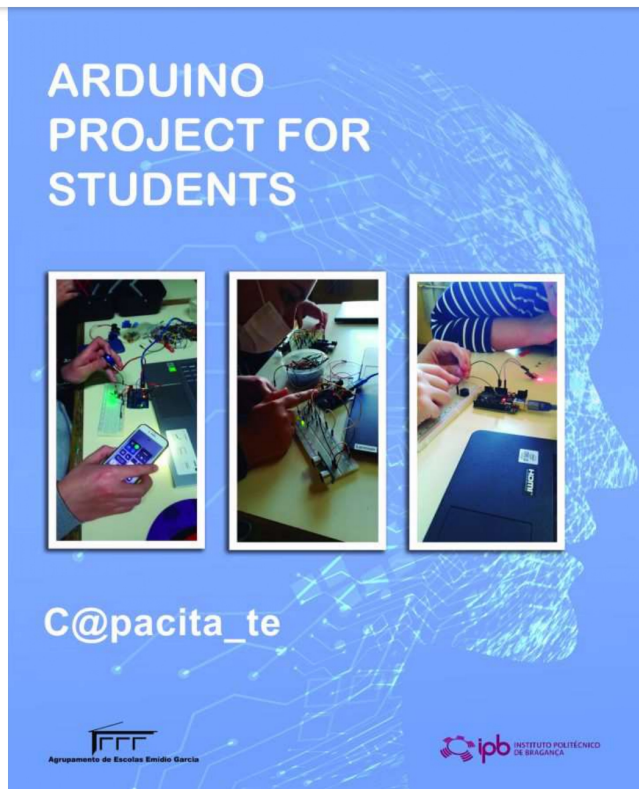


Figura 5.1: Divulgação do Projeto no site da Escola Emídio Garcia

Cada projeto foi programado para ser aplicado na aula semanal de TIC prevista, a qual tem duração de duas horas-aula (90 minutos), afim de proporcionar aos alunos um cronograma focado e estruturado para concluir seu trabalho. A fase de testes ocorreu no total de 12 horas-aula, onde nesse período, as turmas participantes do estudo utilizaram o novo material didático em seu currículo regular nas aulas de TIC.

Durante a aplicação dos projetos, a professora assumiu o papel de facilitadora e não de fonte de informação. Essa abordagem deu aos alunos a oportunidade de serem os atores principais em seu próprio processo de aprendizagem. Ao fornecer um ambiente

que permitia a exploração e a descoberta, os alunos puderam se apropriar de seu próprio conhecimento e construir sua compreensão de conceitos STEM por meio de atividades práticas.

No decorrer das atividades os alunos foram observados e avaliados para ver como eles foram capazes de compreender e aplicar os conceitos ensinados. Para ser feita essa análise, como visto anteriormente no Capítulo 4, a avaliação do material foi feita pelos alunos e pelos professores que utilizaram o material. Os resultados deste teste fornecem informações valiosas sobre a eficácia do método construcionista de aprendizagem na sala de aula de TIC. Ele também oferece a oportunidade de refletir sobre a compreensão dos alunos e fazer os ajustes necessários para aulas futuras.

A Tabela 5.1 mostra o calendário de execução das aulas que foram aplicados cada projeto nas respectivas turmas.

Tabela 5.1: Calendário de aplicação dos projetos

Data	Projeto	Turma
28/10/2022	Blink	9º ano
11/11/2022	Campainha	9º ano
25/11/2022	Campainha	9º ano
5/12/2022	Blink	12º ano Turma 1
6/12/2022	Blink	12º ano Turma 2
9/12/2022	Sensor de distância	9º ano

Nas secções seguintes serão abordados os resultados da avaliação dos materiais, analisando os resultados da turma do 9º ano e do 12º ano separadamente.

5.1 Resultados e avaliações 9º ano

A turma do 9º ano era composta por 18 alunos, com uma média de 12 anos de idade. Para a implementação decidiu-se dividir a turma em trios, para facilitar a colaboração e o trabalho em equipe, totalizando 6 trios, o que possibilitou utilizar os materiais previamente disponíveis na escola, de forma a não gerar gastos adicionais.

Para iniciar a aplicação do material em sala de aula, primeiramente foi apresentado para a turma o objetivo do projeto e esclarecido que estariam fazendo parte de uma investigação, e ao final de cada aula estariam convidados a responder os questionários via Kahoot!, onde os dados adquiridos seriam importantes para análise da eficácia do material.

5.1.1 Projeto 1 - Blink

O Projeto 1 foi aplicado em sala de aula, o qual foi elaborado com atividades afim de introduzir os conceitos iniciais de robótica, eletrônica e programação. Teve uma duração de 1 hora e 33 minutos, inclusos 15 minutos de duração da avaliação dos alunos. A avaliação consistiu em 22 perguntas relacionadas ao conteúdo do projeto, como visto anteriormente, onde as respostas foram fornecidas pelo grupo, totalizando os 6 grupos formados pelos alunos. Os resultados estão resumidos na Tabela 5.2.

Tabela 5.2: Resultados da aplicação do Projeto 1 - 9º ano

Projeto 1	
Número de grupos	6
Média de acertos	65.9%
Tempo médio de projeto	1h 15min
Tempo médio de questionário	15min

A turma obteve uma média de 65.9% de acerto nas respostas do questionário, apesar de ser um resultado satisfatório, decidiu-se realizar uma análise individual das perguntas, demonstrada no gráfico mostrado na Figura 5.2, afim de identificar quais foram as maiores fraquezas.

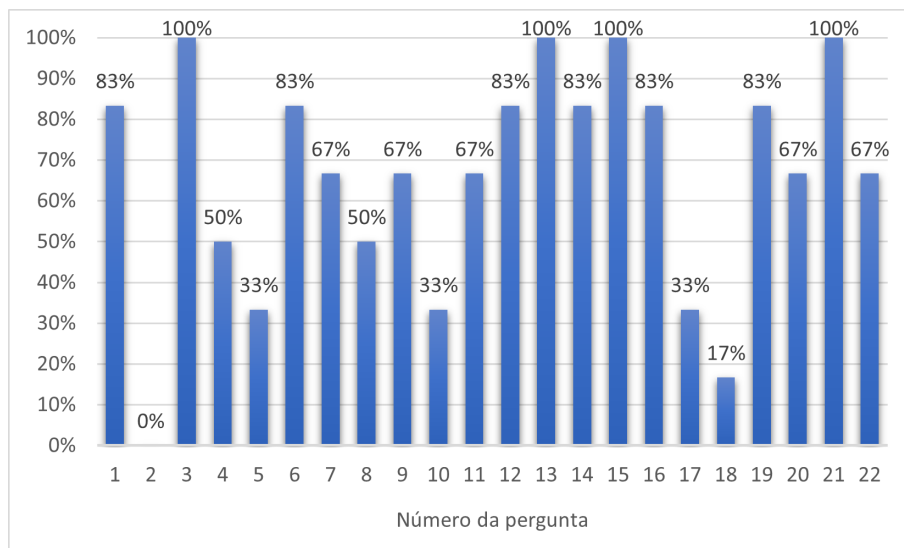


Figura 5.2: Gráfico correspondente a taxa de acerto por pergunta - Projeto 1

A partir dessa análise pode-se perceber uma taxa baixa de acerto nas perguntas de número 2 e 18, em comparação com as outras, as quais:

- Pergunta número 2: O que significa uma plataforma *Open Source*?
- Pergunta número 18: Se a resistência for muito maior que a necessária o que pode acontecer com o LED

Ambas as perguntas foram baseadas em tópicos que foram apenas discutidos nos módulos das gotas de conhecimento, o que pode-se concluir que os alunos não estavam acessando os módulos para aprofundar os fundamentos, como era esperado. Essa análise, juntamente com o *feedback* da professora permitiu a alimentação do ciclo da IDE, a partir da melhoria da estratégia de abordagem dos materiais, onde a mesma propôs que utilizássemos a técnica *flipped classroom*, a qual “inverte o modelo tradicional de sala de aula, introduzindo conceitos antes da aula, permitindo que os educadores usem o tempo de aula para orientar cada aluno por meio de aplicações ativas, práticas e inovadoras de os princípios do curso”[55]. Tendo dito isso, a partir do Projeto 2 os alunos tiveram acesso as Gotas de conhecimento na semana que antecede a aplicação do projeto em sala de

aula, para que pudessem estudar anteriormente os conceitos e aplicá-los em sala de aula no seguimento dos guiões. Nas próximas subsecções serão analisadas as consequências obtidas através desta mudança de abordagem.

5.1.2 Projeto 2 - Campanha

Na aplicação do Projeto 2, como dito anteriormente, foi utilizada a técnica *flipped classroom*, onde foi disponibilizado aos alunos com antecedência os materiais das gotas de conhecimento, com a instrução de os lerem antes do tempo previsto de aula.

O Projeto 2 foi aplicado em sala de aula, o qual foi elaborado com atividades afim de integrar mais componentes no sistema e aumentar o grau de complexidade de programação. Teve uma duração de 3 horas (4 horas-aula, com espaço de uma semana entre duas aulas), inclusos 15 minutos de duração da avaliação dos alunos. A avaliação consistiu em 18 perguntas relacionadas ao conteúdo do projeto, como visto anteriormente, onde as respostas foram fornecidas pelo grupo, totalizando os 6 grupos formados pelos alunos. Os resultados estão resumidos na Tabela 5.3.

Tabela 5.3: Resultados da aplicação do Projeto 2 - 9º ano

Projeto 2	
Número de grupos	6
Média de acertos	79%
Tempo médio de projeto	3 horas
Tempo médio de questionário	15min

A turma obteve uma média de 79% de acerto nas respostas do questionário, apresentando uma melhoria de 19,9% em relação ao resultado do questionário do Projeto 1, o que foi considerado um sucesso pela mudança de estratégia abordada, fechando o ciclo da IDE. Da mesma forma do projeto anterior, realizou-se uma análise individual das perguntas, demonstrada no gráfico mostrado na Figura 5.3, afim de identificar quais foram as maiores fraquezas.

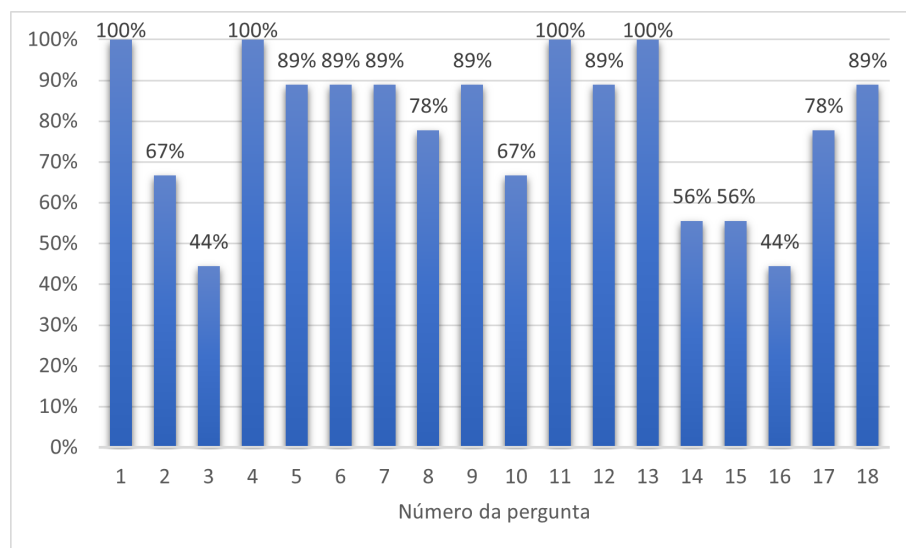


Figura 5.3: Gráfico correspondente a taxa de acerto por pergunta - Projeto 2

As perguntas de número 2 e 15 do questionário deste projeto correspondem às mesmas perguntas 2 e 18 do questionário do Projeto 1, respetivamente, as quais tinham sido analisadas anteriormente revelando as fraquezas da aplicação do projeto anterior. Decidiu-se repetir as mesmas perguntas no questionário para poder avaliar se houve melhoria utilizando a técnica de *flipped classroom*. Pode-se perceber o preenchimento dessa lacuna a partir da análise da taxa de acerto das mesmas, onde ambas apresentaram um aumento significativo.

Na análise destes resultados não percebeu-se uma falha expressiva, levando em consideração o resultado da taxa de acerto, entretanto, como discutido no início do capítulo, o tempo previsto de aplicação era de 90 minutos, porém este teve um tempo final de 180 minutos, levando o dobro do tempo previamente estipulado. Esta necessidade de mais tempo dos alunos para completar o Projeto 2, juntamente com o *feedback* da professora, permitiu a alimentação do ciclo da IDE, onde pode-se visitar o conteúdo do Projeto 2 e adaptá-lo a realidade, diminuindo o número de atividades, para que pudesse ser aplicado em apenas 2 horas-aula, mantendo os alunos comprometidos e motivados do início ao fim.

5.1.3 Projeto 3 - Sensor de distância

Após a conclusão dos projetos 1 e 2, e com a coleta dos questionários e *feedbacks* valiosos de todas as partes envolvidas, os dados foram analisados cuidadosamente para identificar áreas onde melhorias poderiam ser feitas e lições valiosas poderiam ser aprendidas.

Foram aplicadas as lições aprendidas no desenvolvimento do projeto 3, incorporando o *feedback* em todos os aspetos do projeto, desde a abordagem com a técnica *flipped classroom*, e também do nível de dificuldade atrelado ao tempo necessário para aplicação do projeto.

O Projeto 3 foi aplicado em sala de aula, com o intuito de aprofundar a utilização dos sensores e incluir as grandezas analógicas no processo de aprendizagem. Teve uma duração de 1 hora e 30 minutos (2 horas-aula), inclusos 15 minutos de duração da avaliação dos alunos. A avaliação consistiu em 18 perguntas relacionadas ao conteúdo do projeto, como visto anteriormente, onde as respostas foram fornecidas pelo grupo, totalizando os 6 grupos formados pelos alunos. Os resultados estão resumidos na Tabela 5.4.

Tabela 5.4: Resultados da aplicação do Projeto 3 - 9º ano

Projeto 3	
Número de grupos	6
Média de acertos	83.3%
Tempo médio de projeto	1h 30min
Tempo médio de questionário	15min

A turma obteve uma média de 83.3% de acerto nas respostas do questionário, apresentando uma melhoria de 5,4% em relação ao resultado do questionário do Projeto 2, e uma melhoria de 26.4% em relação ao resultado do Projeto 1, o que foi considerado um sucesso pela mudança de estratégia abordada, fechando o clique da IDE. Na Figura 5.4 é mostrada o aumento progressivo da taxa de acerto nos questionários do Kahoot! pelos alunos do 9º ano, mostrando os resultados positivos alcançados pela aplicação dos materiais e métodos utilizados.

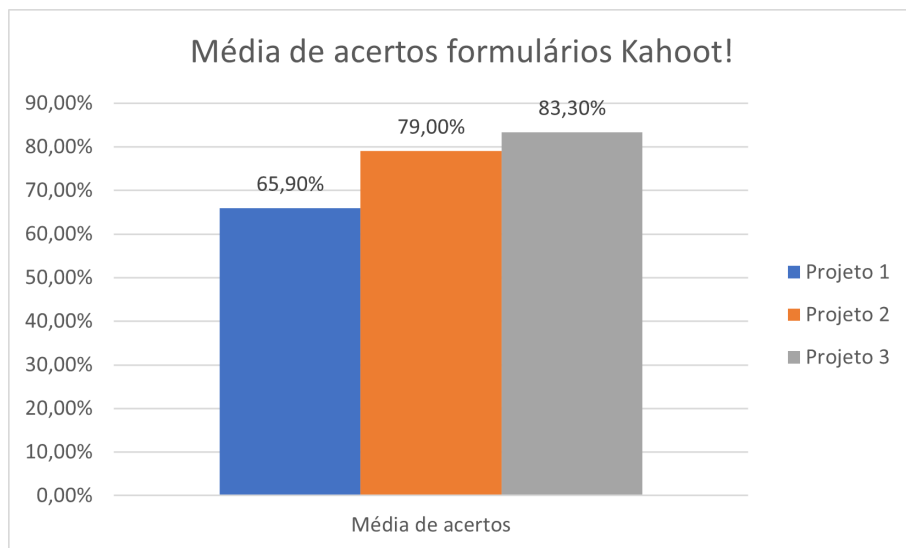


Figura 5.4: Evolução de taxa de acerto dos projetos

5.1.4 *Feedback* dos alunos

A Figura 5.5 resume o feedback fornecido pelos alunos na investigação de avaliação dos projetos. Uma escala Likert é utilizada para avaliar o nível de avaliação em relação a diferentes aspetos de aprendizagem.

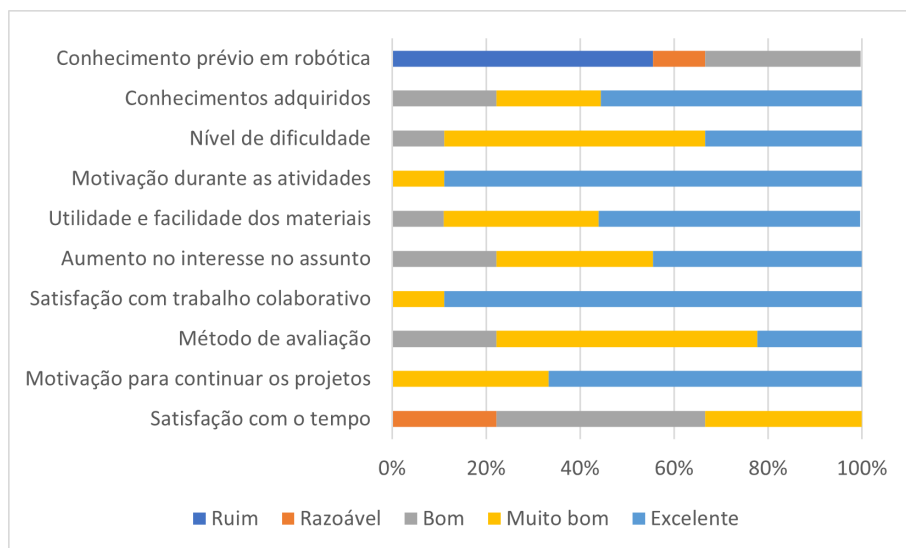


Figura 5.5: Avaliação dos *feedbacks* dos alunos 9º ano

Com base nos resultados da investigação de avaliação, é possível observar que a maioria dos alunos do 9º ano (78%) ficou satisfeita com o nível de conhecimento adquirido. No entanto, eles expressaram o desejo de que alguns projetos tivessem uma duração maior, especialmente considerando que 67% dos participantes tinham pouco ou nenhum conhecimento prévio na área de robótica. A colaboração entre os alunos também foi muito valorizada, com aprovação unânime do trabalho em equipa, "porque se fosse individual iria haver mais dificuldade em fazer, e ao ser em grupo já nos ajudávamos uns aos outros".

Além disso, a grande maioria dos participantes (89%) considerou o material utilizado como uma ferramenta de ensino e aprendizagem eficaz, sendo um "novo aprendizagem e divertido de se fazer" e acharam que a dificuldade das atividades propostas era adequada para a faixa etária e conhecimento dos alunos, já que "algumas das atividades eram um pouco complicadas e desafiadoras".

É encorajador constatar que 78% dos alunos declararam estar mais interessados em conteúdos STEM após participar dos projetos. Isso demonstra o impacto positivo do currículo desenvolvido na motivação dos estudantes nessa área específica. Além disso, todos eles estão motivados a continuar utilizando os materiais e participando de projetos futuros, buscando uma aprendizagem contínua no campo STEM, e ainda declaram que "se interessaram mais por robótica e aprenderam algumas coisas e nomes que não faziam ideia do que eram".

Esses resultados evidenciam o sucesso do material desenvolvido em proporcionar uma experiência de aprendizagem enriquecedora, despertando o interesse dos alunos pela robótica e incentivando sua contínua participação e comprometimento na área.

5.2 Avaliação e resultados 12º ano

Após a finalização do Projeto 1 no 9º ano, o mesmo foi implementado em duas turmas de 12º ano. Os materiais foram fornecidos com antecedência, seguindo as mudanças necessárias percebidas através do ciclo da IDE. As turmas eram compostas, no total, por 21 alunos, com uma média de 15 anos de idade. A turma foi dividida em grupos de três,

como no 9º ano, totalizando 7 grupos.

No início da aplicação foi apresentado o objetivo do projeto e esclarecido que estariam fazendo parte de uma investigação, e ao final de cada aula estariam convidados a responder os questionários via Kahoot!, onde os dados adquiridos seriam importantes para análise da eficácia do material.

A aplicação do Projeto 1 teve uma duração de 1 hora e 25 minutos, inclusos 15 minutos de duração da avaliação dos alunos. A avaliação consistiu em 22 perguntas relacionadas ao conteúdo do projeto, como visto anteriormente, desta vez as respostas foram colhidas individualmente, totalizando 21 respostas ao questionário, os resultados estão resumidos na Tabela 5.5.

Tabela 5.5: Resultados da aplicação do Projeto 1 - 12º ano

Projeto 1	
Número de participantes	21
Média de acertos	82.5%
Tempo médio de projeto	1h 25min
Tempo médio de questionário	15min

É evidente uma melhoria significativa na taxa de acerto do Projeto 1 em comparação com o 9º ano, o que resultou em um aumento de 25.18% na média de acertos da turma. Esse resultado reforça mais uma vez o sucesso do material desenvolvido e do método IDE em promover uma melhoria contínua no material para os alunos.

5.2.1 *Feedback* dos alunos

A Figura 5.6 apresenta os resultados dos *feedbacks* dos alunos, fornecendo uma representação visual das respostas em cada aspecto avaliado. Isso permite uma análise mais objetiva e comparativa dos resultados, identificando áreas de destaque e possíveis melhorias.

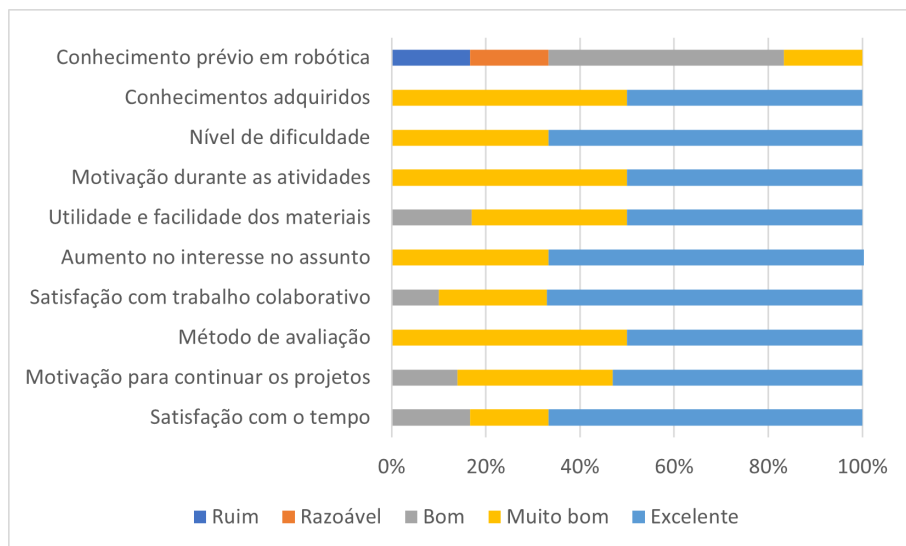


Figura 5.6: Avaliação dos *feedbacks* dos alunos 12º ano

Com base nos resultados da investigação de avaliação, pode-se constatar que os alunos do 12º ano estão satisfeitos com o nível de conhecimento adquirido nos projetos. Mesmo considerando que 67% dos participantes já possuíam algum conhecimento prévio na área de robótica, todos os alunos mostraram-se satisfeitos com a aprendizagem proporcionada.

A colaboração entre os alunos foi valorizada, com a maioria (90%) aprovando o trabalho em equipe. Os alunos destacaram que "desenvolveram as competências todos juntos e surgiam dúvidas de pessoas diferentes".

O material utilizado nos projetos recebeu uma avaliação positiva da grande maioria dos participantes (83%). Os alunos consideraram o material como uma ferramenta eficaz para o ensino e aprendizagem, e julgaram que a dificuldade das atividades propostas estava adequada para a faixa etária e o nível de conhecimento dos alunos.

Outro ponto positivo observado foi o aumento do interesse dos alunos pela robótica e programação após participar dos projetos. Esse interesse crescente foi expresso por 86% dos alunos, que relataram que "foi interessante, pois fez ganhar mais gosto por programação que é uma matéria mais aborrecida e difícil de compreender".

Esses resultados indicam o impacto positivo do currículo e dos projetos na motivação

dos alunos e no desenvolvimento de interesse contínuo pela área de robótica e programação. Além disso, os alunos estão motivados a continuar utilizando os materiais e a participar de projetos futuros, buscando uma aprendizagem contínua no campo STEM.

Essas conclusões evidenciam o sucesso do material desenvolvido e dos projetos na promoção de uma experiência de aprendizagem enriquecedora e no estímulo ao comprometimento dos alunos na área de robótica e programação.

5.3 *Feedback* docentes

Com relação ao feedback das docentes, ambas testemunharam uma mudança notável na atitude e comprometimento dos alunos diante das atividades propostas, além de observarem melhorias significativas no pensamento crítico e nas capacidades de resolução de problemas dos estudantes. Um objetivo adicional alcançado foi a abordagem das chamadas "*soft skills*", por meio das atividades desenvolvidas. As duas docentes destacaram um aumento significativo na motivação dos estudantes para se envolverem nas atividades propostas, superando outros temas abordados em sala de aula. Elas consideraram que as atividades ofereceram desafios adequados, mantendo os alunos envolvidos e motivados.

As professoras ressaltaram como aspectos positivos dos projetos o estímulo à inovação, criatividade e pensamento crítico dos alunos, bem como o ensino da aplicação prática de programação e pensamento computacional. Foi levantada a questão da disponibilidade limitada de equipamentos por parte da escola, e tanto as docentes quanto os alunos acreditam que alguns projetos poderiam demandar um tempo mais adequado para serem realizados em apenas uma aula.

Em resumo, ambas as docentes afirmam que os projetos aplicados estão alinhados com o currículo e os objetivos de aprendizagem estabelecidos, e obtiveram sucesso em alcançar esses objetivos.

Capítulo 6

Conclusões e trabalhos futuros

Os projetos foram criados com a intenção de apresentar aos iniciantes o empolgante mundo da eletrônica e da programação, caminhando com a necessidade percebida da inclusão de ensino STEM. O objetivo foi fornecer um material simples e intuitivo que ajudasse os alunos a entender os conceitos e princípios básicos desses campos.

A concepção dos projetos foi idealizada levando em consideração a necessidade de instruções claras e concisas e orientações passo a passo. O projeto foi estruturado de forma a facilitar o acompanhamento dos alunos, mesmo sem conhecimento prévio ou experiência em eletrônica e programação. Dessa forma, os alunos podem se concentrar em aprender os fundamentos sem se sentirem sobrecarregados ou frustrados.

No entanto, como visto anteriormente, também foram abordadas técnicas construcionistas, as quais foram aplicadas através das atividades interativas e do desafio proposto no final do projeto. Resultando em um modelo híbrido de métodos tradicionais e construcionistas.

Além disso, apesar de ser pensado para iniciantes, também reconheceu que alguns alunos provavelmente gostariam de se aprofundar no assunto e obter uma compreensão mais abrangente. Para resolver isso, foram criados módulos separados, chamados de "Gotas de Conhecimento". Esses módulos foram projetados para fornecer conhecimento mais técnico e aprofundado sobre vários conceitos, permitindo que os alunos obtivessem uma compreensão mais robusta de eletrônica e programação.

Os resultados da fase de testes e das avaliações foram altamente positivos. Tanto as turmas do 9º quanto do 12º ano apresentaram uma melhora significativa na compreensão e aplicação dos conceitos abordados no material didático. Os alunos também se mostraram mais comprometidos e motivados em sua aprendizagem, conforme evidenciado por sua maior participação nas discussões e atividades em sala de aula. Os professores também relataram um alto nível de satisfação com o material, observando que ele foi eficaz em atender às necessidades de uma gama diversificada de alunos.

Junto a isso, o projeto "Arduino Project for Students", criado em parceria com a Escola Secundária Emídio Garcia, foi selecionado a nível nacional pela ERTE-DGE (Equipa de Recursos e Tecnologias Educativas - Direção Geral de Educação) como um projeto exemplar no uso de práticas inovadoras com o digital nas escolas. Essa seleção é um reconhecimento significativo do sucesso do desenvolvimento dos materiais do presente trabalho, e é um estímulo para continuar a aprimorar as abordagens e compartilhar as experiências com outras escolas e educadores.

Em conclusão, o resultado do desenvolvimento do material didático, juntamente com a seleção do projeto pela ERTE-DGE, mostraram que o uso dessa abordagem pode levar a uma experiência de aprendizagem mais eficaz e envolvente para os alunos na educação de eletrónica e robótica no ensino secundário.

É importante notar que este estudo foi feito em uma amostra pequena e mais investigações são necessárias para dar robustez aos resultados. Além disso, algumas melhorias foram percebidas durante as aplicações, as quais podem ser aplicadas para alcançar resultados ainda melhores.

Da forma que os projetos foram desenvolvidos, ao final dos guiões sempre são deixados desafios para os alunos investigarem e desenvolverem em casa durante a semana, com a plataforma do Tinkercad, caso não possuam os materiais necessários.

Uma sugestão de melhoria é a avaliação continuada dos desafios propostos, a partir da demonstração em sala de aula das soluções encontradas por cada aluno para o desafio, onde os alunos podem discutir os vários caminhos encontrados, gerando uma troca de conhecimento entre eles.

Os projetos implementados nas aulas de TIC não fizeram parte das atividades de avaliação que compõem a nota final da disciplina. Uma sugestão é que no futuro essas atividades sejam consideradas atividades de avaliação, aumentando o comprometimento dos alunos, permitindo a expansão do material para ser utilizado durante todo o ano letivo.

Bibliografia

- [1] J. Pontes, C. A. Geraldes, F. P. Fernandes et al., «Relationship between Trends, Job Profiles, Skills and Training Programs in the Factory of the Future,» em *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, IEEE, vol. 1, 2021, pp. 1240–1245.
- [2] T. J. Kennedy e M. R. Odell, «Engaging students in STEM education.,» *Science Education International*, vol. 25, n.º 3, pp. 246–258, 2014.
- [3] S. Kucuk e B. Sisman, «Behavioral patterns of elementary students and teachers in one-to-one robotics instruction,» *Computers & Education*, vol. 111, pp. 31–43, 2017.
- [4] M. Menekse, R. Higashi, C. D. Schunn e E. Baehr, «The role of robotics teams' collaboration quality on team performance in a robotics tournament,» *Journal of Engineering Education*, vol. 106, n.º 4, pp. 564–584, 2017.
- [5] A. Gomoll, C. E. Hmelo-Silver, S. Šabanović e M. Francisco, «Dragons, ladybugs, and softballs: Girls' STEM engagement with human-centered robotics,» *Journal of Science Education and Technology*, vol. 25, n.º 6, pp. 899–914, 2016.
- [6] A. Master, S. Cheryan e A. N. Meltzoff, «Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science.,» *Journal of educational psychology*, vol. 108, n.º 3, p. 424, 2016.
- [7] R. W. Bybee, «What is STEM education,» *Science*, vol. 329, pp. 995–996, 2010.

- [8] J. O. Revilla e I. Arriasecq, «Construcción de un marco teórico para el enfoque STEAM en la Educación Primaria,» em *Encuentros de Didáctica de las Ciencias Experimentales: iluminando el cambio educativo*, Servizo de Publicacións, 2018, pp. 823–828.
- [9] H.-E. Chu, S. N. Martin e J. Park, «A theoretical framework for developing an intercultural STEAM program for Australian and Korean students to enhance science teaching and learning,» *International Journal of Science and Mathematics Education*, vol. 17, n.º 7, pp. 1251–1266, 2019.
- [10] C. Siry e A. Gorges, «Young students’ diverse resources for meaning making in science: learning from multilingual contexts,» *International Journal of Science Education*, vol. 42, n.º 14, pp. 2364–2386, 2020.
- [11] T. Kelley, «A conceptual Framework for Integrated STEM,» *International Journal of STEM Education. Springer*, vol. 3, n.º 1, 2016.
- [12] C.-W. Chang, J.-H. Lee, P.-Y. Chao, C.-Y. Wang e G.-D. Chen, «Exploring the possibility of using humanoid robots as instructional tools for teaching a second language in primary school,» *Journal of Educational Technology & Society*, vol. 13, n.º 2, pp. 13–24, 2010.
- [13] J. L. Ramos e R. G. Espadeiro, «Os futuros professores e os professores do futuro. Os desafios da introdução ao pensamento computacional na escola, no currículo e na aprendizagem,» *Educação, Formação & Tecnologias-ISSN 1646-933X*, vol. 7, n.º 2, pp. 4–25, 2014.
- [14] B. S. Jang, «The meta-analysis on effects of Arduino-based education for secondary school students,» *Journal of Industrial Convergence*, vol. 19, n.º 3, pp. 61–65, 2021.
- [15] M. Resnick, J. Maloney, A. Monroy-Hernández et al., «Scratch: programming for all,» *Communications of the ACM*, vol. 52, n.º 11, pp. 60–67, 2009.
- [16] D. A. Roberts, «Scientific literacy/science literacy,» *Handbook of research on science education*, vol. 1, pp. 120–144, 2007.

- [17] I. M. Greca Dufranc, E. García Terceño, M. Fridberg, B. Cronquist e A. Redfors, «Robotics and early-years STEM education: The botSTEM framework and activities,» *European Journal of STEM Education*, n.º 1, pp. 1–13, 2020.
- [18] E. Hazelkorn, C. Ryan, Y. Beernaert et al., «Science education for responsible citizenship,» *Report to the European Commission of the expert group on science education*, 2015.
- [19] M. M. Ali, R. Yager, E. Hacieminoglu e I. Caliskan, «Changes in student attitudes regarding science when taught by teachers without experiences with a model professional development program,» *School Science and Mathematics*, vol. 113, n.º 3, pp. 109–119, 2013.
- [20] A. Zollman, «Learning for STEM literacy: STEM literacy for learning,» *School Science and Mathematics*, vol. 112, n.º 1, pp. 12–19, 2012.
- [21] A. Eguchi, «Robotics as a learning tool for educational transformation,» em *Proceeding of 4th international workshop teaching robotics, teaching with robotics & 5th international conference robotics in education Padova (Italy)*, 2014, pp. 27–34.
- [22] J. Li, H. Luo, L. Zhao, M. Zhu, L. Ma e X. Liao, «Promoting STEAM education in primary school through cooperative teaching: A design-based research study,» *Sustainability*, vol. 14, n.º 16, p. 10 333, 2022.
- [23] P. C. Gonçalves, «Protótipo de um robô móvel de baixo custo para uso educacional,» *Programa de Pós-Graduação da CAPES. Universidade Estadual de Maringá*, 2007.
- [24] F. Martin, B. Mikhak, M. Resnick, B. Silverman e R. Berg, «To mindstorms and beyond,» *Robots for kids: Exploring new technologies for learning*, pp. 9–33, 2000.
- [25] C. Vidal-Silva, J. Serrano-Malebran e F. Pereira, «Scratch and arduino for effectively developing programming and computing-electronic competences in primary school children,» em *2019 38th International Conference of the Chilean Computer Science Society (SCCC)*, IEEE, 2019, pp. 1–7.

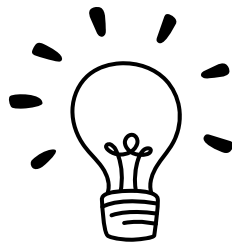
- [26] L. S. Vygotsky, *A Formação Social da Mente. Tradução: Monica Stahel M. da Silva*, 1998.
- [27] W. Frawley, M. A. G. Domingues e M. K. de Oliveira, *Vygotsky ea ciência cognitiva: linguagem e integração das mentes social e computacional*. Artmed, 2000.
- [28] V. Oscar, *A Zona de Desenvolvimento Proximal segundo Vygotsky*, jun. de 2022.
- [29] E. D. S. A. Marque e M. V. C. D. Carvalos, «Prática educativa bem-sucedida na escola: reflexões com base em LS Vigotski e Baruch de Espinosa,» *Revista Brasileira de educação*, vol. 22, 2017.
- [30] S. Ali, B. H. Payne, R. Williams, H. W. Park e C. Breazeal, «Constructionism, ethics, and creativity: Developing primary and middle school artificial intelligence education,» em *International workshop on education in artificial intelligence k-12 (eduai'19)*, 2019, pp. 1–4.
- [31] S. Papert, *Constructionism: A new opportunity for elementary science education*. Massachusetts Institute of Technology e Media Laboratory, 1986.
- [32] R. T. Wisnieski, L. J. Sverzut e B. Mendez, «A união do construcionismo e instrucionismo para o desenvolvimento de um jogo educativo.,» *Revista Ibero-Americana de Humanidades, Ciências e Educação*, vol. 8, n.º 4, pp. 1864–1871, 2022.
- [33] C. W. A. Carneiro, «Construcionismo e instrucionismo: Pedagogia em diálogo com a modernidade,» *identidade!*, vol. 24, n.º 2, pp. 69–78, 2020.
- [34] P. Freire, «Pedagogia do oprimido,» 1971.
- [35] T. M. Duffy e D. H. Jonassen, *Constructivism and the technology of instruction: A conversation*. Routledge, 2013.
- [36] S. Harlow, R. Cummings e S. M. Aberasturi, «Karl Popper and Jean Piaget: A rationale for constructivism.,» em *The Educational Forum*, Taylor & Francis, vol. 71, 2007, pp. 41–48.
- [37] S. Papert e I. Harel, *Constructionism: research reports and essays, 1985-1990*. Ablex publishing corporation, 1991.

- [38] A. Sharma, H. Dutt, C. N. V. Sai e S. M. Naik, «Impact of project based learning methodology in engineering,» *Procedia Computer Science*, vol. 172, pp. 922–926, 2020.
- [39] W. N. Bender, *Aprendizagem baseada em projetos: educação diferenciada para o século XXI*. Penso Editora, 2015.
- [40] L. Mutakinati, I. Anwari e Y. Kumano, «Analysis of Students’s Critical Thinking Skill of Middle School through STEM Education Project-Based Learning,» *Jurnal Pendidikan IPA Indonesia*, vol. 7, n.º 1, pp. 54–65, 2018.
- [41] M. J. S. Marin, E. F. G. Lima, A. B. Paviotti et al., «Aspectos das fortalezas e fragilidades no uso das metodologias ativas de aprendizagem,» *Revista brasileira de educação médica*, vol. 34, pp. 13–20, 2010.
- [42] T. Plomp et al., «Educational design research: An introduction,» *Educational design research*, pp. 11–50, 2013.
- [43] J. Van den Akker, «Principles and methods of development research,» em *Design approaches and tools in education and training*, Springer, 1999, pp. 1–14.
- [44] A. L. Brown, «Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings,» *The journal of the learning sciences*, vol. 2, n.º 2, pp. 141–178, 1992.
- [45] R. J. Shavelson, D. C. Phillips, L. Towne e M. J. Feuer, «On the science of education design studies,» *Educational researcher*, vol. 32, n.º 1, pp. 25–28, 2003.
- [46] D.-B. R. Collective, «Design-based research: An emerging paradigm for educational inquiry,» *Educational researcher*, vol. 32, n.º 1, pp. 5–8, 2003.
- [47] P. L. Lijnse, «“Developmental research” as a way to an empirically based “didactical structure” of science,» *Science education*, vol. 79, n.º 2, pp. 189–199, 1995.
- [48] H. Burkhardt, «From design research to large-scale impact: Engineering research in education,» *Educational design research*, pp. 121–150, 2006.

- [49] Q. Wang, C. L. Quek e X. Hu, «Designing and improving a blended synchronous learning environment: An educational design research,» *International Review of Research in Open and Distributed Learning*, vol. 18, n.º 3, pp. 99–118, 2017.
- [50] T. Anderson e J. Shattuck, «Design-based research: A decade of progress in education research?» *Educational researcher*, vol. 41, n.º 1, pp. 16–25, 2012.
- [51] M. Block, *mBot*, URL: <https://www.makeblock.com/steam-kits/mbot>, jan. de 2023.
- [52] A. I. Wang e R. Tahir, «The effect of using Kahoot! for learning—A literature review,» *Computers & Education*, vol. 149, p. 103818, 2020.
- [53] C. M. Plump e J. LaRosa, «Using Kahoot! in the classroom to create engagement and active learning: A game-based technology solution for eLearning novices,» *Management Teaching Review*, vol. 2, n.º 2, pp. 151–158, 2017.
- [54] A. de Escolas Emídio Garcia, *Plano de Ação para o Desenvolvimento Digital da Escola (PADDE)*, [urlhttp://padde.aeemidiogarcia.pt](http://padde.aeemidiogarcia.pt), mai. de 2023.
- [55] G. Onodipe, M. Robbins, G. Ayuninjam, T. Howse, A. Cottrell-Yongye e J. Curry-Savage, «Growth of Pedagogical Practice in an Active Multidisciplinary FLC on Flipped Learning.,» *International Journal for the Scholarship of Teaching and Learning*, vol. 14, n.º 2, p. 2, 2020.

Apêndice A

Projeto 1 - Blink

 Iniciante Intermediário Avançado

BLINK - PISCANDO MEU PRIMEIRO LED

COMO UTILIZAR OS EXEMPLOS PRONTOS AO NOSSO FAVOR

Neste projeto podemos verificar e aprender alguns conceitos básicos que irão nos acompanhar por toda nossa caminhada pelo mundo da robótica! Vamos aprender a utilizar o LED incorporado na placa e no final levaremos mais adiante e teremos um LED externo piscando no ritmo que escolhermos!

Vamos lá?

COMPETÊNCIAS QUE VOCÊ VAI APRENDER COM ESSE PROJETO:

Hardware

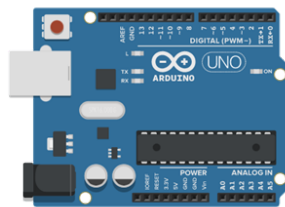
- Funcionamento LED
- Resistência
- Funcionamento Protoboard

Software

- Como carregar o sketch na placa
- Criação de variáveis
- Construção de laço

Você vai precisar de:

- 1 placa Arduino UNO



- Arduino IDE



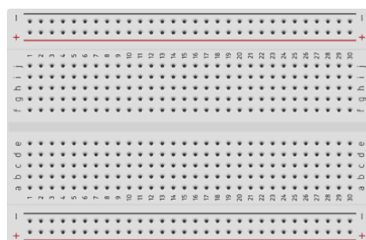
- 1 LED vermelho 5mm



- Resistência 150 ohms e 1/4 W

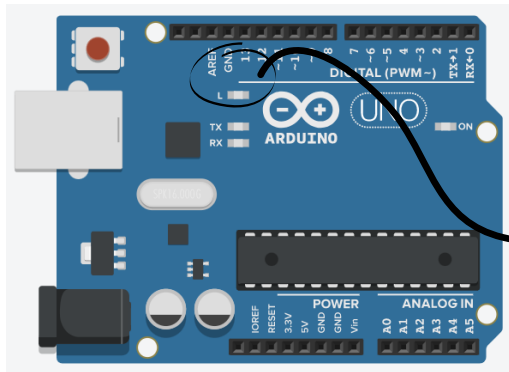


- Protoboard
(Breadboard)



PARTE 1 - Simulação do LED na placa

Você sabia que a placa Arduino já vem com um LED dentro dela? E além disso, podemos controlar como ele acende ou apaga? Vamos aprender juntos!



Posição do LED incorporado na placa!

Vamos primeiro simular no TINKERCAD (<https://www.tinkercad.com/>) para ver como funciona!

Para isso vamos precisar somente da placa Arduino UNO, e colocar o seguinte código para rodar:



O bloco de controle "para sempre" caracteriza nosso *loop*, e o código dentro do *loop* será rodado para sempre nesse caso

Neste código o LED incorporado está recebendo a informação para ligar, aguardar 1s e desligar.

Pronto! Agora é só iniciar a simulação e você verá o LED incorporado piscando!



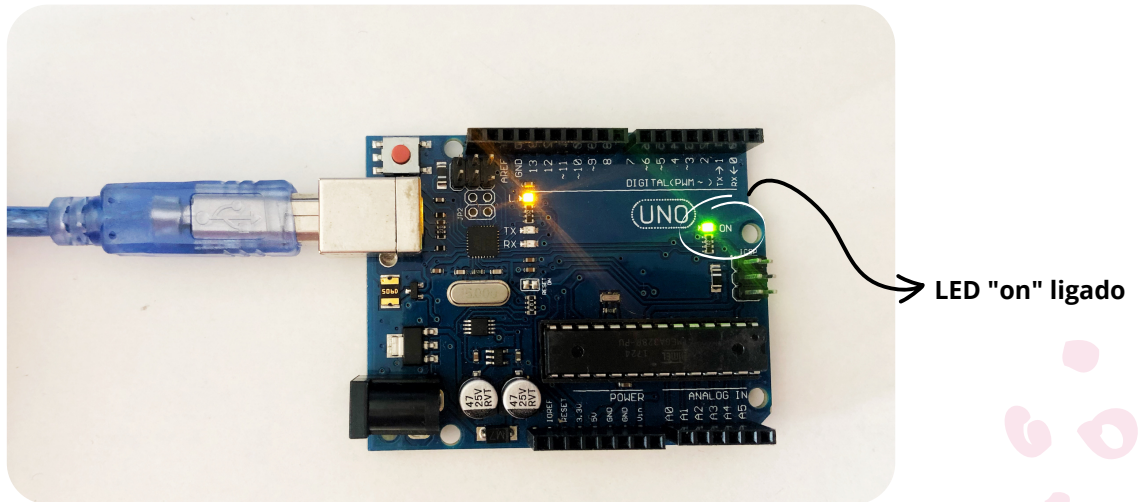
PENSE SEMPRE ALÉM!

O que acontece se aumentar o tempo que o LED deve aguardar para acender e apagar? Faça alguns testes para ver a diferença e discuta com os seus colegas os resultados!

PARTE 2 - Piscar o LED incorporado na placa real

Agora que você já viu na simulação nosso código funcionando, chegou a hora de conectar a placa Arduino UNO ao computador e vê-la funcionando ao vivo!

Para isso basta conectar, com o cabo USB, a placa ao computador. Quando ela estiver corretamente conectada a luz de "on" irá acender, ficará dessa forma:



Como estamos lidando com o Arduino UNO físico, e não mais a simulação, precisamos programar no ambiente próprio para nossa placa, o Arduino IDE.



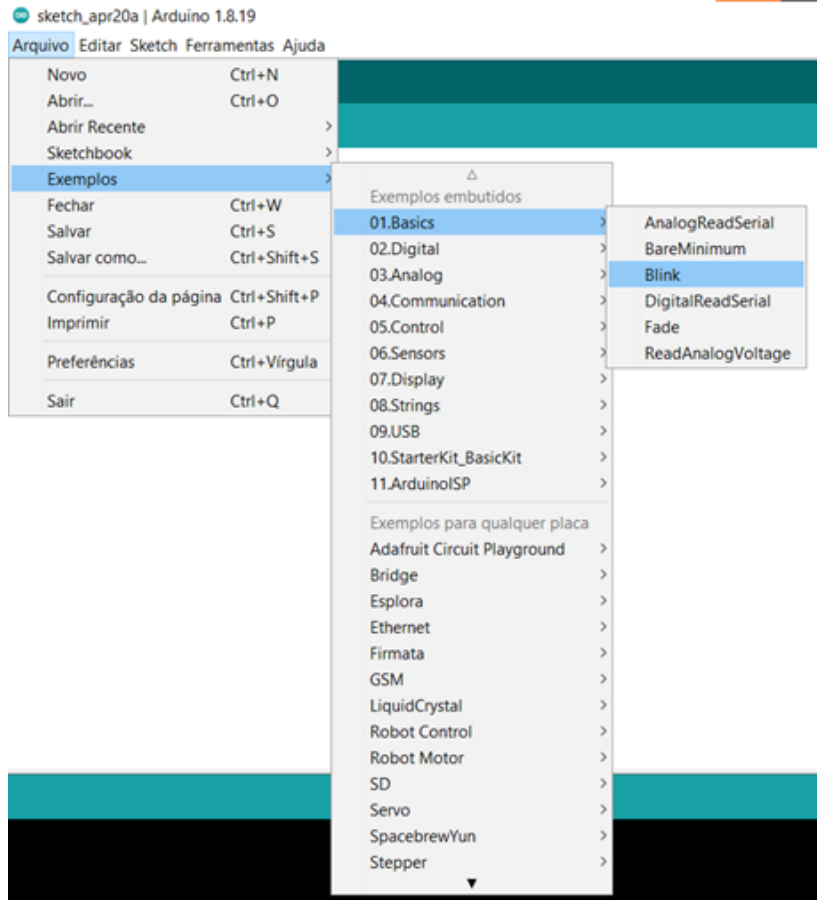
? Você sabia?

O Arduino IDE é o Software da plataforma Arduino, que nos permite fazer nosso código e ter uma conexão direta com a placa pelo cabo USB. O download pode ser feito diretamente no site da plataforma pelo link: <https://www.arduino.cc/en/software>

Com nosso ambiente de programação instalado no computador e pronto para usar, vamos aprender a carregar nosso primeiro sketch!

O Arduino IDE já possui alguns programas (sketches) de exemplos prontos. Podemos usar isso ao nosso favor, já que o sketch vem com a programação completa para uso. Vamos encontrar onde ficam esses exemplos! Siga o caminho da imagem a seguir:

Arquivo -> Exemplos -> 01. Basics -> Blink



Após clicar no sketch "Blink" uma nova janela irá abrir com o código do projeto. Ficando dessa forma:

A screenshot of the Arduino IDE showing the 'Blink' sketch. The title bar reads 'Blink | Arduino 1.8.19'. The menu bar includes 'Arquivo', 'Editar', 'Sketch', 'Ferramentas', and 'Ajuda'. The code editor contains the following text:

```
Turns an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
the correct LED pin independent of which board is used.
If you want to know what pin the on-board LED is connected to on your Arduino
model, check the Technical Specs of your board at:

This example code is in the public domain.

https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Vamos entender juntos a estrutura do código?

Diferente da programação em blocos que fizemos, esse código está escrito na linguagem de programação C++, a linguagem utilizada na plataforma Arduino.

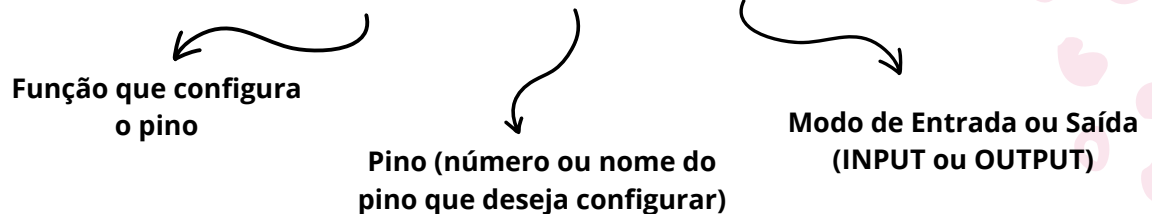
O código base sempre deve ter duas funções, a função *setup* e a função *loop*.

FUNÇÃO SETUP

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
```

A função *setup* roda sempre que o botão reset é acionado, ou quando a placa com o código é ligada. Nela buscamos inicializar as variáveis que serão utilizadas no projeto. No caso do projeto Blink, declaramos que nosso LED incorporado está sendo visto como uma saída.

Para isso foi usado a função `pinMode(LED_BUILTIN, OUTPUT);`



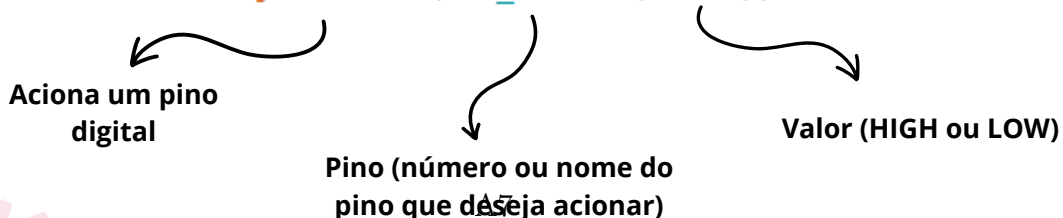
Dessa forma configuramos nosso LED incorporado (LED_BUILTIN) para ser um atuador (OUTPUT), já que ele deverá acender e apagar conforme o código.

FUNÇÃO LOOP


```
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

A função *loop*, como visto na programação em blocos, é nossa função "para sempre". É onde colocamos o código que será repetido indefinidamente.

Como enviaremos apenas sinal de HIGH e LOW (aceso e apagado, respectivamente), utilizaremos a função `digitalWrite(LED_BUILTIN, HIGH);`



Em resumo, no código Blink temos nosso LED incorporado da placa sendo configurado como uma saída na função *setup* e temos o mesmo recebendo uma informação na função *loop* para ligar (HIGH), aguardar um intervalo (delay) de 1000ms (1s) e desligar (LOW).

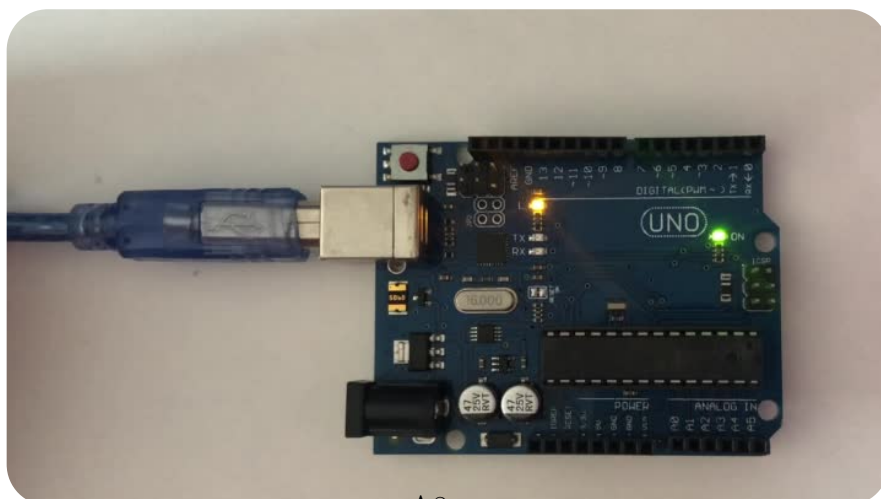
Vamos ligar a placa e carregar nosso sketch! Conecte sua placa e carregue o programa no botão  no canto superior esquerdo.

Após carregar o código na placa, ela deverá estar piscando o LED incorporado com intervalos de 1s, dessa forma:



PENSE SEMPRE ALÉM!

Agora vamos fazer o LED piscar mais rápido! Para isso você deve aumentar ou diminuir o tempo de delay? Faça alguns testes! Vamos tentar um delay de 200ms e ver o que acontece?



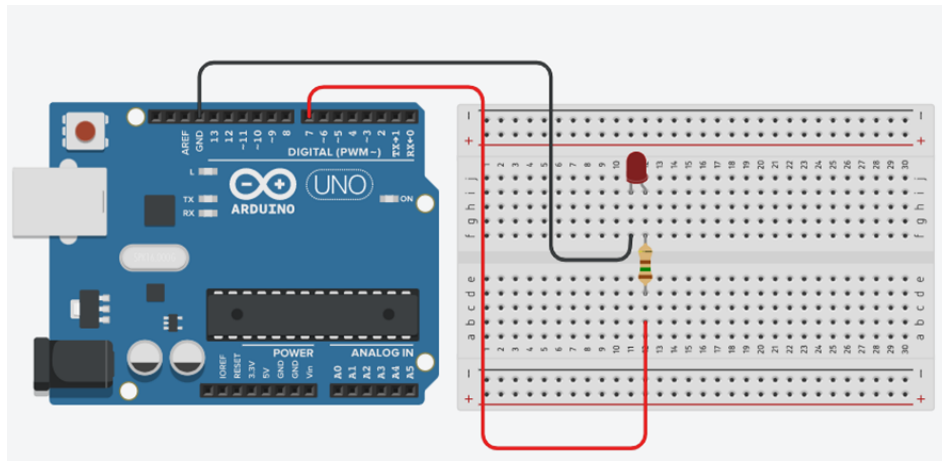
PARTE 3 - Piscar um LED externo

Para esse próximo passo iremos utilizar os materiais descritos no início do projeto! Então pegue seus equipamentos e vamos juntos!

HARDWARE

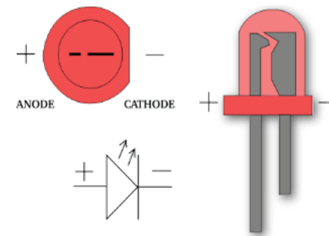
Para incluirmos um LED externo precisamos primeiro aprender alguns conceitos básicos de eletrônica, não queremos queimá-lo, certo?

A figura mostra o esquema de ligação que iremos utilizar para o LED externo:

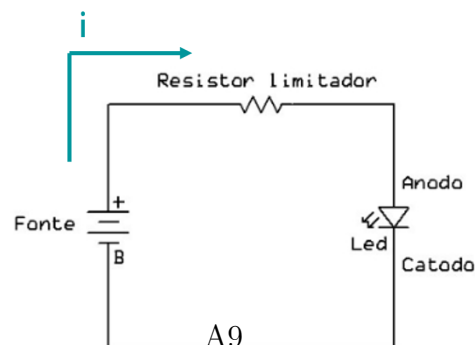


Para os LEDs acenderem, é necessário passar corrente elétrica por eles, e para isso o LED tem uma maneira correta de ligação, pois ele só acende quando a corrente elétrica corre no sentido certo.

No LED há dois terminais: o polo positivo, também chamado de ânodo (terminal mais comprido) e o polo negativo, também chamado de catodo (terminal mais curto).



Para que funcione corretamente, a corrente elétrica deve passar primeiro pelo polo positivo, porém temos que cuidar para não passar muita corrente (correndo o risco de queimá-los), nem pouca corrente (correndo o risco de não acendê-los), para isso utilizaremos resistências limitadoras. Assim temos nosso diagrama esquemático de ligação do LED:



Porém, como visto antes, não é qualquer resistência que podemos ligar. Para isso utilizaremos a Lei de Ohm para calcular qual podemos utilizar nesse caso.



Saída de alimentação Arduino – 5V

Alimentação do LED – 2V e 20mA



Resistor – Diminui a tensão no LED
Limita a corrente que passa por ele

$$V = R \cdot i$$
$$(5-2) = R \cdot 0,02$$
$$R = 150 \text{ ohm}$$

Como calculado, nesse projeto iremos utilizar uma resistência de 150Ω para um LED vermelho.

Como mostra no nosso diagrama, iremos utilizar o pino digital 7. Então faça todas as conexões para finalizar o Hardware e vamos partir para o Software!

SOFTWARE

Lembra que no início utilizamos o exemplo pronto “Blink” para piscar o LED incorporado? Podemos apenas modificá-lo para que agora pisque nosso LED externo. Para isso precisamos declarar uma variável mostrando para o programa onde está conectado nosso LED. Ficaremos com o sketch essa forma:

Blink §

```
int led = 7; //Mostra em qual pino digital está conectado o LED

void setup()
{
  pinMode(led, OUTPUT); //declara o LED como uma saída
}

// the loop function runs over and over again forever
void loop()
{
  digitalWrite(led, HIGH); // Liga o LED (HIGH é o nível alto de tensão)
  delay(1000); // aguarda um segundo
  digitalWrite(led, LOW); // Desliga o LED (LOW é o nível baixo de tensão)
  delay(1000); // aguarda um segundo
}
```

CRIAÇÃO DE NOVAS VARIÁVEIS

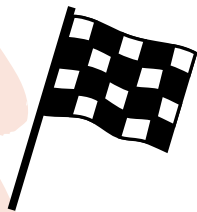
Para fazer o LED funcionar, o ligamos na porta digital 7. Porém é preciso colocar isso no código de uma forma que a placa processe e execute exatamente o que pretendemos. Para isso, no início do código foi criada uma variável (chamada "led") afim de armazenar em qual pino está ligado o LED.

```
int led = 7; //Mostra em qual pino digital está conectado o LED
```

Variável que aceita valores 8 bits

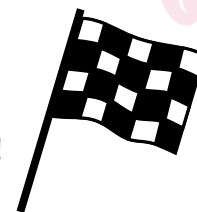
Nome escolhido da variável e qual a porta que está conectada

Pronto! Agora basta carregar o código na placa e vê-la funcionando!



PARABÉNS!

Você deu os primeiros passos no mundo da eletrônica!



DESAFIO


Você já prestou atenção nos postes de iluminação na rua? Eles não ficam ligados o dia todo, certo? Ligam em certa altura do dia, quando a luminosidade está começando a diminuir. Eles tem um *sensor* que permite que eles "percebam" quando a luz está diminuindo. Esse sensor é chamado LDR (Light Dependent Resistor). Pesquise sobre ele e programe um código capaz de ligar nosso LED quando tivermos em um ambiente escuro, e desligar quando estivermos em um ambiente mais claro!



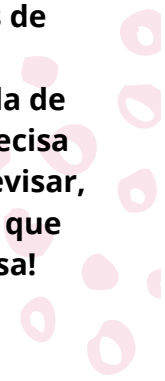
GOTAS DE CONHECIMENTO

Nesse primeiro projeto aprendemos vários conceitos, mas sempre podemos aprofundar nossos conhecimentos. Para isso foram feitos módulos, chamados de "Gotas de conhecimento" que irão explicar mais detalhadamente alguns desses conceitos:

-  Módulo 1 - Introdução ao Hardware e Software
-  Módulo 2 - Comentários e variáveis
-  Módulo 3 - Grandezas digitais e analógicas
-  Módulo 4 - Funções
-  Módulo 5 - Breadboard
-  Módulo 6 - Resistências
-  Módulo 7 - LEDs



Use e abuse das gotas de conhecimento!
Sempre que tiver dúvida de algum conceito, não precisa voltar no projeto para revisar, basta ir direto na Gota que tem o que você precisa!



Apêndice B

Módulo 1 - Introdução ao *Hardware* e *Software*



INTRODUÇÃO AO HARDWARE E SOFTWARE DA PLATAFORMA ARDUINO

PARTE 1 - Conhecendo o Arduino

Antigamente, para desenvolver um circuito interativo, era necessário conhecimentos aprofundados de engenharia, e para cada projeto deveria ser desenvolvido um produto específico para aquele problema. A prototipagem era algo muito caro e trabalhoso, por isso o projeto demandava muito estudo crítico e planejamento.

Porém, com o surgimento dos microcontroladores, os problemas que antes eram resolvidos via Hardware, podiam agora ser tratados via Software. Pensando nisso, um grupo de 5 pesquisadores italianos criou um dispositivo capaz de criar esses sistemas de forma mais simples, e assim foi criado a plataforma Arduino.

Massimo Banzi
David Cuartielles
Tom Igoe
Gianluca Martino
David Mellis

Segundo o site oficial do Arduino:

“Arduino é uma plataforma open-source de prototipagem eletrônica com hardware e software flexíveis e fáceis de usar, destinado a artistas, designers, hobbistas e qualquer pessoa interessada em criar objetos ou ambientes interativos.”

? Mas o que significa ser *Open Source*?

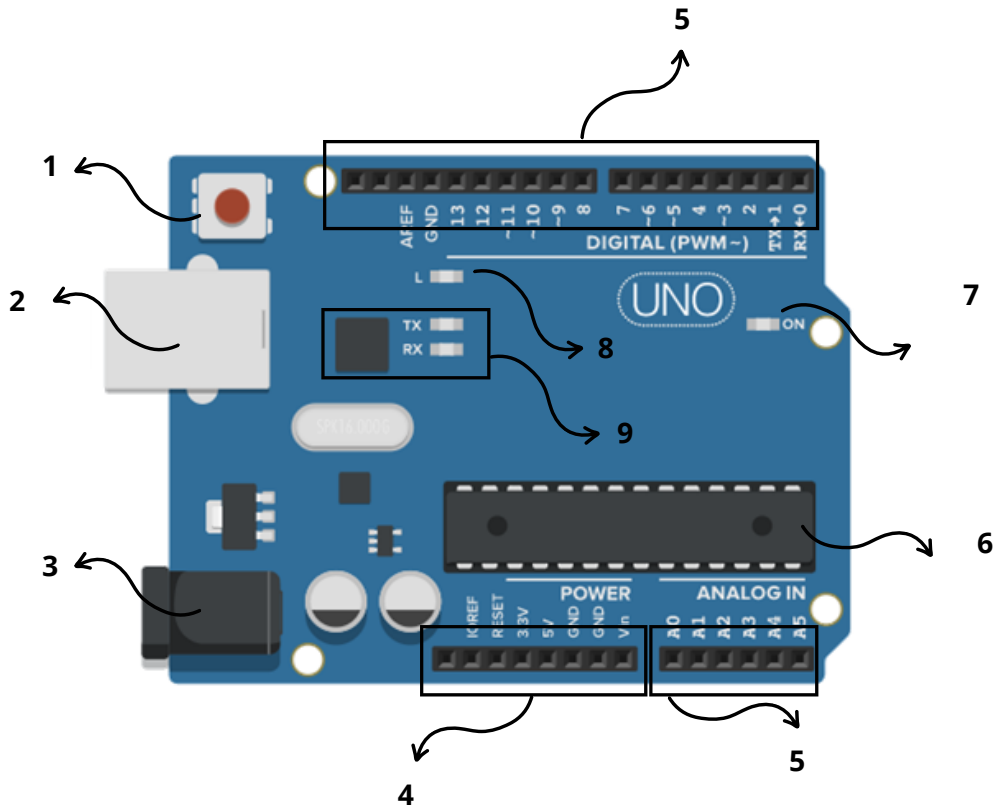
Significa que todas as partes que integram a plataforma podem ser usadas, modificadas e distribuídas gratuitamente e livremente, desde que sejam licenciadas. Uma grande vantagem, já que permite a colaboração de uma grande comunidade. Essa característica permite que seja uma plataforma de baixo custo, já que podem ser vendidas cópias sem o pagamento de royalties.

Agora iremos aprofundar nosso conhecimento no Hardware e no Software!

HARDWARE

A plataforma apresenta diversas placas para níveis do iniciante ao avançado. No nível iniciante, encontram-se placas, como a placa Arduino UNO, Arduino Leonardo, Arduino NANO e Arduino Micro. Entre elas, foi escolhida a placa Arduino UNO

O hardware da placa Arduino UNO é composto por:



1 - Botão de reset: Reinicia o Arduino e roda o último código carregado novamente.

2 - Conector USB: Entrada por onde a placa e o computador se comunicam com o auxílio de um cabo USB tipo A e tipo B.

3 - Conector de alimentação externa: Entrada de alimentação externa que recebe um mínimo de 7 Volts e um máximo de 20 Volts com uma corrente de entrada mínima de 300 Amperes. A tensão recomendada é de 9 Volts. Se a placa estiver alimentada com o cabo USB também, a alimentação externa terá preferência automaticamente.

4 - Pinos de alimentação: Pinos que fornecem tensões pré-estabelecidas (5 e 3 Volts) para os componentes conectados à placa. Deve-se tomar cuidado para não sobrecarregar a corrente suportada da placa.

5 – Pinos de entrada e saída: Pinos para conexão dos componentes que podem ser entradas ou saídas, dependendo da programação. São 14 portas digitais (I/O), 6 portas de entrada analógicas e 6 portas de saída analógicas (PWM).

6 – Microcontrolador ATmega328: O computador do Arduino. Dispositivo responsável por processar o código que enviamos para a placa, é o cérebro do nosso Arduino.

7 – LED de alimentação: LED para indicar se a placa está energizada.

8 – LED incorporado: LED já presente na placa ligado ao pino 13. Pode ser acionado diretamente pelo comando LED_BUILTIN ou acionando diretamente o pino 13.

9 – LEDs TX e RX: indicam que o Arduino está trocando informações. Quando está transmitindo o LED TX está ativado, e quando está recebendo o LED RX está ativado.

HARDWARE

Em todos os nossos projetos iremos utilizar o software Arduino IDE (Integrated Development Environment ou Ambiente de Desenvolvimento Integrado). Para isso vamos instalar o Software e aprender como cada parte funciona.

DOWNLOAD DO ARDUINO IDE

O download pode ser feito através do site oficial de forma gratuita.



<https://www.arduino.cc/en/software>

Basta escolher qual seu sistema operacional e fazer o download do programa indicado para o mesmo. Neste módulo iremos tratar do download para o Windows Win 7 and newer.

Downloads



Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer

Windows ZIP file

Windows app Win 8.1 or 10

Linux 32 bits

Linux 64 bits

Linux ARM 32 bits

Linux ARM 64 bits

Mac OS X 10.10 or newer

[Release Notes](#)

[Checksums \(sha512\)](#)

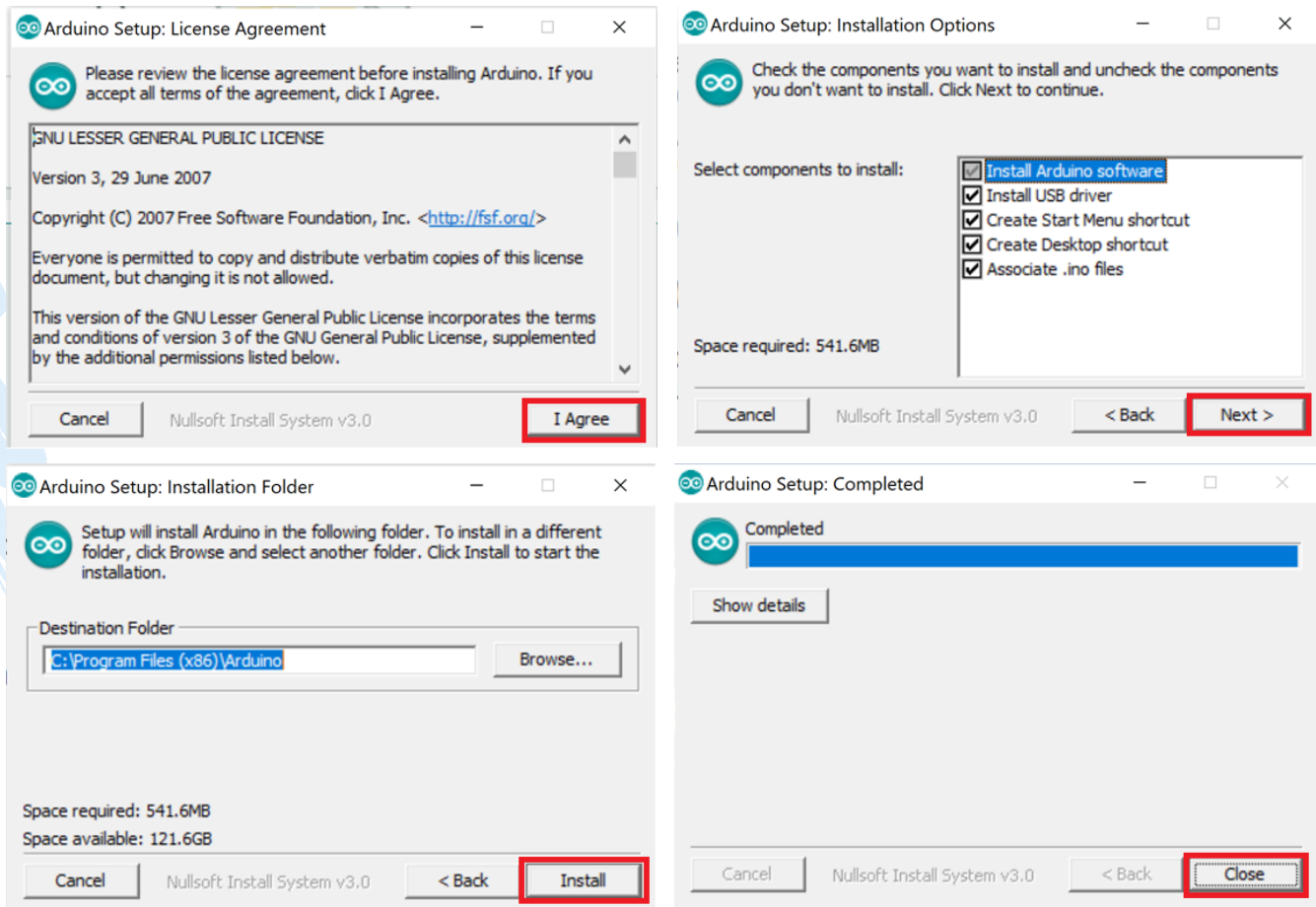
Após escolher o download, irá aparecer a tela para doação (caso queira) de uma certa quantia para apoiar a plataforma. Não é obrigatório.

Support the Arduino IDE

Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **64 696 655** times — impressive! Help its development with a donation.

[Learn more about donating to Arduino.](#)

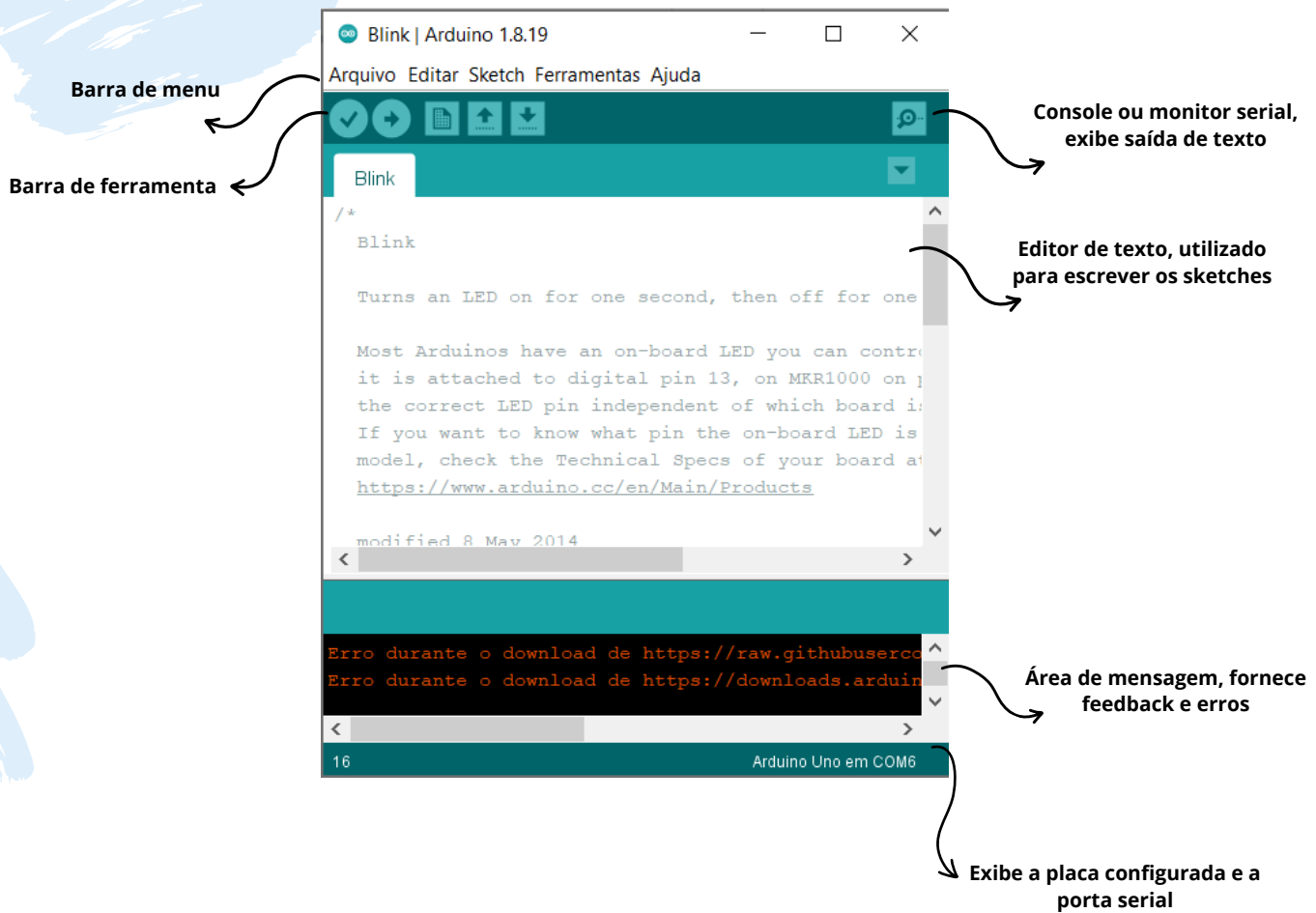
Logo depois irá iniciar o download de um executável, basta clicar em cima dele para dar início ao processo de instalação e seguir os passos:



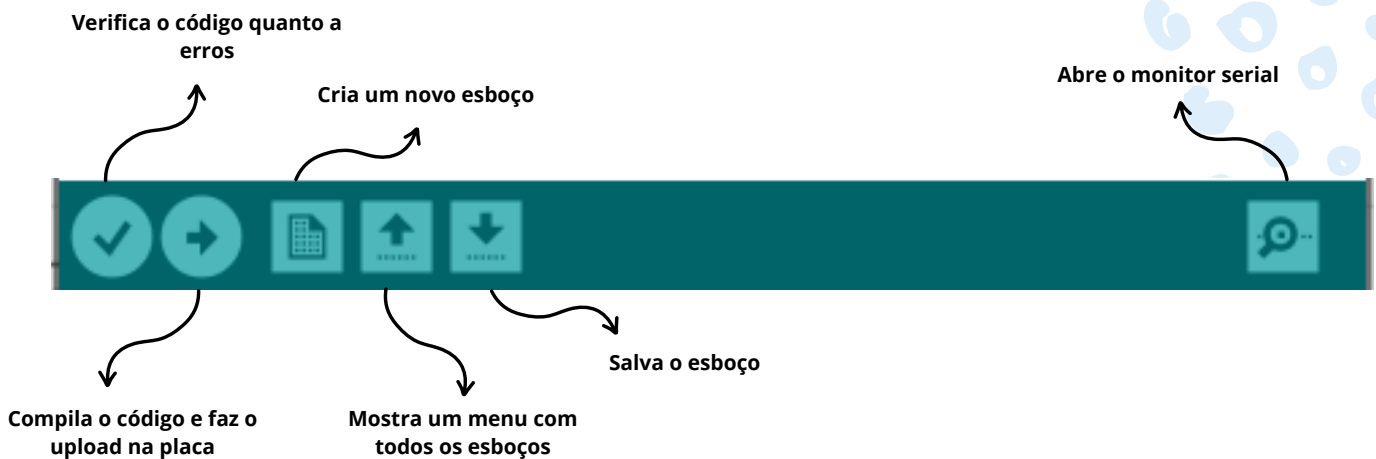
Uma vez que o download for finalizado, aparecerá na tela inicial o ícone do Arduino IDE.

PARTE 2 - ENTENDENDO O ARDUINO IDE

Este software tem como objetivo ser de fácil entendimento e utilização. Consiste em um editor de texto destinado a escrita do código, uma área de mensagens afim de se comunicar com o utilizador, um console de texto (monitor serial), uma barra de ferramentas e alguns menus.



Na barra de ferramentas temos as seguintes:



Agora que já vimos nossa barra de ferramentas, vamos ver algumas funções dos menus.

MENU ARQUIVO

Arquivo	Editar	Sketch	Ferramentas	Ajuda
Novo				Ctrl+N
Abrir...				Ctrl+O
Abrir Recente				>
Sketchbook				>
Exemplos				>
Fechar				Ctrl+W
Salvar				Ctrl+S
Salvar como...				Ctrl+Shift+S
Configuração da página				Ctrl+Shift+P
Imprimir				Ctrl+P
Preferências				Ctrl+Vírgula
Sair				Ctrl+Q

Arquivo	Função
Novo	Cria um novo esboço, já com a estrutura básica do código
Abrir	Abre um arquivo previamente salvo nos diretórios
Abrir recente	Abre uma lista dos arquivos salvos recentemente
Sketchbook	Mostra uma lista dos esboços salvos no sketchbook. Clicar no nome do arquivo abre uma nova janela.
Exemplos	Abre uma lista dos exemplos fornecidos pelo software, incluindo bibliotecas prontas.
Fechar	Fecha a janela atual
Salvar	Salva o esboço com o nome atual. Caso não esteja salvo abre a janela "Salvar como"
Salvar como	Possibilita salvar o esboço atual com um novo nome
Configuração da página	Abre a página de configuração da página de impressão
Imprimir	Envia o esboço completo para a impressora conforma as configurações
Preferências	Abre a janela onde podem ser feitas algumas configurações, inclusive de idioma.
Sair	Fecha todas as janelas abertas. Da próxima vez que o Arduino IDE for inicializado todas as janelas fechadas por esse botão serão abertas novamente.

MENU EDITAR

Editar	
Desfazer	Ctrl+Z
Refazer	Ctrl+Y
Cortar	Ctrl+X
Copiar	Ctrl+C
Copiar para Fórum	Ctrl+Shift+C
Copiar como HTML	Ctrl+Alt+C
Colar	Ctrl+V
Selecionar tudo	Ctrl+A
Vá para a linha...	Ctrl+L
Comentar/descomentar	Ctrl+Barra
Aumentar indentação	Guia
Diminuir indentação	Shift+Guia
Aumentar Tamanho da Fonte	Ctrl+Mais
Diminuir Tamanho da Fonte	Ctrl+Menos
Localizar...	Ctrl+F
Localizar próximo	Ctrl+G
Localizar anterior	Ctrl+Shift+G

Editar	Função
Desfazer/Refazer	Volta as etapas anteriores da edição com Desfazer e avança com Refazer
Recortar	Remove o texto selecionado e coloca na área de transferência
Copiar	Copia o texto selecionado para a área de transferência
Copiar para fórum	Copia o código completo do esboço para o formato ideal para postagem em fórum, com coloração e sintaxe
Copiar como HTML	Copia o código completo no formato HTML, ideal para incorporação em páginas da web.
Colar	Cola o conteúdo da área de transferência no local do cursor.
Select All	Seleciona todas as linhas do esboço.
Comentar/Descomentar	Adiciona ou remove o marcado de comentário //
Aumentar/Diminuir recuo	Adiciona ou remove o espaço no início da linha selecionada.
Localizar	Abre a janela localizar, utilizada para pesquisar um texto dentro do esboço atual.
Localizar próximo	Destaca o próximo texto colocado na janela Localizar, tomando como referência a posição do cursor.
Localizar anterior	Destaca o texto anterior colocado na janela Localizar, tomando como referência a posição do cursor.

MENU SKETCH

Sketch Ferramentas Ajuda

Verificar/Compilar	Ctrl+R
Carregar	Ctrl+U
Carregar usando programador	Ctrl+Shift+U
Exportar Binário compilado	Ctrl+Alt+S
<hr/>	
Mostrar a página do Sketch	Ctrl+K
Incluir Biblioteca	>
Adicionar Arquivo...	

Sketch	Função
Verificar/Compilar	Verifica o esboço afim de encontrar erros. Mostra na área de mensagens o uso de memória e variáveis.
Upload	Compila o arquivo e carrega na placa configurada
Carregar usando programador	Sobrescreve o bootloader na placa. Permite que toda a memória Flash seja usada para o esboço.
Exportar binário compilado	Salva o arquivo na extensão .hex
Mostrar a página do Sketch	Abre a pasta do esboço atual
Incluir biblioteca	Permite adicionar uma biblioteca ao seu esboço. Pode também acessar o gerenciador de bibliotecas para importar arquivos .zip
Adicionar arquivo	Inclui um arquivo adicional a uma subpasta do esboço. Esse arquivo não será compilado, então não fará parte do esboço.

MENU FERRAMENTAS

Ferramentas	Ajuda
Autoformatação	Ctrl+T
Arquivar Sketch	
Corrigir codificação e recarregar	
Gerenciar Bibliotecas...	Ctrl+Shift+I
Monitor serial	Ctrl+Shift+M
Plotter serial	Ctrl+Shift+L
WiFi101 / WiFININA Firmware Updater	
Placa: "Arduino Uno"	>
Porta	>
Obter informações da Placa	
Programador: "AVRISP mkII"	>
Gravar Bootloader	

Ferramentas	Função
Formatação automática	Corrige a indentação do código. Deixa as chaves alinhadas e o recuo padronizado.
Arquivar sketch	Cria um arquivo .zip do esboço atual. Será salvo no mesmo diretório que o esboço se encontra.
Corrigir codificação e recarregar	Corrige discrepâncias de caracteres entre a codificação do mapa de caracteres do editor e os mapas de caracteres de outros sistemas operacionais. Usado para corrigir os erros '\302' e '\304'
Monitor Serial	Abre a janela do monitor serial e começa a troca de informações com a placa conectada.
Placa	Selecionar a placa que está sendo utilizada.
Porta	Mostra os dispositivos seriais conectados à máquina. Normalmente é atualizado automaticamente quando uma nova placa é conectada.
Programador	Permite selecionar um programador de hardware caso não vá utilizar a conexão serial USB. Utilizado para gravar um bootloader em um novo microcontrolador.
Burn bootloader	Permite gravar um bootloader no microcontrolador em uma placa Arduino. Não é necessário utilizar normalmente, apenas quando for utilizar um microcontrolador que não vem com bootloader (um ATmega por exemplo)

MENU AJUDA

Uma ferramenta muito importante para quem está iniciando nesse mundo do Arduino. Tem acesso a inúmeros documentos disponibilizados pela plataforma, todos disponibilizados localmente, sem necessidade de acesso à internet.

Ajuda

- Iniciando
- Ambiente
- Resolução de problemas
- Referência

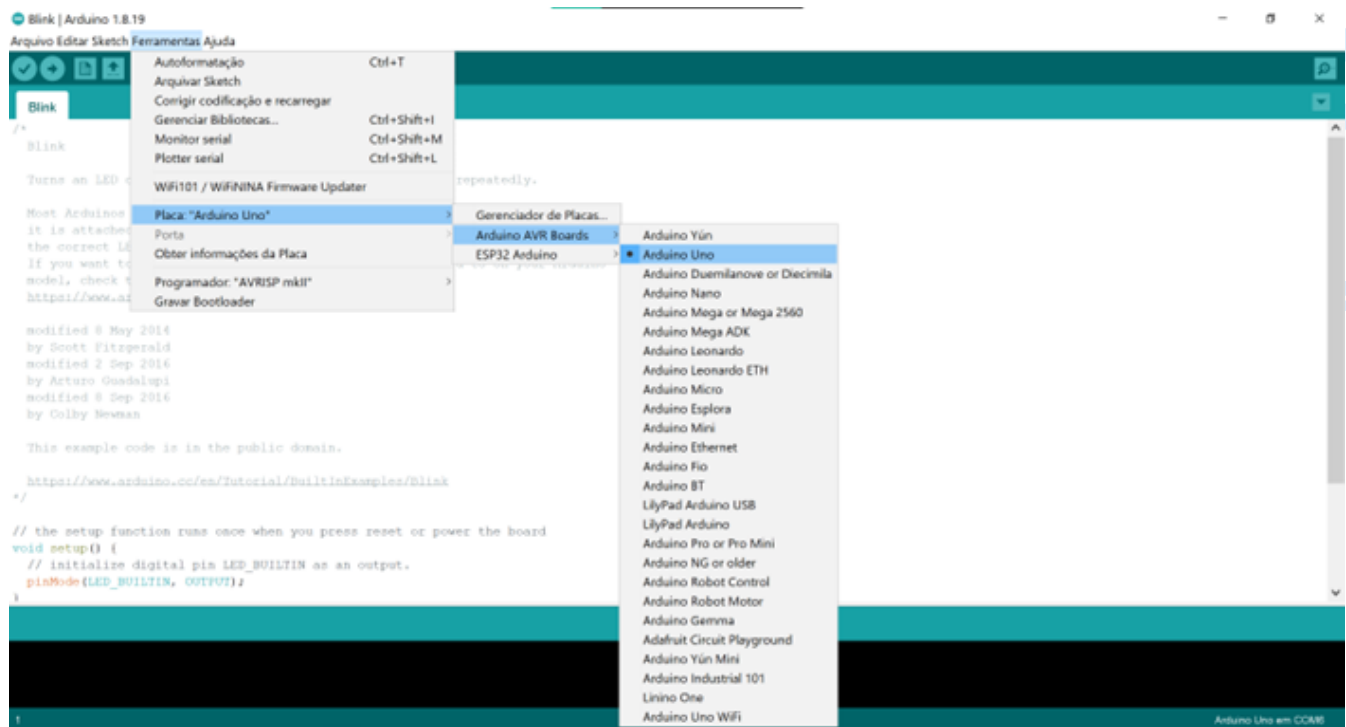
- Procurar na Referência Ctrl+Shift+F
- Perguntas Frequentes - FAQ
- Visite Arduino.cc

- Sobre Arduino

PASSO 3 - COMO CARREGAR O SKETCH NA PLACA

O software Arduino IDE consiste em um local padrão afim de armazenar os programas, chamado de sketchbook. Quando o programa for rodado pela primeira vez, será criado automaticamente um diretório para o sketchbook.


Para enviar corretamente o sketch para a placa é necessário selecionar qual a placa que você está utilizando. Conecte o cabo USB na placa e selecione Ferramentas > Placa e então selecione a placa que está sendo usada. No nosso caso será o Arduino UNO, mas são várias as opções de placas.





Após selecionar a placa e a porta serial correta, basta pressionar o botão de upload diretamente na barra de ferramentas, ou pelo menu Sketch -> Upload.

Com isso feito, os LEDs RX e TX piscarão enquanto o código é enviado para a placa. Uma vez que o código foi carregado, o software indicará na área de mensagem que o upload foi concluído, ou se houve algum erro.



Quando o sketch é carregado na placa, está sendo utilizado o bootloader do Arduino, um programa que foi carregado no microcontrolador da placa utilizada, que permite carregar o código sem precisar utilizar hardwares adicionais. Ele fica ativo por alguns segundos e depois inicia o sketch que foi carregado mais recentemente. O LED incorporado irá piscar quando iniciar, ou seja, quando a placa reiniciar.



Apêndice C

Módulo 2 - Comentários e variáveis no código



COMENTÁRIOS E VARIÁVEIS NO CÓDIGO

PARTE 1 - Entendendo o programa

O cérebro do Arduino, o microcontrolador ATmega328 processa informações a partir de uma lista de tarefas, pois não tem a inteligência da mesma forma do Ser Humano.

Por isso todas as ações que desejamos que o Arduino desempenhe devem ser especificadas de uma forma que ele entenda o que deve ser feito, utilizando uma espécie de passo a passo, chamado Algoritmo.

Por exemplo, para fazer uma gelatina precisamos seguir um um passo a passo, uma receita

Vamos ver como seria o algoritmo para fazer gelatina.

//RECEITA PARA GELATINA

INÍCIO

Passo 1: Separar os ingredientes

Ingredientes:

1 pacote de gelatina do sabor de preferência;

1 copo de água quente;

1 copo de água gelada;

Modo de preparo:

Passo 2: Encher um copo com água;

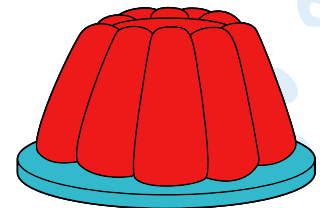
Passo 3: Aquecer o copo com água;

Passo 4: Colocar a água quente em um recipiente maior;

Passo 5: Misturar a gelatina na água quente;

Passo 6: Misturar a água gelada no recipiente;

Passo 7: Colocar para gelar por 3 horas;



Assim como precisamos no algoritmo, essa receita define no início o que iremos utilizar e possui passos bem definidos. Tem uma ordem clara de como devem ser os processos, de uma forma que o passo 5 não pode ser feito antes do passo 4.

No Arduino não é diferente, também precisamos definir o que iremos utilizar e estruturar bem os passos que deve seguir para chegar ao objetivo final. Além disso, para o microcontrolador entender, os passos devem ser escritos em uma linguagem compatível com o computador. No caso do Arduino é a linguagem C++, e assim como qualquer língua, possui a forma correta de se escrever e regras que devem ser seguidas.

PARTE 2 - CONCEITOS INICIAIS LINGUAGEM C++

COMENTÁRIOS

Pelo Arduino ser uma plataforma open source, existem muitos códigos prontos disponíveis. Porém, na maioria das vezes é complicado entender o código que outra pessoa fez, pois pensamos de formas diferentes. Além disso, é importante colocar notas (comentários) durante o código para que nós mesmos possamos entender o que foi feito. Por esse motivo temos a ferramenta de comentários, onde permite colocar texto no corpo do código sem ser considerado quando o código for rodado.



ATENÇÃO!

Sempre faça comentários no código



O comentário sempre fica em cinza, como no exemplo abaixo:

```
void setup() {  
  Serial.begin(115200);  
  //configura o pino do LED como saída  
  pinMode(LED, OUTPUT);  
}
```

Temos duas formas de fazer comentários no Arduino IDE, comentando uma linha ou comentando um bloco de texto. Para comentar um bloco de texto é o usado o `/*` no início do texto e `*/` no final.

```
/*Para fazer comentários em bloco de texto  
basta utilizar os marcadores no início e no final do texto */
```

Já para comentar uma linha, basta colocar `//`. Não necessariamente precisa ser no início da linha. Por exemplo:

```
digitalWrite(LED, HIGH); //liga o LED
```

O comentário é muito utilizado quando queremos testar apenas uma parte do programa. Dessa forma não precisamos deletar o resto, basta comentá-lo e colocar para rodar. Assim as linhas comentadas serão ignoradas pelo Arduino, podendo testar apenas a parte que queremos.

VARIÁVEIS

Sempre que estamos programando precisamos guardar algumas informações que serão usadas posteriormente. Por exemplo, caso queira fazer a soma de dois números, o resultado deve ser armazenado em algum lugar. Esse “lugar” é chamado de variável, e podemos chamá-la como achamos melhor, mas sempre escolhendo algo que faça sentido e que possa ser facilmente entendido posteriormente. Nesse exemplo poderíamos chamar nossa variável de “x” ou até de “soma”. Ex:

```
//variável 'soma' recebe o valor da soma de 1+1  
soma = 1+1
```

Podemos pensar na memória do Arduino como um conjunto de caixinhas, onde cada caixinha deve ter um nome e a mesma guarda uma informação. Cada caixinha é uma variável.

Quando damos um nome a uma variável, além de nomeá-la precisamos especificar que tipo de conteúdo vai ser guardado nela. No caso da nossa soma, será uma variável do tipo int, pois ela receberá um valor de até 8 bits. Devemos fazer isso antes de utilizar a variável. Ficando dessa forma:

```
int soma; //declara a variável soma do tipo inteiro  
soma = 1+1 //variável 'soma' recebe o valor da soma de 1+1
```

DICA: É sempre bom escolher o nome da variável relacionada com o tipo de informação que ela está armazenando. Assim como chamamos nosso resultado da soma de “soma”, faz sentido.

Tipo de dado	RAM	Intervalo numérico
Boolean	1 byte	0 a 1 (TRUE or FALSE)
int	2 bytes	-32.768 a 32.767
unsigned int	2 bytes	0 a 65.535
word	2 bytes	0 a 65.535
char	1 byte	-128 a 127
unsigned char	1 byte	0 a 255
byte	1 byte	0 a 255
void keyword	N/A	N/A
long	4 bytes	-2.147.483.648 a 2.147.483.647
unsigned long	4 bytes	0 a 4.294.967.295
float	4 bytes	-3,4028236e+38 a 3,4028236e+38
double	4 bytes	-3,4028236e+38 a 3,4028236e+38
string	1 byte + x	Sequência de caracteres
array (vetor)	1 byte + x	Sequência de variáveis

No site oficial do Arduino também tem muito conteúdo sobre as variáveis. Pode acessar em:

<https://www.arduino.cc/reference/pt/>

Apêndice D

Módulo 3 - Grandezas digitais e analógicas



GRANDEZAS DIGITAIS E ANALÓGICAS

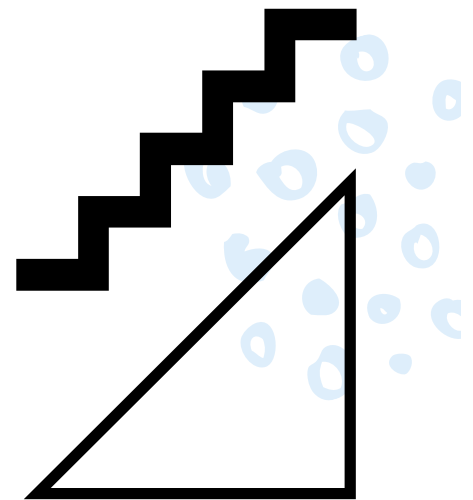
Um dos conceitos fundamentais utilizados na programação é a diferença entre grandezas analógicas e grandezas digitais, já que o Arduino nos dá a possibilidade de manipular ambas.

A maior diferença entre as duas grandezas é a forma como variam. A grandeza digital não varia continuamente no tempo, e sim em saltos de valores bem definidos. Já as grandezas analógicas é justamente o contrário, pois variam continuamente dentro de uma faixa de valores.



Para melhor entendimento vamos pensar nos termômetros digitais e nos termômetros de mercúrio. Apesar da temperatura variar continuamente, o visor do termômetro digital varia, normalmente, de um em um grau Celsius. Dessa forma, um termômetro desse não irá mostrar uma temperatura de 26,7 graus, pois para ele só existe 26 e 27 graus. Os valores intermediários não estão definidos. Já em um termômetro analógico, a altura da coluna de mercúrio varia continuamente com o aumento da temperatura.

Outra analogia que podemos fazer é entre uma escada e uma rampa. Enquanto subimos a rampa, a distância entre a base e o topo aumenta continuamente, já quando subimos uma escada, a distância entre a base e o topo aumenta em saltos, com os valores definidos sendo a altura dos degraus. Nesse caso a altura percorrida na rampa seria uma grandeza analógica, e a altura percorrida na escada uma grandeza digital. Quanto menores os degraus e mais degraus tivermos na escada do mesmo tamanho, mais ela se aproxima a uma rampa.



No caso de circuitos e equipamentos elétricos digitais, apenas dois valores de tensão são definidos. Um nível lógico alto, e um nível lógico baixo. No caso do Arduino tem 5V como alto e 0V como baixo. Já os analógicos trabalham em uma certa faixa de valores, chamada de range, que no Arduino varia de 0V a 5V.

ENTRADAS E SAÍDAS DIGITAIS

Os pinos digitais do Arduino podem ser configurados como entrada ou saída, a partir do comando `pinMode()`. Se o pino for configurado como entrada (INPUT), utilizando a função `digitalWrite()` pode-se ativar (HIGH) ou desativar (LOW) o pino. Se o pino for configurado como saída (OUTPUT), terá a tensão de 5V para HIGH e 0V para LOW. O pino irá fornecer ou drenar corrente, que varia de placa para placa, mas no Arduino UNO é de 30mA. Se por um acaso uma corrente maior percorrer pelo pino o mesmo poderá ser danificado.

ENTRADAS E SAÍDAS ANALÓGICAS

O Arduino possui um total de 6 entradas analógicas e 5 saídas PWM, que simulam uma saída analógica.

Como todas as informações são processadas de forma digital, é necessário fazer a conversão digital -> analógica e analógica -> digital, e para isso existem conversores embutidos na placa.

Cada conversor possui uma resolução, o qual podemos fazer um paralelo com o exemplo da rampa e da escada, onde a quantidade de degraus pode ser comparada ao conceito de resolução.

A resolução condiz com quantos degraus ele pode representar um valor, quanto mais intervalos maior a resolução, chegando mais perto de um sinal analógico. Essa resolução é expressa em bits. No caso do Arduino é um conversor de 10 bits, ou seja consegue representar um valor com $2^{10} = 1024$ intervalos (degraus).

ENTRADAS ANALÓGICAS

Tendo a entrada digital do Arduino de 0V a 5V e um conversor 10 bits, nosso intervalo de referência será dividido em 1024 degraus. Ou seja, se uma leitura analógica for de 4,12V o valor da conversão será de:

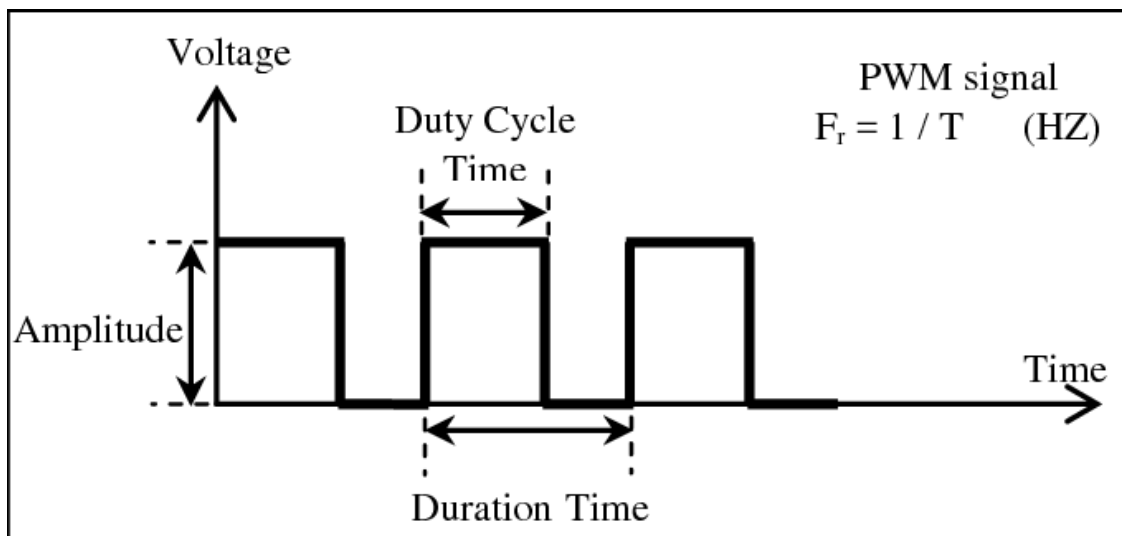
$$4,12 * (1024/5) = 843,8$$

Como o conversor só consegue representar números inteiros, devemos arredondar o valor para 844 por ser o degrau mais próximo. O valor 844 (digital) representa a tensão de 4,121953125 V contando com o erro de 0,001953125 da medida pela limitação da resolução.

SAÍDAS PWM (Pulse Width Modulation)

Essa é uma técnica utilizada para obter resultados analógicos a partir de meios digitais. Utiliza a modulação por largura de pulso de uma onda quadrada com alta frequência. Onde podemos controlar a saída a partir do controle do tempo em que a onda permanece em nível lógico alto.

Com o controle do tempo, chamado de Duty Cycle, podemos controlar o valor médio de tensão de saída, com valores indo de 0V (0% de Duty Cycle) a 5V (100% de Duty Cycle).



O valor do Duty Cycle é obtido a partir da razão entre o tempo em nível lógico alto e o tempo total da onda

$$Duty\ Cycle\ (\%) = \left(\frac{x}{x + y} \right) * 100\% = \frac{x}{T} * 100\%$$

$$V_{médio} = V_{máx} * Duty\ Cycle\ (\%)$$

Diferente da entrada analógica que utiliza um conversor 10 bits, o valor do Duty Cycle é um valor inteiro armazenado em 8 bits, ou seja, vai de 0 (0V) a 255 (5V).

Apêndice E

Módulo 4 - Funções

FUNÇÕES



Suponha que suas ações sejam programadas pela linguagem C++, e precise executar tarefas durante o dia. Agora pense em uma tarefa que você faz várias vezes durante o dia. Tomar um copo de água por exemplo. Para isso deve desempenhar uma lista de tarefas. Desta forma:

```
Pegar um copo de água;  
Encher o copo com água;  
Tomar a água;
```

Como durante o dia provavelmente você toma vários copos de água, para não precisar descrever toda a lista de tarefas novamente, podemos simplesmente dar um nome para esse conjunto. Por exemplo:

```
void tomarAgua()  
{  
    Pegar um copo de água;  
    Encher o copo com água;  
    Tomar a água;  
}
```

Desta forma, toda vez que quiser tomar água basta colocar o nome da função, sem precisar descrever toda a lista novamente.

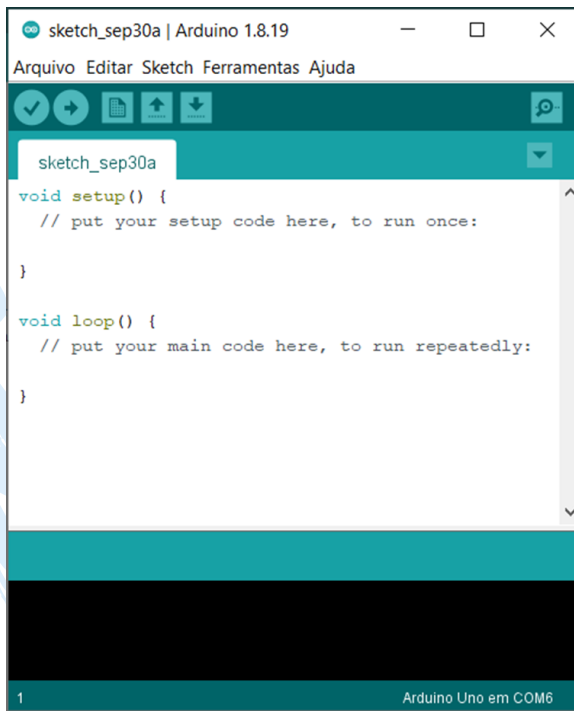
Resumindo, uma função é um subprograma que contém um conjunto de tarefas que será executada quando chamada pelo programa principal.

Todas as funções devem ter um nome, preferencialmente compatível com a tarefa que irá executar, e um tipo. No caso da nossa função `tomarAgua()`, ela é do tipo `void`, pois não retorna nenhum valor.

Uma função sempre deve estar entre `{ }`. Indicando o início com `{` e o final com `}`.

FUNÇÃO SETUP E FUNÇÃO LOOP

Quando abrimos o Arduino IDE, um novo programa é iniciado, e nele já vem a estrutura básica para criarmos nosso sketch:



```
sketch_sep30a | Arduino 1.8.19
Arquivo Editar Sketch Ferramentas Ajuda
sketch_sep30a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
1 Arduino Uno em COM6
```

Nesse programa “base” já vem duas funções pré-definidas: a função loop e a função setup. Ambas são chamadas pelo próprio microcontrolador assim que ele é ligado.

FUNÇÃO SETUP

Como podemos ver pelo que está escrito no comentário, a função setup roda apenas uma vez, quando liga ou reinicia a placa. Ela é destinada para colocarmos aquelas informações que rodam apenas uma vez. Normalmente é onde definimos os parâmetros e as variáveis. Por exemplo no nosso sketch Blink:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
```

Nesse caso utilizamos o comando pinMode para configurar o LED incorporado como uma saída. Essa informação será rodada apenas uma vez, quando a placa for ligada ou reiniciada. Dessa forma, não é necessário repetir que o LED incorporado é uma saída ao longo do programa.

FUNÇÃO LOOP

A função loop, diferente da função setup, roda indefinidamente. Quando chega na última linha do código escrito dentro da função loop, ela volta ao início, e assim por diante. Aqui escrevemos o código principal que queremos rodar.

```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

No código do Blink são utilizados dois comandos: digitalWrite e delay.

No digitalWrite, nós escrevemos o estado que desejamos que a variável obtenha. Na primeira linha:

```
digitalWrite(LED_BUILTIN, HIGH);
```

Já na segunda linha:

```
delay(1000);
```

Delay cria uma pausa no algoritmo a partir do tempo determinado entre (). Esse tempo deve ser escrito em milissegundos, então temos 1000ms = 1s.

PONTO E VÍRGULA E CHAVES

O Arduino lê o código linha a linha, e por esse motivo quando escrevemos um comando, devemos especificar quando ele inicia e quando ele termina. Para isso serve o ponto e vírgula (;). Ele determina onde termina um comando e começa o próximo. Quando o ponto e vírgula não é utilizado corretamente um erro é acusado.



```
expected ';' before '}' token
exit status 1
expected ';' before '}' token
```

Para especificar onde inicia um bloco de instruções utilizamos chave aberta “{” e onde termina, utilizamos a chave fechada “}”.

Outras funções:

Nome da função	RAM
Função Se e senão	<pre>if (condição) { } else { }</pre>
Função se em cascata	<pre>if (condição) { } else if (condição 2) { }</pre>
Função caso	<pre>switch (expressão) { case expressão = x: Bloco 1; break; case expressão 2 = y: Bloco 2; break; default: Bloco 3; }</pre>
Função enquanto	<pre>while (condição) { bloco de funções }</pre>
Função faça enquanto	<pre>do { bloco de instruções } while (condição)</pre>
Função para	<pre>for (variável;condição;incremento) { }</pre>

Apêndice F

Módulo 5 - *Breadboard*



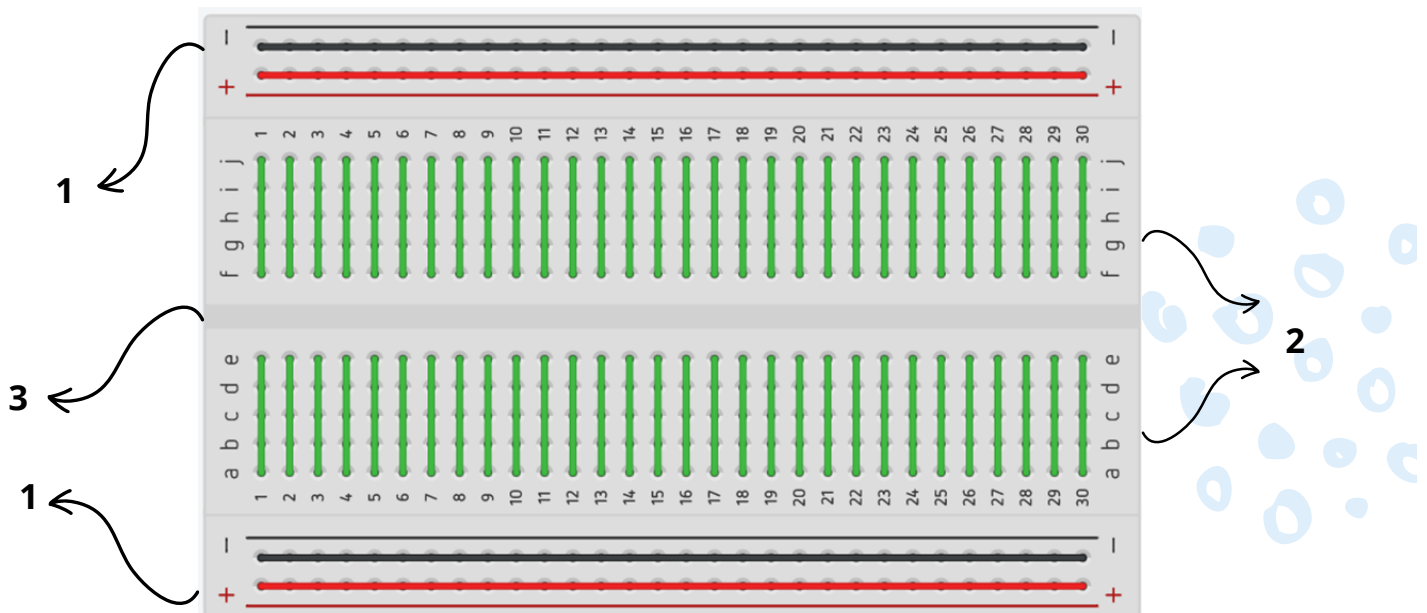
BREADBOARD (PROTOBOARD)

Continuando com o pensamento de prototipagem, vamos aprender sobre a matriz de contatos. Também chamada de Protoboard ou Breadboard.

É uma das ferramentas mais usadas quando estamos fazendo projetos temporários, pois é possível montar vários circuitos sem precisar soldar nenhum componente. Dessa forma, podemos testar rapidamente os circuitos de várias formas diferentes para ver como será seu comportamento.

Existem vários modelos de protoboard no mercado, e os que as diferencia é a quantidade de furos. Existem desde com 170 furos, até com 3220 furos. Porém esse número é ilimitado, já que é possível utilizar várias protoboards juntas.

Vamos ver a estrutura da protoboard (nesse caso de 420 furos):



Ela consiste em três partes principais:

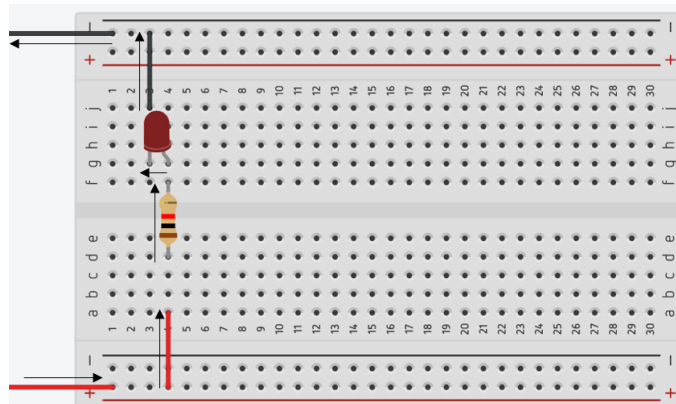
- 1) Área para distribuição de energia elétrica. É de boa conduta utilizar as linhas pretas para o GND e as linhas vermelhas para o positivo da fonte.
- 2) Área para montagem dos componentes (linhas verdes).
- 3) Área para montagem de circuitos integrados (parte central da protoboard).

A forma como as linhas estão representadas são justamente como estão interligadas.

As linhas de alimentação tem seus furos interligados na horizontal, e as colunas destinadas à montagem dos componentes estão ligadas na vertical.

Ou seja, como estão interligadas, podemos considerar cada furo na linha como se fosse o mesmo ponto. Dessa forma, quando um componente é inserido na linha, ele é conectado com todos que estão ligados nessa mesma linha.

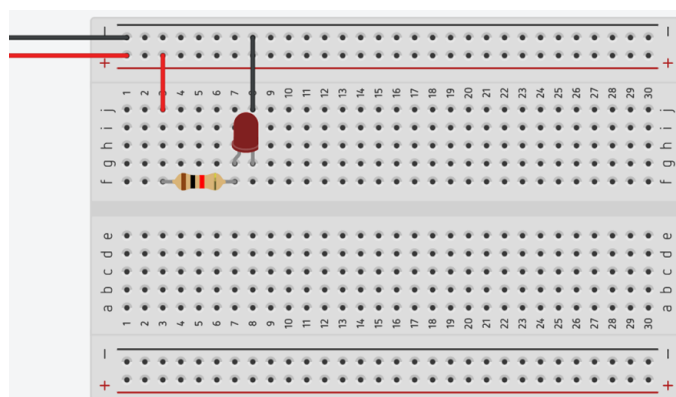
Vamos utilizar de exemplo nosso LED externo. Existem várias formas de conectar os componentes, um deles é assim:



Repare no sentido da corrente: ela sai da alimentação positiva, circula pela resistência, circula pelo LED e retorna ao negativo da fonte.

Perceba que a resistência funciona como uma “ponte” entre a parte de cima e a de baixo da protoboard (que não são interligadas), dessa forma a corrente passará pela resistência e irá até a coluna de cima que a resistência está ligada.

Outra forma é assim:



O resultado será o mesmo. Temos a resistência ligada ao terminal positivo do LED, e o GND no terminal negativo. Porém utilizamos um espaço muito menor da placa. Repare que agora a resistência faz como uma “ponte” entre duas colunas que não eram interligadas. Assim a corrente chega até a outra coluna através da resistência.

As possibilidades de montagem de circuito são muitas, basta ponderar qual a melhor forma para o seu projeto!

Apêndice G

Módulo 6 - Resistências elétricas



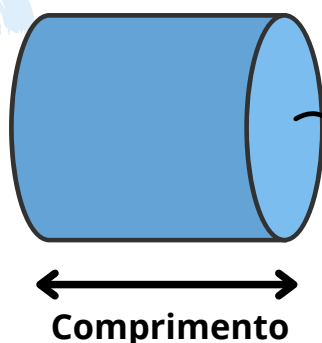
RESISTÊNCIAS ELÉTRICAS

Quando existe uma corrente elétrica em um condutor, os elétrons livres se deslocam, gerando colisão entre si e também nas paredes do condutor.

Essas colisões geram uma certa dificuldade para se deslocar, caracterizando a resistência elétrica.

Em resumo: A resistência elétrica é a dificuldade física que o material oferece à passagem de corrente elétrica, e segundo o Sistema Internacional (SI) é medida em ohms (Ω).

Cada condutor oferece uma resistência diferente à passagem de elétrons, que pode ser caracterizada por alguns fatores:



- comprimento do condutor (L) : quanto maior o condutor, maior a resistência.
- área da seção transversal (A): quanto menor a área, maior a resistência.
- resistividade do material (p): capacidade do material se opor a corrente elétrica. Quanto maior a resistividade, maior a resistência elétrica.

Área transversal

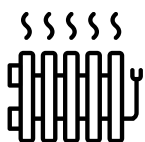
A Segunda Lei de Ohm resume esses fatores na seguinte equação:

$$R = p \frac{L}{A}$$

Vamos pensar em alguns exemplos de resistências elétricas no nosso dia a dia?



Secador de cabelo, aquecedor elétrico, ferro de passar roupa.



Percebeu que todos esquentam? Essa transformação de energia elétrica em energia térmica (calor) é chamada de Efeito Joule. Lembra que o deslocamento dos elétrons geram colisão com os átomos do condutor? O aumento da temperatura é gerado através do aumento da agitação dos átomos.



Esse aumento de temperatura deve ser levado em consideração quando formos escolher a resistência que utilizaremos nos nossos projetos.

Mas como saber qual valor de resistência deve ser usado? Pra isso iremos utilizar a Lei de Ohm:

$$\Delta U = R * i$$

Onde U é a diferença de potencial, R a resistência e i a corrente

Deve-se conhecer a diferença de potencial que temos em cima dos componentes para fazer o cálculo da resistência. Iremos utilizar o exemplo do LED vermelho do nosso primeiro projeto.

Para calcularmos a resistência utilizada, precisamos primeiro buscar no manual do LED qual sua corrente e tensão de funcionamento. Esse manual se chama Datasheet. Segue abaixo a tabela de onde retiramos essas informações:

Características Optoeletricas	Símbol.	Mín.	Tip.	Máx.	Unid.	Condição de Teste
Intensidade Luminosa	I_V		140		mcd	$I_F = 20\text{mA}$ (Nota 1)
Ângulo de Visada	2θ1/2		30		Grau	(Nota 2)
Comprim. de Onda	λ_d		640	660	nm	$I_F = 20\text{mA}$ (Nota 3)
Tensão Direta	V_F		2,0	2,5	V	$I_F = 20\text{mA}$
Corrente Reversa	I_R		---	50	μA	$V_R = 6\text{V}$

Fonte: Datasheet LED 5mm - VERMELHO DIFUSO
SYM-R503-30-D Symtronic

Para o LED vermelho, temos uma tensão direta de 2,0 V com uma corrente de 20mA.

Iremos utilizar esses dados para calcular.

Como o Arduino tem uma fonte de tensão de 5V, e o LED irá utilizar apenas 2V, teremos uma diferença de potencial (deltaU) de 3V. Ficando assim:

$$V = R * i$$
$$(5-2) = R * 0,02$$
$$R = 150 \text{ ohm}$$

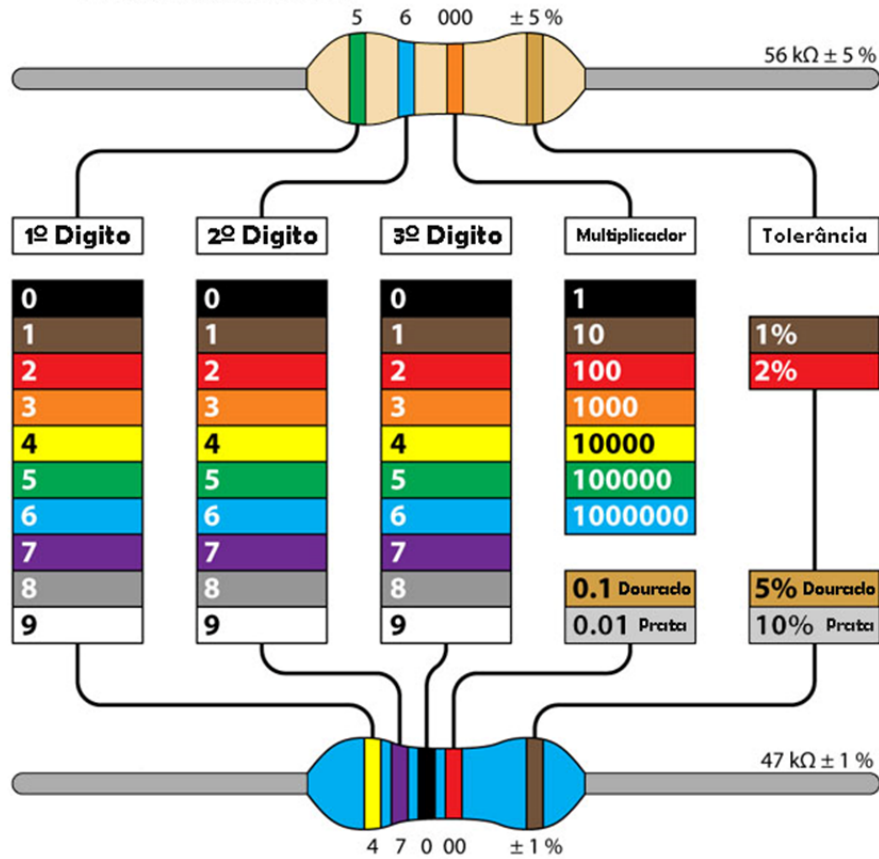
Ou seja, chegamos a conclusão de que precisamos de uma resistência de 150 ohms para fazer nosso LED funcionar. Agora, como encontramos qual LED é o de 150 ohms?

Temos duas formas mais simples:

Testar as resistências uma a uma até encontrar uma de 150 ohms (parece um pouco demorado certo?). Ou conhecer o código de cores das resistências:

Tabela de Cores dos Resistores

ComoFazerAsCoisas.com.br



Vamos ver alguns exemplos de resistências para implementarmos o código de cores?



Resistência de 4 faixas:

Primeira - Vermelho = 2

Segunda - Roxo = 7

Terceira - Amarela = 10000 (multiplicador)

Quarta - Dourado = 5% (tolerância)

270000 ohms
ou 270 k ohms



Resistência de 4 faixas:

Primeira - Marrom = 1

Segunda - Preto = 0

Terceira - Laranja = 1000 (multiplicador)

Quarta - Dourado = 5% (tolerância)

10000 ohms
ou 10 k ohms

E a nossa resistência de 150 ohms? Como seria?



150 ohms

Resistência de 4 faixas:

Primeira - Marrom = 1

Segunda - Verde = 5

Terceira - Marrom = 10 (multiplicador)

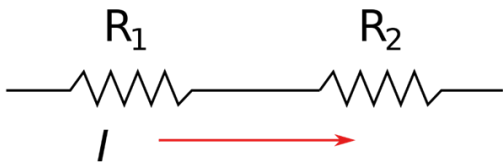
Quarta - Dourado = 5% (tolerância)

Porém pode ser que o cálculo da resistência não dê um valor comercial, ou seja, pode ser que precise arredondar para cima para encontrar um valor disponível no mercado.

Mas podemos também fazer associação de resistências:

RESISTÊNCIAS EM SÉRIE

Elas estarão ligadas em série quando uma estiver ligada em seguida da outra, sem nenhum outro componente no meio:



Percebe-se que a corrente que circula nas resistências é a mesma, e a diferença de potencial no sistema será a soma da diferença de potencial em cada uma.

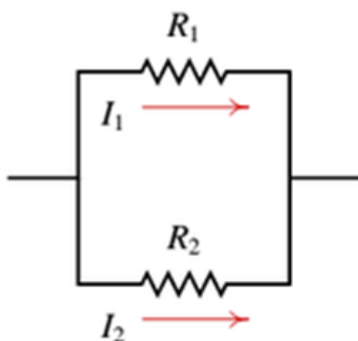
$$\Delta V = \Delta V_1 + \Delta V_2 = (R_1 + R_2) * i$$

$$R_{eq} = R_1 + R_2$$

Ou seja, quando colocamos duas resistências em série elas tem suas resistências somadas.

RESISTÊNCIAS EM PARALELO

Duas resistências estão em paralelo quando seus terminais estão ligados no mesmo ponto. Dessa forma:



Diferente da associação em série, agora a diferença de potencial é a mesma nas duas resistências, e a corrente no sistema é a soma das correntes nas resistências.

$$i = i_1 + i_2 = \left(\frac{1}{R_1} + \frac{1}{R_2} \right) * \Delta V$$

$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} \text{ ou } R_{eq} = (R_1 * R_2) / (R_1 + R_2)$$

Apêndice H

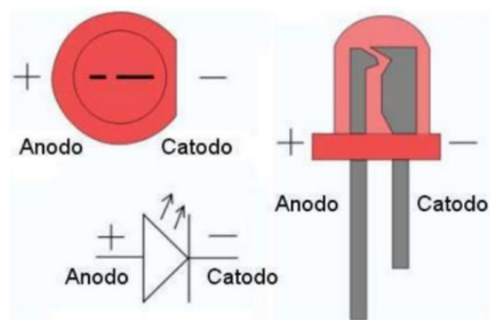
Módulo 7 - LED



O QUE É UM LED?

O LED (Light Emitting Diode – Diodo Emissor de Luz) é um componente eletrônico semicondutor bipolar conhecido como diodo emissor de luz, tem como função transformar a energia elétrica em energia luminosa. Como o LED é constituído por um diodo emissor de luz, é considerado um componente bipolar, e por esse motivo seus terminais têm polos definidos.

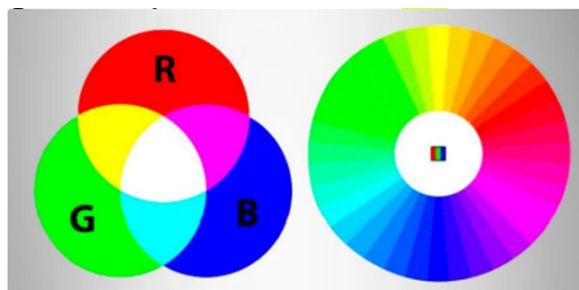
Anodo sendo o polo positivo e catodo o polo negativo. Desta forma o LED só funcionará caso seja polarizado diretamente, por isso o polo positivo deve ser conectado ao positivo da fonte e o polo negativo ao terra.



Fonte: Site Filipe Flop

LED RGB

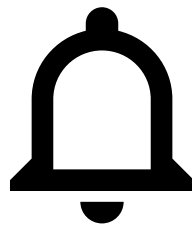
O LED RGB é um conjunto encapsulado de três LEDs de cores distintas, sendo um vermelho (red), um verde (green) e um azul (blue). O RGB se refere às três cores primárias presentes no encapsulamento. Elas podem ser associadas para formar cores de todo o espectro visível. Cada cor pode variar o fluxo de sua luminância de 0 a seu valor máximo, com intervalo definido de 0 a 255 (1 byte ou 8 bits). A variedade de cores se dá a partir do controle via software do fluxo de luminância de cada um dos 3 LEDs



Fonte: Site Baú da Eletrônica

Apêndice I

Projeto 2 - Campanha



Iniciante

Intermediário

Avançado

CAMPAINHA

A PARTIR DE UM BOTÃO ACIONAR UM LED, UM BUZZER E MOSTRAR NO MONITOR SERIAL QUANDO O BOTÃO FOR ATIVADO

Neste projeto podemos verificar e aprender alguns conceitos sobre a utilização do botão de pressão, o buzzer e o funcionamento do monitor serial.

Ferramentas indispensáveis para nossa jornada na aprendizagem da robótica!

Vamos lá?

COMPETÊNCIAS QUE VOCÊ VAI APRENDER COM ESSE PROJETO:

Hardware

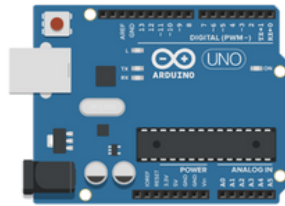
- Botão de pressão (*Push Button*)
- *Buzzer* (campainha)

Software

- Criação de novas funções (*if else*)
- Programação do botão
- Monitor Serial

Você vai precisar de:

- 1 placa Arduino UNO



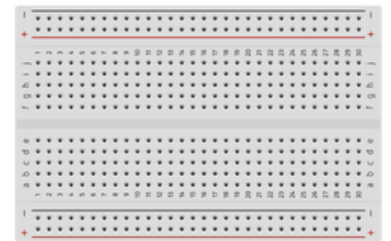
- Arduino IDE



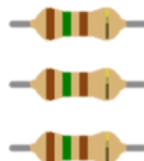
- 1 LED vermelho 5mm



- Protoboard (Breadboard)



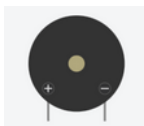
- 3 Resistências 150 ohms e 1/4 W



- Botão de pressão

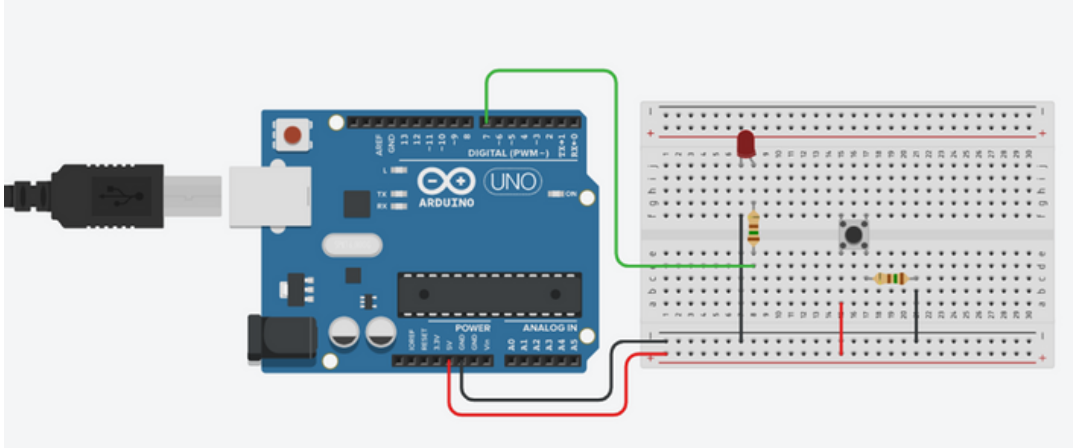


- *Buzzer*



PARTE 1 - Ligar um LED acionando um botão

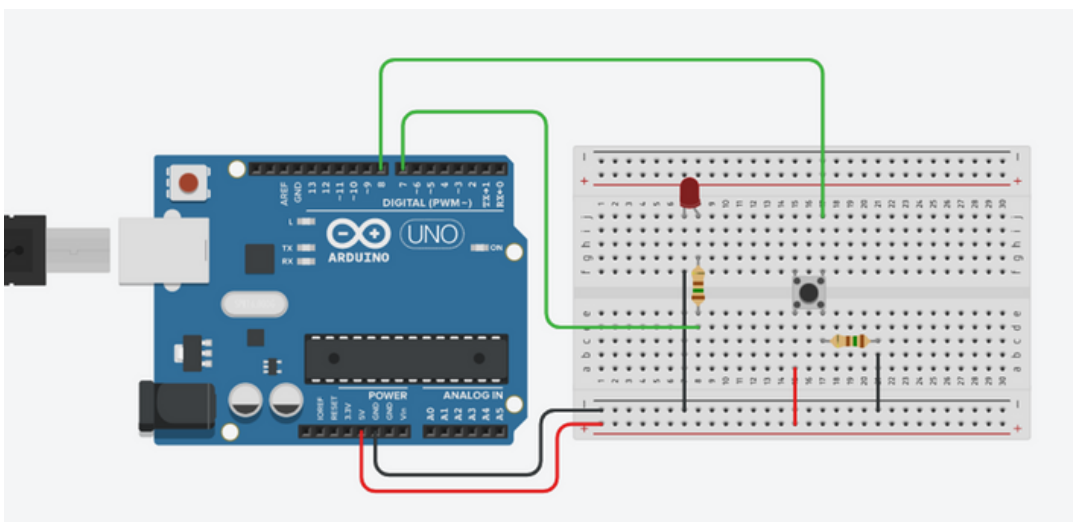
Aproveitando a estrutura que tínhamos no nosso antigo projeto, e utilizando os conhecimentos, iremos utilizar o LED da mesma forma, apenas agregando um botão de pressão. Vamos inicialmente simular no Tinkercad para vermos funcionar e poderemos pegar os códigos.

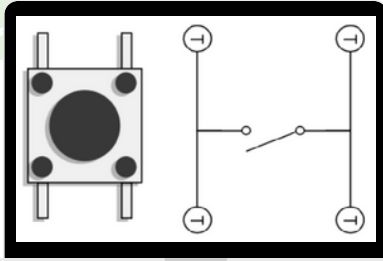


Note que agora colocamos o negativo (GND) e o positivo (5V) nos barramentos inferiores para poder ficar mais organizado, dessa forma podemos utilizar esses barramentos para ligar nossos componentes.

Para o botão funcionar é necessário alimentá-lo com 5V. Para isso devemos ligar um lado no 5V e outro no GND. A resistência é usada para não haver interferência, dessa forma a energia é equilibrada no botão, trazendo estabilidade.

Agora que o botão foi alimentado, precisamos conectar um pino que receba a informação de que o botão está pressionado. Para isso ligaremos ao pino 8.





? Você sabia?

O botão de pressão é um contato elétrico, quando ele não está apertado o contato está aberto, por isso é chamado de "contato normalmente aberto (NA)". O contato entre as placas é feito apertando o botão, possibilitando que o sinal seja enviado de um lado ao outro.

Temos como objetivo final fazer com que o LED ligue quando o botão for pressionado. Para isso precisamos criar um algoritmo que leia o estado da entrada digital onde o botão foi conectado, e a partir dessa informação tome uma decisão para seguir para os próximos passos. Vá na janela de código do Tinkercad e coloque o seguinte código:

SE a leitura do pino digital 8 (botão) for igual a ALTO (estiver pressionado), ENTÃO enviará sinal alto para o pino 7 (LED)

para sempre

se ler pino digital 8 = ALTO, então

definir pino 7 como ALTO

outro

definir pino 7 como BAIXO

Bloco se, então outro

Se a condição não for satisfeita, enviará o sinal baixo para o pino 7 (LED)

Vamos entender juntos a estrutura do código?

O bloco se, então outro (em laranja) é chamado na programação C++ de **if else**. Resumindo, SE uma certa condição for satisfeita ENTÃO o código que está dentro do bloco será executada SENÃO (outro), executará o outro código dentro do bloco SENÃO.

No nosso caso, se a leitura do pino digital 8 (o pino onde o botão está ligado) for ALTO (estiver acionado), então mandaremos sinal alto para o pino 7 (onde o LED está conectado) e o LED acenderá. Caso a leitura do pino digital 8 não for ALTO, será enviado o sinal baixo para o pino 7 e o LED ficará apagado.

Se quiser entender e ver como esse código é escrito em C++ basta ir na janela de código e selecionar a opção Blocos + texto ou texto.

PENSE SEMPRE ALÉM!

Como faremos para manter o LED aceso mesmo depois de soltar o botão? E fazê-lo apagar quando apertarmos o botão novamente?

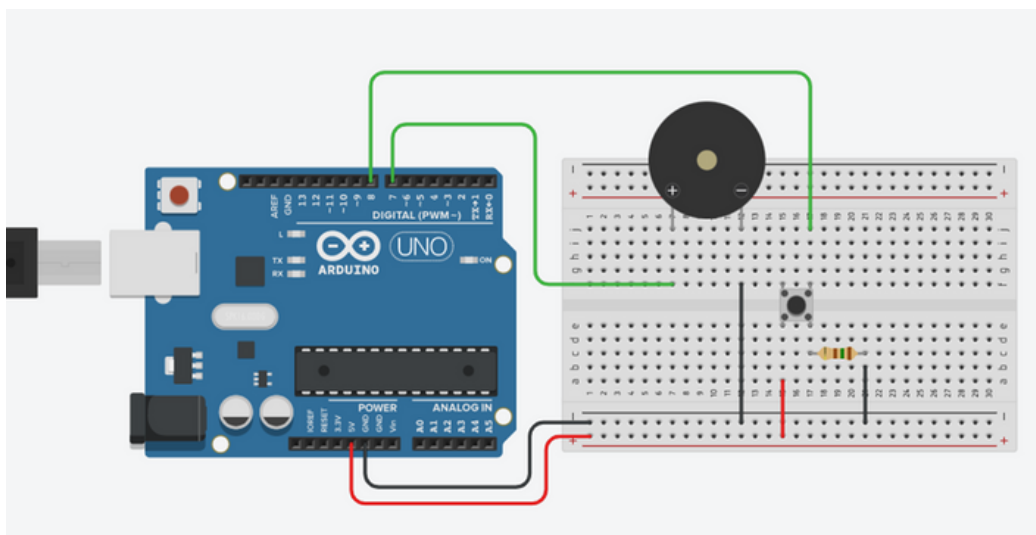
PARTE 2 - Incluir Buzzer

Agora que entendemos o funcionamento do botão, chegou a hora de fazer nosso buzzer funcionar! Ao invés de acender o LED, iremos acionar o buzzer.

Antes de mais nada, o buzzer é um dispositivo que é capaz de produzir frequências sonoras, e é muito utilizado em projetos do dia-a-dia, como o som do semáforo. Tem capacidade de produzir sons de frequências muito altas, chegando até 7kHz. Por isso são ideias para utilizar em sinalização, pois dificilmente não são escutados.

No nosso caso o buzzer emitirá um som assim que for alimentado, recebendo o sinal de comando (5V) e o GND (0V).

Podemos reutilizar a estrutura da simulação e apenas substituir o LED pelo buzzer. Como visto anteriormente, deve ser alimentado com 5V na entrada positiva e 0V (GND) na entrada negativa. Como queremos que ele seja um atuador, deverá ser acionado por um pino digital, no caso o mesmo pino 7 que estávamos utilizando para o LED.

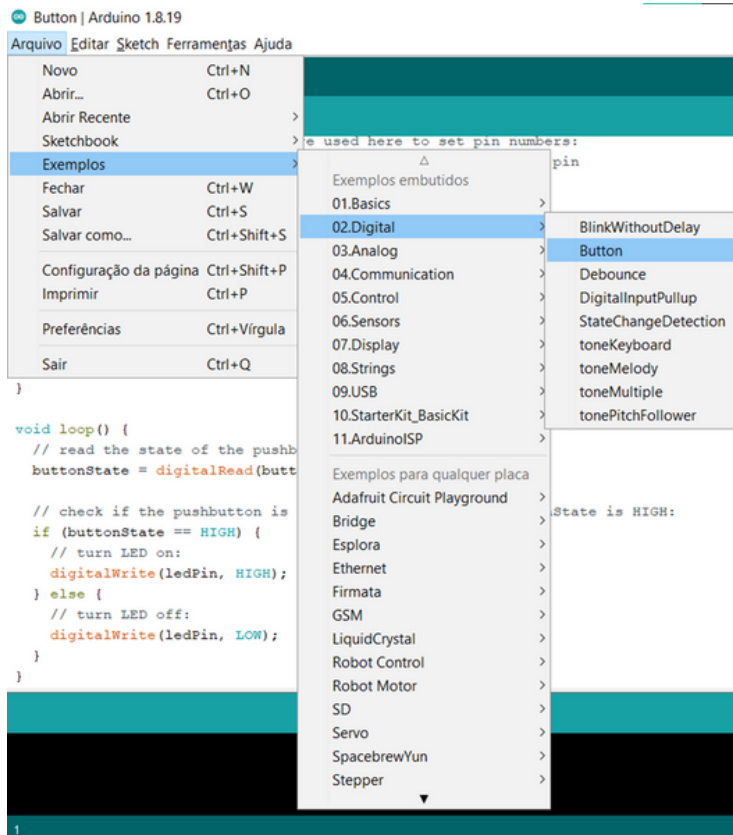


A programação continua a mesma. Podendo partir diretamente para a simulação!

PARTE 3 - Montagem na Breadboard

Agora que fizemos a simulação, podemos montar na nossa placa física! Para isso basta seguir a montagem que foi feita na simulação e conectar tudo nos lugares corretos. Vamos lá!

Com tudo montado podemos utilizar outro exemplo pronto disponível na IDE Arduino. Para isso abra o exemplo "Button" seguindo o caminho:



Você terá algo parecido com isso:

```
Arquivo Editar Sketch Ferramentas Ajuda
[Icons]
Button §
// constantes não mudam. (const). Aqui elas são usadas para configurar os pinos que são utilizados
const int buttonPin = 2; // pino do botão
const int ledPin = 13; // pino do LED ou buzzer

// Variáveis que irão mudar
int buttonState = 0; // variável que lê o estado do botão. Alto (1) ou baixo (0).

void setup() {
  // inicializa o LED como uma saída:
  pinMode(ledPin, OUTPUT);
  // inicializa o botão como uma entrada:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // lê o estado da porta digital onde o botão está ligado:
  buttonState = digitalRead(buttonPin);

  // verifica se o botão está pressionado. Se estiver, o estado do botão será HIGH:
  if (buttonState == HIGH) {
    // liga o LED:
    digitalWrite(ledPin, HIGH);
  } else {
    // desliga o LED:
    digitalWrite(ledPin, LOW);
  }
}
```

Vamos entender juntos a estrutura do código?

CONSTANTES

```
// constantes não mudam. (const). Aqui elas são usadas para configurar os pinos que são utilizados
const int buttonPin = 2;    // pino do botão
const int ledPin = 13;     // pino do LED ou buzzer
```

Antes da função setup foram definidas duas variáveis: buttonPin e ledPin. Como são variáveis que não irão mudar ao longo do código, pois são pinos do Arduino, podemos declará-las como const.

```
// Variáveis que irão mudar
int buttonState = 0;        // variável que lê o estado do botão. Alto (1) ou baixo (0).
```

Já o estado do botão (buttonState) irá variar entre HIGH (1) e LOW (0), então declaramos apenas como int, pois não será uma variável constante.

FUNÇÃO SETUP

```
void setup() {
    // inicializa o LED como uma saída:
    pinMode(ledPin, OUTPUT);
    // inicializa o botão como uma entrada:
    pinMode(buttonPin, INPUT);
}
```

Na função setup temos o comando pinMode, que como vimos no Projeto 1, é uma função que configura o pino. Nesse caso temos um pino que é uma saída (OUTPUT), para onde enviamos a informação para ligar e desligar nosso LED, e um pino que é uma entrada (INPUT), o qual recebe a informação do estado do botão. Todas essas informações devem ser fornecidas para o Arduino funcionar corretamente.

FUNÇÃO LOOP

```
void loop() {
    // lê o estado da porta digital onde o botão está ligado:
    buttonState = digitalRead(buttonPin);

    // verifica se o botão está pressionado. Se estiver, o estado do botão será HIGH:
    if (buttonState == HIGH) {
        // liga o LED:
        digitalWrite(ledPin, HIGH);
    } else {
        // desliga o LED:
        digitalWrite(ledPin, LOW);
    }
}
```

Na função loop, como visto antes, temos aquele código que será repetido indefinidamente. É nossa função “para sempre”.

Comando digitalRead:

Nela temos a linha onde lemos o estado do pino, e aí entra um comando que ainda não conhecíamos, o digitalRead.

```
// le o estado da porta digital onde o botao esta ligado:  
buttonState = digitalRead(buttonPin);
```

Como o objetivo é acender o LED a partir do botão, precisamos saber se o botão está apertado ou não, e é para isso que o comando digitalRead serve. Ele retorna o valor de estado da entrada digital utilizada como parâmetro. Nesse caso ele retorna o valor do estado da entrada digital que declaramos como buttonPin, no caso a entrada digital do pino 2.

Aqui nós armazenamos esse resultado na variável buttonState, por questões de organização, e para que quando a gente quiser utilizar a leitura do botão, não seja necessário escrever todo o comando por extenso novamente, apenas a variável que guardamos a leitura. Isso nos ajuda a manter o código mais “clean”.

Bloco if else:

```
// verifica se o botao esta pressionado. Se estiver, o estado do botão será HIGH:  
if (buttonState == HIGH) {  
    // liga o LED:  
    digitalWrite(ledPin, HIGH);  
} else {  
    // desliga o LED:  
    digitalWrite(ledPin, LOW);  
}  
}
```

Aqui nós forçamos nosso código a tomar uma decisão a partir da informação coletada do estado do botão.

O bloco if else (“se senão” em inglês) serve para SE a condição que está entre parênteses for satisfeita, ENTÃO o código entre as chaves será executado. SENÃO o outro código abaixo será executado. Da mesma forma que foi visto anteriormente na programação de blocos.

Agora que finalizamos o estudo do código, podemos modificá-lo para utilizar o Hardware que temos montado. Mudando os pinos que estão sendo utilizados e também os nomes das variáveis para que fique de melhor entendimento. Ficando assim:

```

// constantes não mudam. (const). Aqui elas são usadas para configurar os pinos que são utilizados
const int buttonPin = 8;    // pino do botão
const int buzzer = 7;      // pino do LED ou buzzer

// Variáveis que irão mudar
int buttonState = 0;       // variável que lê o estado do botão. Alto (1) ou baixo (0).

void setup() {
  // inicializa o LED como uma saída:
  pinMode(buzzer, OUTPUT);
  // inicializa o botão como uma entrada:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // lê o estado da porta digital onde o botão está ligado:
  buttonState = digitalRead(buttonPin);

  // verifica se o botão está pressionado. Se estiver, o estado do botão será HIGH:
  if (buttonState == HIGH) {
    // liga o LED:
    digitalWrite(buzzer, HIGH);
  } else {
    // desliga o LED:
    digitalWrite(buzzer, LOW);
  }
}

```



PENSE SEMPRE ALÉM!

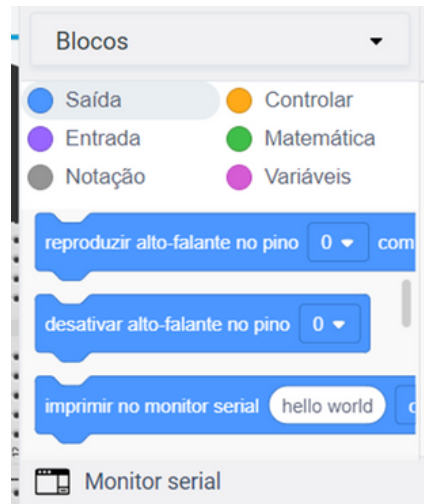
Agora além de acionar apenas o buzzer, como faríamos para acionar o LED e o buzzer ao mesmo tempo? Você pode testar na simulação para ver funcionando e depois montar no físico! Vamos lá!

PARTE 4 - Monitor Serial

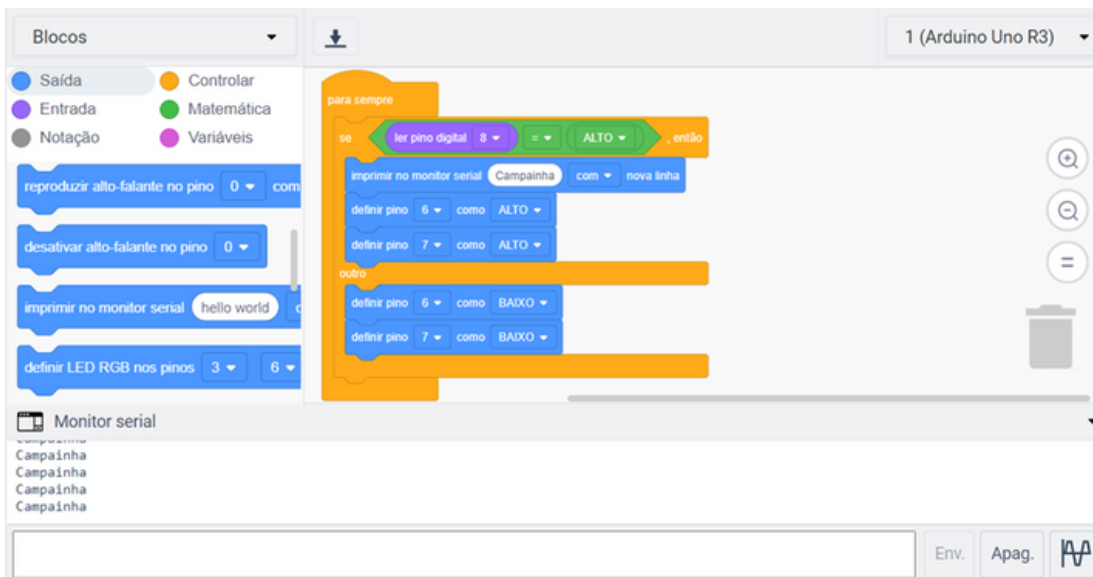
Agora iremos aprender como utilizamos uma ferramenta que já vem junto no Software do Arduino IDE, o monitor serial. Ele é a comunicação que temos entre o computador e o Arduino, já que conseguimos enviar informações para o Arduino por ele, e também receber informações e visualizá-las na tela do computador, sem precisar colocar outro periférico a parte.

No tinkercad conseguimos utilizar também essa ferramenta, onde podemos simular o uso do monitor e entender melhor seu funcionamento antes de partir para a placa física.

Para isso, na aba de código, nos comandos de saída, você encontra um bloco para imprimir informações no monitor serial. E embaixo tem um botão onde pode clicar para abrir o monitor.



Vamos então configurar o monitor serial para escrever “Campainha” toda vez que o botão for acionado?



Quando colocar para simular, mantenha essa janela aberta para poder verificar no Monitor Serial a informação sendo impressa!



PENSE SEMPRE ALÉM!

Como faríamos para imprimir uma outra informação enquanto o botão não tiver pressionado?

Monitor Serial no Arduino IDE:

E para fazer isso no nosso sistema físico? Como faríamos? Vamos aprender a inicializar o monitor serial no Arduino IDE e utilizá-lo.

Para isso podemos continuar usando nosso código anterior, do botão que aciona o LED e o buzzer. Mas agora vamos incluir ainda mais o monitor serial, sendo necessário incluir algumas configurações iniciais.

Vamos ver diretamente no código:

```
void setup() {
  Serial.begin(9600);
  // inicializa o LED e o buzzer como uma saída:
  pinMode(buzzer, OUTPUT);
  pinMode(led, OUTPUT);
  // inicializa o botao como uma entrada:
  pinMode(buttonPin, INPUT);
}
```

O comando `Serial.begin` inicializa a comunicação do computador com o Arduino e indica (entre parênteses) em qual taxa de transmissão vai ser feita essa comunicação. No caso 9600 é o máximo de bits por segundo que podem ser transferidos.

Após ser feito isso na função `setup`, basta colocarmos um comando no nosso código que permita imprimir a informação que desejamos.

```
void loop() {
  // le o estado da porta digital onde o botao esta ligado:
  buttonState = digitalRead(buttonPin);

  // verifica se o botao esta pressionado. Se estiver, o estado do botão será HIGH:
  if (buttonState == HIGH) {
    // liga o LED:
    Serial.println("Campainha");
    digitalWrite(buzzer, HIGH);
    digitalWrite(led, HIGH);
  } else {
    // desliga o LED:
    digitalWrite(buzzer, LOW);
    digitalWrite(led, LOW);
  }
}
```

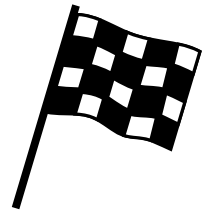
Quando a informação é um texto, devemos sempre colocar entre aspas. Agora basta compilar o código e enviar para a placa. Para abrir o monitor serial basta clicar no canto superior direito:





PARABÉNS!

Você construiu mais um degrau no conhecimento de robótica!



DESAFIO

Os buzzers não precisam necessariamente ser apenas um "alarme" sinalizando sons básicos, eles tem um alcance de 1 a 7kHz, logo são capazes de produzir melodias que utilizam diferentes frequências. Procure como fazer para que o buzzer reproduza melodias, e também se todos são capazes de fazer isso ou apenas alguns tipos específicos!



GOTAS DE CONHECIMENTO

Nesse projeto aprendemos vários conceitos, mas sempre podemos aprofundar nossos conhecimentos. Para isso foram feitos módulos, chamados de "Gotas de conhecimento" que irão explicar mais detalhadamente alguns desses conceitos:



Módulo 8 - Push Button e Buzzer



Módulo 9 - Monitor Serial



Use e abuse das gotas de conhecimento!
Sempre que tiver dúvida de algum conceito, não precisa voltar no projeto para revisar, basta ir direto na Gota que tem o que você precisa!

Apêndice J

Módulo 8 - *Push button e Buzzer*



PUSH BUTTON E BUZZER

PUSH BUTTON

Sempre que configuramos um pino da nossa placa como uma entrada, vemos uma característica em comum chamada alta impedância. Ou seja, para fazer que o estado mude de LOW para HIGH basta que uma pequena corrente passe pelo circuito. Essa característica nos permite ler diversos sensores, com o botão sendo um deles. Dessa forma pode-se observar a mudança de estado dos sensores e tirar conclusões para tomada de decisões descritas em código.

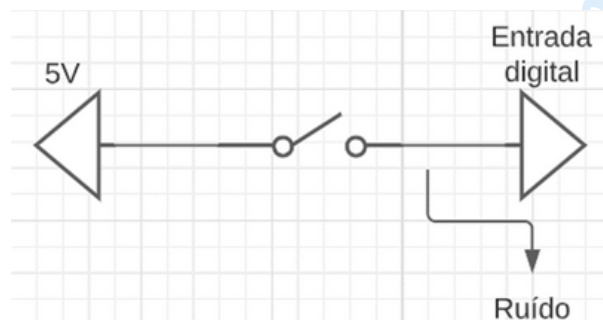
Como basta apenas uma pequena corrente passando pelo circuito para mudar o estado, o botão fica muito suscetível a ruídos do ambiente, já que podem causar essa pequena corrente e resultar em medições erradas.

Para resolver esse problema é necessário utilizar resistores de pull up ou pull down. Essas ligações permitem que esteja bem definida a tensão que está chegando na entrada digital, resolvendo o problema dos ruídos.



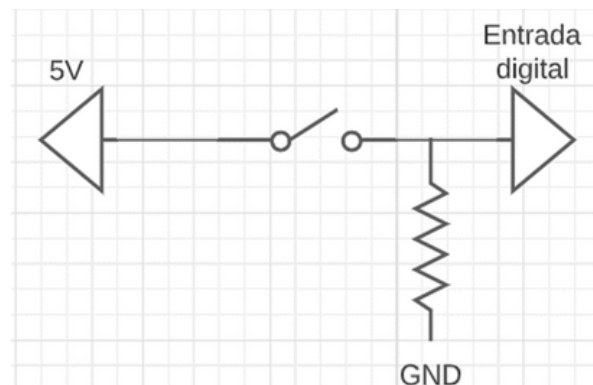
Entrada sem pull-up ou pull-down

A entrada digital que recebe o estado do pino estará sujeita a ruídos, já que a tensão não está bem definida (não se sabe se recebe 0V ou 5V)



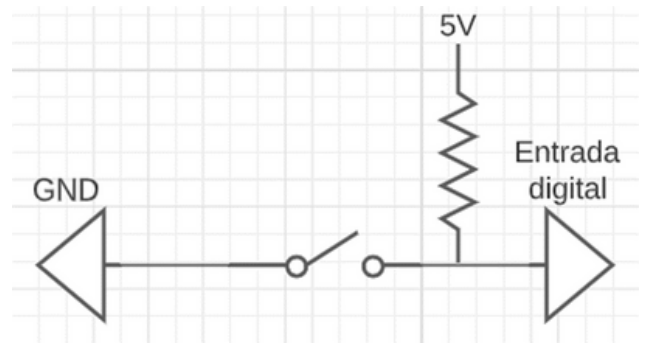
Entrada com resistor pull-down

A entrada digital que recebe o estado do pino estará sujeita a ruídos, já que a tensão não está bem definida (não se sabe se recebe 0V ou 5V)



Entrada com resistor pull-up

Com essa resistência definimos exatamente seu estado, então ele “enxergará” 5V quando não estiver apertado, logo com o contato aberto, seu estado será alto, já que a entrada digital estará recebendo 5V.



No caso do projeto 2 estamos utilizando uma resistência pull-down, o que significa que quando o botão está sem pressionar terá seu estado como LOW (baixo) e quando estiver pressionado será HIGH (alto).

BUZZER



O Buzzer, ou também comumente conhecido como campainha, é um dispositivo sonoro capaz de gerar frequências sonoras de 1Hz a 7kHz. Amplamente utilizado em experiências de robótica e automação como sinalizadores, já que alcança frequências mais altas gerando um som mais agudo.

Buzzer ativo:

O buzzer ativo é o mais comum do mercado, pois é o mais simples. Basta alimentá-lo adequadamente que o circuito interno já começará a gerar as vibrações no diafragma e produzir som, sem a necessidade de ter um circuito externo. Apesar do uso ser bem simples, a desvantagem é que possui uma frequência fixa, não possibilitando produzir sons diferentes.

Mas e se precisarmos de frequências distintas no projeto? Para isso existe o buzzer passivo.

Buzzer passivo:

Por ser mais complexo, este já precisa de um circuito externo para que possa produzir som. Dependendo de como esse circuito é montado podemos produzir sons diferentes, o que nos oferece ainda mais aplicações.

Dentre as categorias de buzzer ativo e passivo existem ainda os que se diferem pelo modo de funcionamento, dependendo de como são construídos internamente.

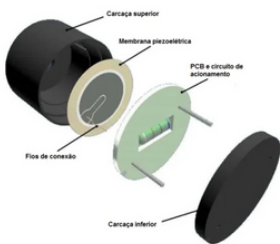
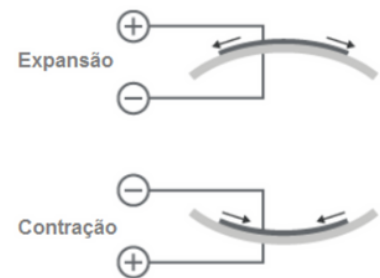
BUZZER PIEZOELÉTRICO

O efeito piezoelétrico é um efeito da física que descreve o comportamento de um dado cristal que é submetido a deformações e se resume em dois comportamentos:

- Quando o cristal é deformado (comprimido, por exemplo), é gerada uma tensão elétrica
- Quando o cristal é submetido a uma tensão elétrica ocorre uma deformação no cristal (efeito inverso do caso anterior)

Nos dispositivos elétricos acontece o último caso, o buzzer irá receber uma tensão elétrica, causando a deformação no cristal

O cristal está, geralmente, conectado a uma membrana metálica, e a deformação do cristal faz com que o ar em volta vibre, consequentemente vibrando a membrana, produzindo o som.

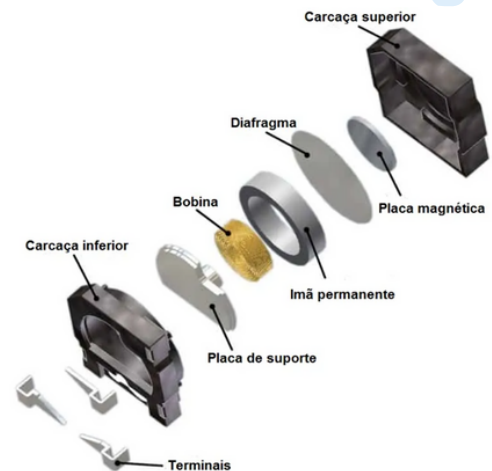


Como é um componente com poucas partes, é facilmente compactado, o tornando mais acessível.

BUZZER ELETROMAGNÉTICO

Esse buzzer já é mais complexo e com mais componentes. É o que é usado no fone de ouvido, por exemplo.

O funcionamento desse buzzer consiste na magnetização da bobina a partir de uma corrente. Como essa bobina está dentro de um ímã permanente anelar, teremos um ímã com campo eletromagnético variável. Esse efeito causa repulsão e atração desses ímãs, devido à variação do sinal enviado, o que faz com que a bobina se desloque. Como o diafragma está preso à bobina, essa movimentação faz com que o diafragma se movimente e o deslocamento de ar produz as ondas sonoras.



Apêndice K

Módulo 9 - Monitor serial



MONITOR SERIAL

O monitor serial é o canal de comunicação entre o Arduino e o computador, a partir dele podemos rodar testes e usá-lo para depuração.

A comunicação serial é o processo de mandar informações um bit de dados por vez, e essa comunicação pode ser feita via USB (Universal Serial Bus).

Para utilizar o Monitor Serial é necessário primeiramente criar um sketch e incluir algumas configurações. A configuração principal é a definição da taxa de transmissão de comunicação entre a placa e o computador. Para isso coloca-se o comando `Serial.begin(9600)`, onde o valor 9600 é a taxa de transmissão, que é o máximo de bits por segundo que pode ser transferido.

Vamos imprimir nosso primeiro “Hello World”, um teste que marca o início de muitas aprendizagens em linguagens de programação.

Para isso vamos utilizar o seguinte código:

```
1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6   Serial.println("Hello world!");
7   delay(1000);
8 }
```

Com esse código será impresso no monitor serial a frase “Hello World” a cada segundo, basta selecionar a placa que estamos utilizando e fazer o upload.

Ao fim do carregamento clique no botão Monitor Serial localizado no canto direito superior da IDE. Com isso irá iniciar o Monitor Serial e será possível visualizar nossa mensagem sendo enviada.



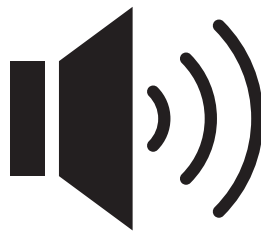
Você sabia?

A biblioteca Serial tem diversas funcionalidades, e pode ser visualizada nas documentações fornecidas pelo fabricante no site

<https://www.arduino.cc/reference/pt/language/functions/communication/serial/>

Apêndice L

Projeto 3 - Sensor de aproximação



- Iniciante
- Intermediário
- Avançado

SENSOR DE APROXIMAÇÃO

CONSTRUIR O CONHECIMENTO PARA FAZER UM SENSOR DE APROXIMAÇÃO DE UM OBJETO

Você já parou pra pensar qual tipo de sensor é utilizado no sensor de ré dos carros? É utilizado um sensor ultrassônico, que, assim como os morcegos, utilizam ondas sonoras para “enxergar” o que está a sua volta. Neste projeto iremos aprender os conceitos de utilização do sensor ultrassônico e implementar soluções muito úteis para construir nosso conhecimento!

Vamos lá?

COMPETÊNCIAS QUE VOCÊ VAI APRENDER COM ESSE PROJETO:

Hardware

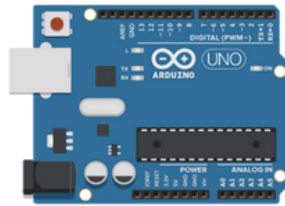
- Sensor de distância HC-SR04

Software

- Programação e configuração do sensor

Você vai precisar de:

- 1 placa Arduino UNO



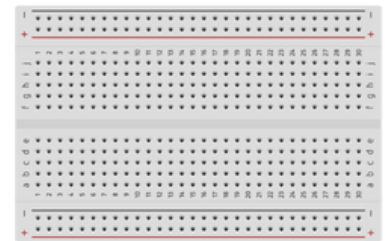
- Arduino IDE



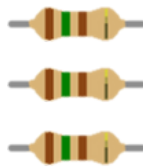
- 3 LEDs (1 vermelho, 1 amarelo e 1 verde)



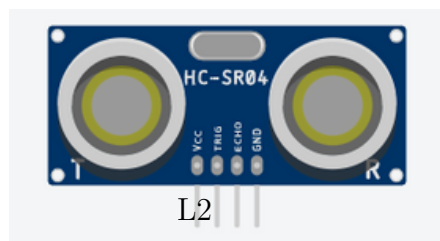
- Protoboard (Breadboard)



- 3 Resistências 150 ohms e ¼ W



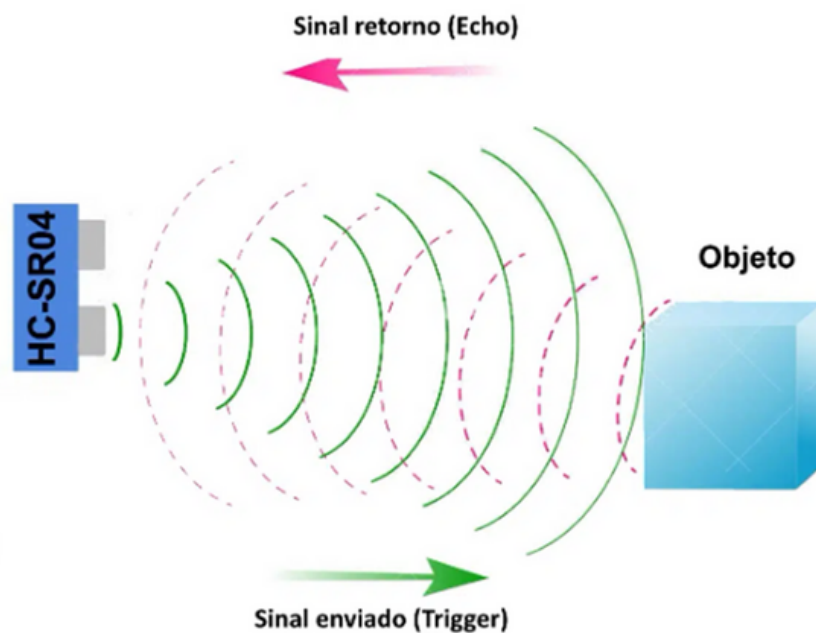
- Sensor de distância HC-SR04



SENSOR ULTRASSÔNICO

O sensor ultrassônico é amplamente utilizado em projetos de Arduino, seja para apenas medir a distância de um objeto ao sensor, até acionar portas digitais para tomadas de decisões, como desviar de um obstáculo, apitar uma buzina, etc.

O funcionamento se baseia em sinais ultrassônicos enviados pelo sensor (trigger), que fica aguardando o retorno do sinal (echo), e calcula a distância entre o sensor e o objeto através do tempo que levou para esse sinal retornar.



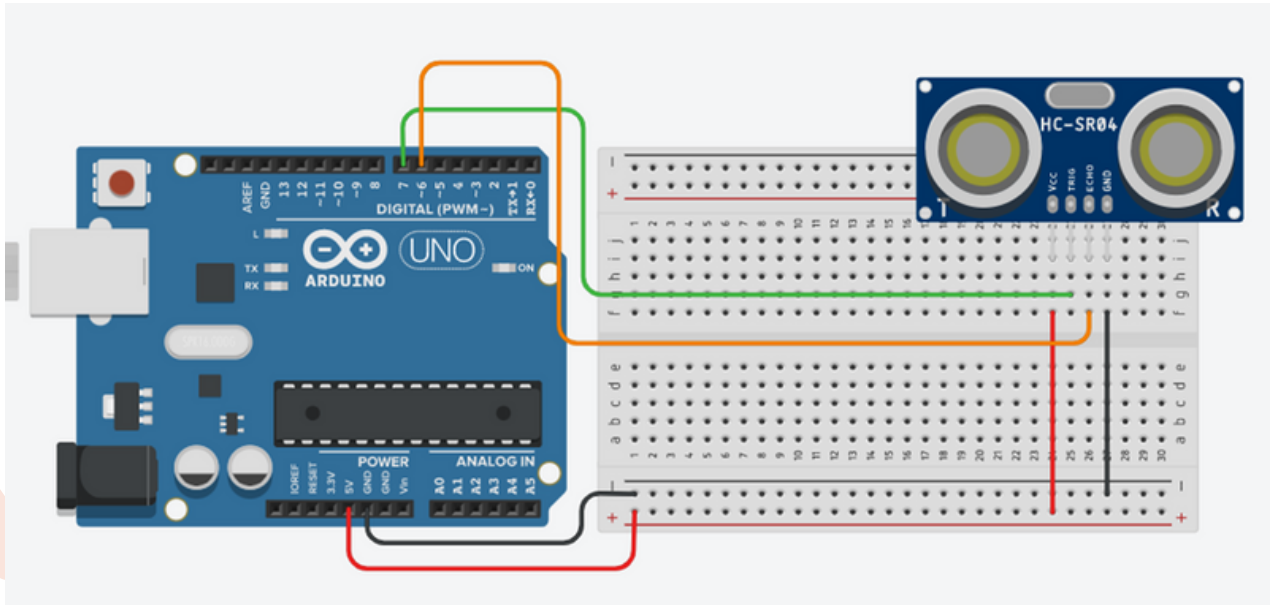
Iremos utilizar o módulo sensor HC-SR04, amplamente difundido na eletrônica e facilmente encontrado para venda em sites e em lojas de componentes eletrônicos.



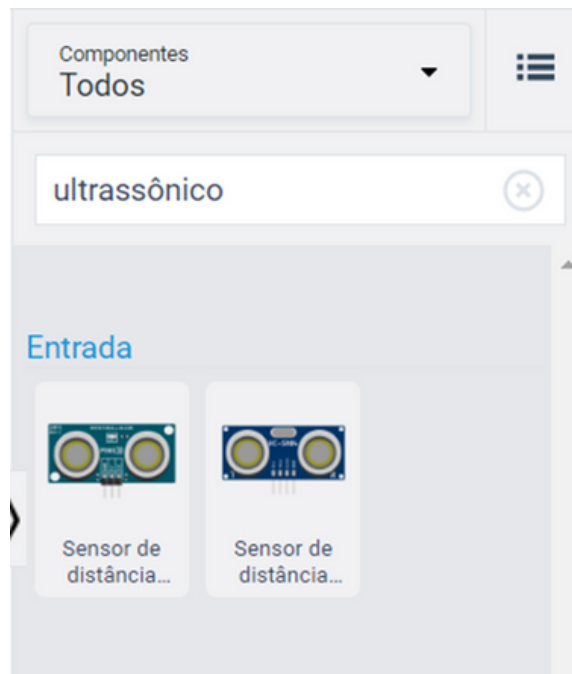
PARTE 1 - Simulação no Tinkercad

Mostrar distância no monitor serial

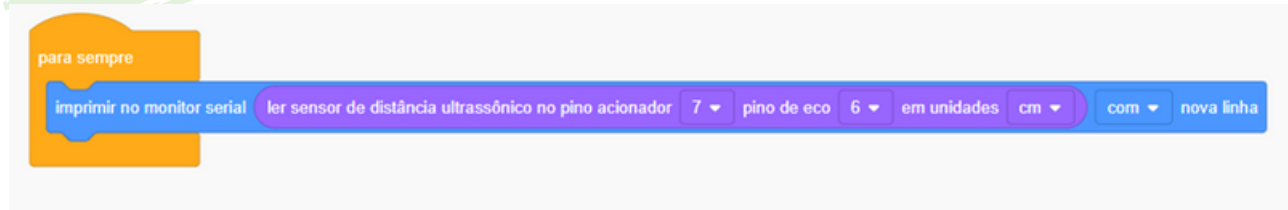
Para iniciarmos os estudos do sensor ultrassônico vamos começar fazendo a simulação no tinkercad, para isso vamos montar o seguinte esquema



O módulo HC-SR04 do sensor ultrassônico você encontra na aba Componentes - Todos:

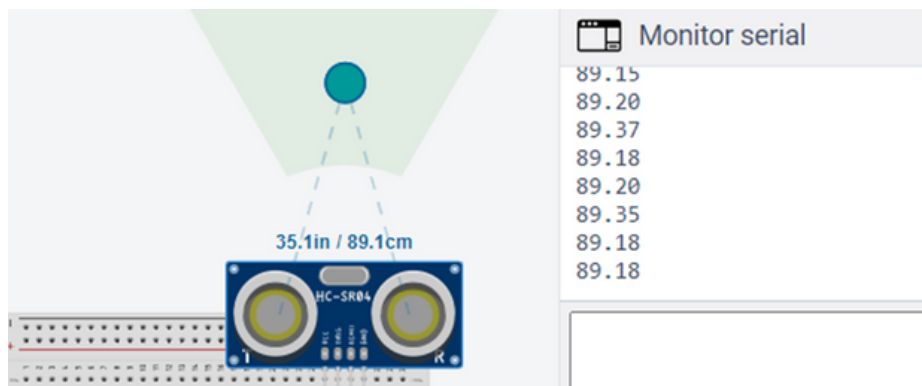


Na aba código pode-se encontrar no bloco de Entradas, o bloco para a leitura da distância pelo sensor de distância. Monte o seguinte código em blocos:



Aqui estamos imprimindo no Monitor Serial a leitura do sensor de distância, onde conectamos no pino 7 do Arduino o trigger (que aqui é chamado de acionador), e no pino 6 o echo (chamado de eco), e estaremos vendo a distância medida na unidade centímetros.

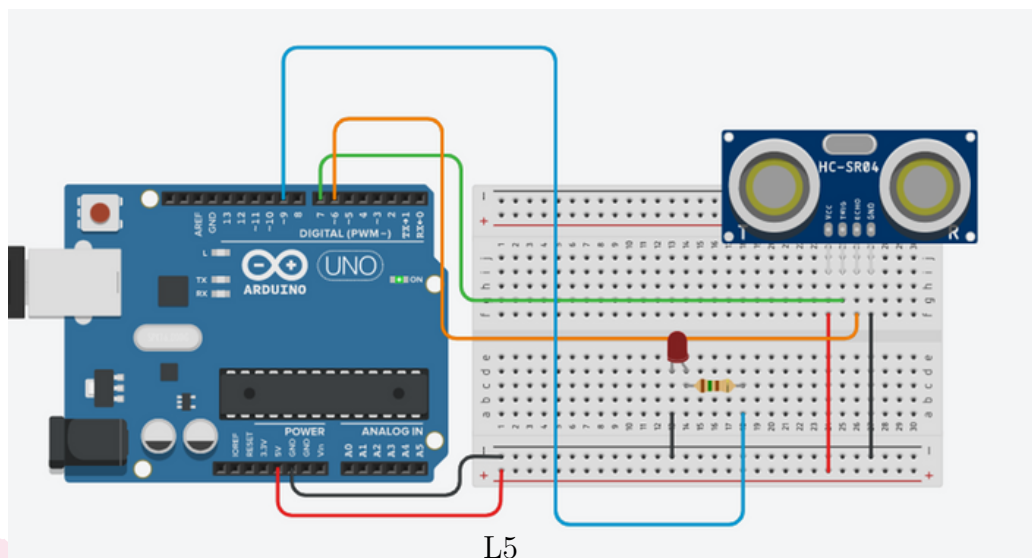
Desta forma, quando for colocado para iniciar a simulação, pode-se ver no Monitor Serial a medida da distância.



Acender led quando chegar a 30cm de distância

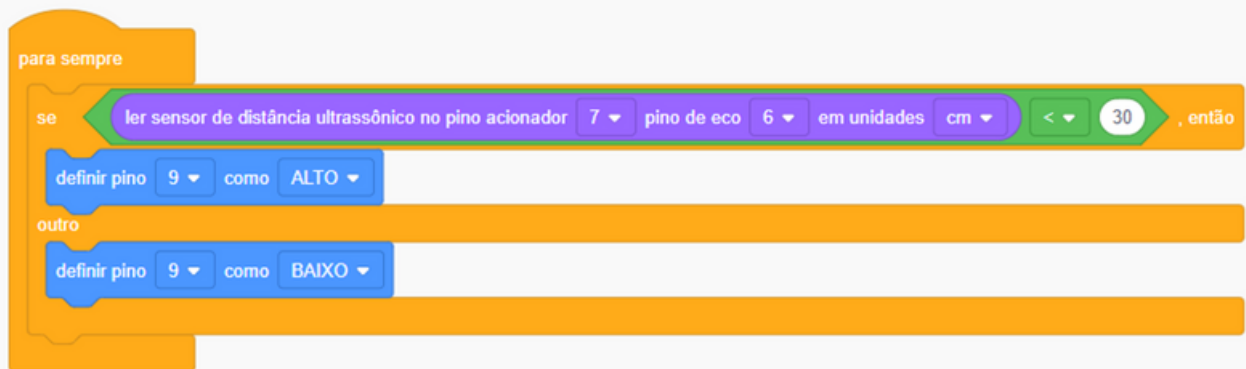
Agora vamos dificultar um pouco, colocando a leitura do sensor ultrassônico como uma condição para tomada de decisão! Suponha que este sensor está conectado na traseira de um carro que está dando ré e precisa acender uma luz quando o mesmo chega a 30 cm de uma parede. Vamos fazer?

Para isso precisamos primeiro incluir um LED no nosso esquema, para que possamos enviar o sinal para o mesmo acender.

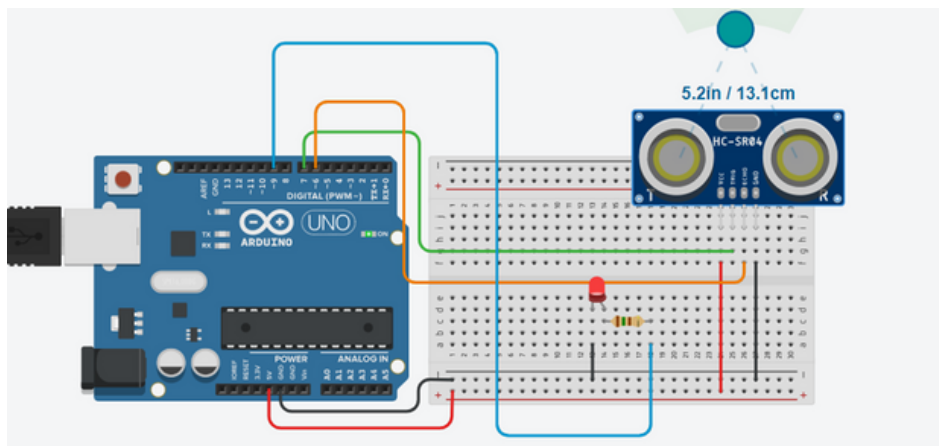


E agora no código?

Precisamos incluir uma condição para que o LED acenda quando o sensor ultrassônico tem uma leitura menor que 30cm e apague quando a leitura é maior que 30cm.



Vamos simular! Agora tudo já deve estar correto, e quando o sensor estiver lendo uma distância menor que 30cm o LED deve acender!

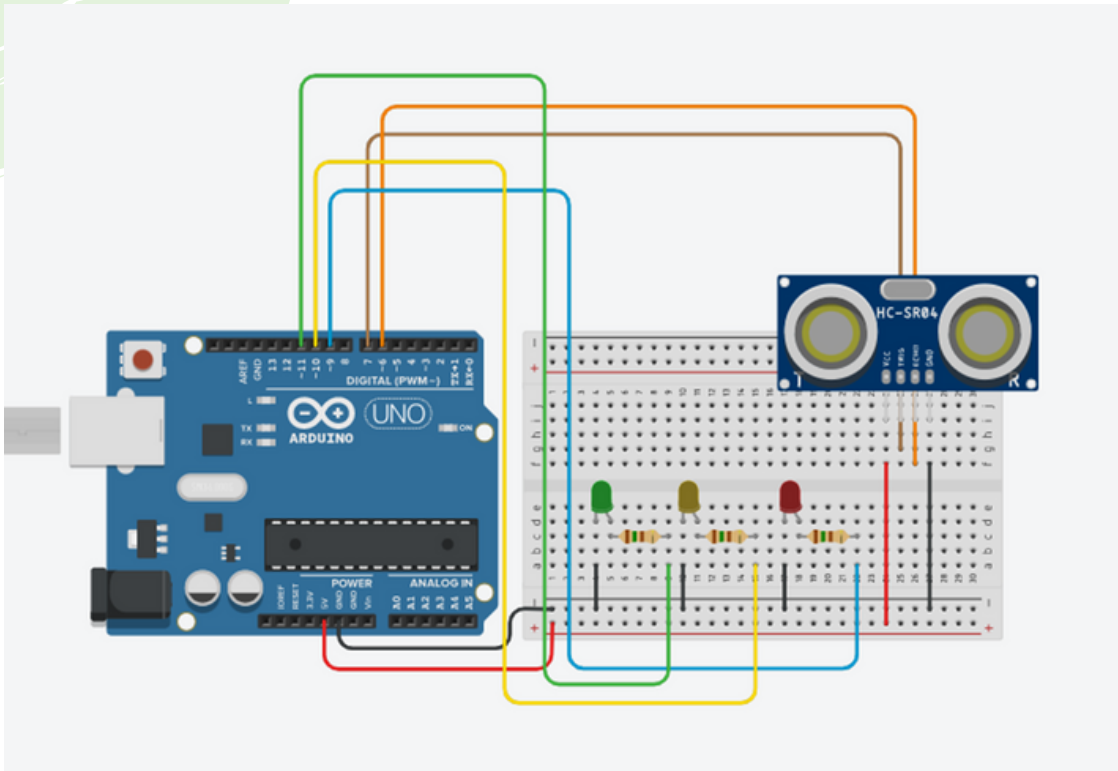


Agora vamos complicar um pouquinho mais e adicionar mais dois LEDs, um verde e um amarelo. Para podermos fazer um controle melhor da distância.

Sensor de aproximação com três LEDs

Nosso objetivo agora é fazer com que o LED verde acenda quando o objeto está longe (aproximadamente 80 cm), quando o objeto está se aproximando do sensor (aproximadamente entre 80cm e 50cm) acenda o LED amarelo, e por fim que o LED vermelho acenda quando o objeto estiver a menos de 50cm. Dessa forma podemos perceber o objeto se aproximando conforme os LEDs vão acendendo.

Vamos incluir o LED amarelo e o LED verde:

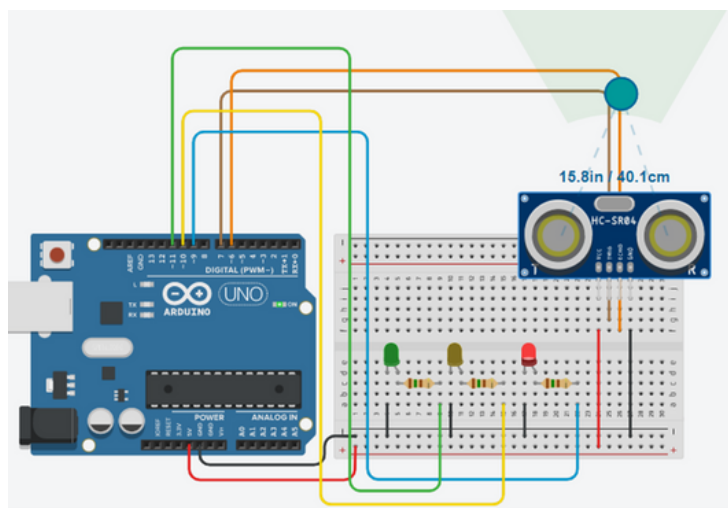
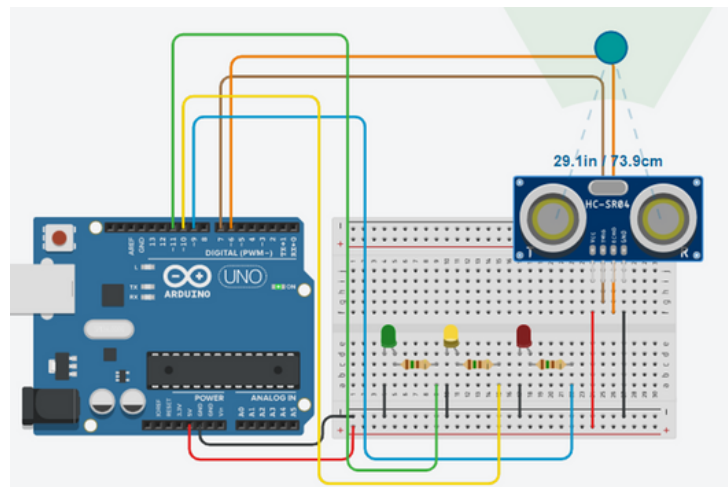
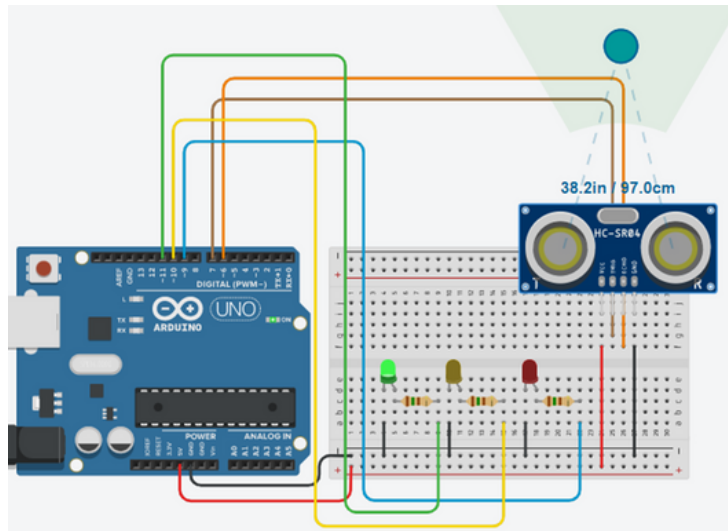


Para o código iremos utilizar dois blocos de if else (se outro na linguagem do tinkercad)

```
para sempre
  se ler sensor de distância ultrassônico no pino acionador 7 pino de eco 6 em unidades cm > 80 , então
    definir pino 9 como BAIXO
    definir pino 10 como BAIXO
    definir pino 11 como ALTO
  outro
  se ler sensor de distância ultrassônico no pino acionador 7 pino de eco 6 em unidades cm > 50 , então
    definir pino 10 como ALTO
    definir pino 9 como BAIXO
    definir pino 11 como BAIXO
  outro
  definir pino 11 como BAIXO
  definir pino 9 como ALTO
  definir pino 10 como BAIXO
```

Dessa forma primeiro verifica-se se a distância é maior que 80cm, se for acenderá o LED verde, se for menor iremos para o bloco "outro". Neste bloco verifica-se se a distância é maior que 50cm, e se for, estará entre 80cm e 50cm, acendendo o LED amarelo. Caso seja menor que 50cm iremos para o último bloco, que acende o LED verde.

Na simulação teremos algo parecido com isso:

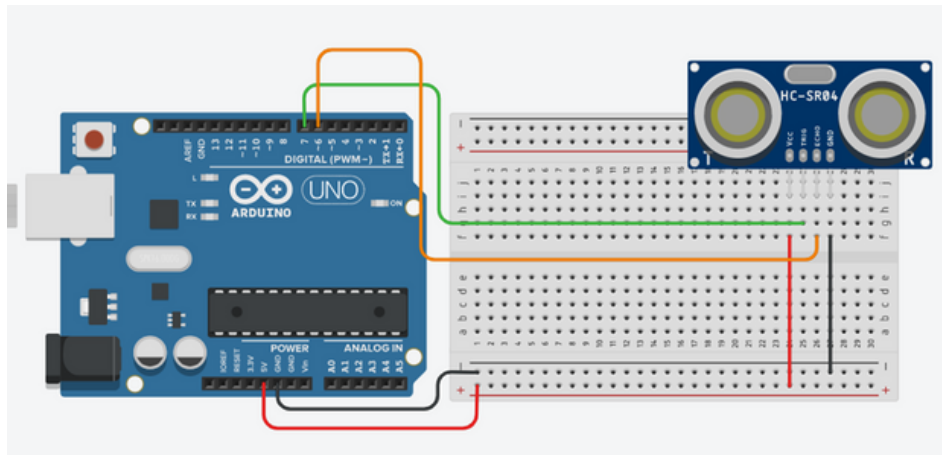


PARTE 2 - Sistema físico

Vamos agora montar tudo no nosso sistema físico?

Primeiro vamos apenas ler a distância do sensor ao objeto e imprimir no monitor serial!

Para isso basta montar o mesmo esquema que fizemos na primeira parte da simulação:



E agora basta colocar na Arduino IDE o código abaixo:

```
Sensor_ultrassonico
// C++ code
//
long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT); // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.println(0.01723 * readUltrasonicDistance(7, 6));
  delay(10); // Delay a little bit to improve simulation performance
}
```

Apenas uma explicação rápida sobre a função

```
long readUltrasonicDistance(int triggerPin, int echoPin)
```

Ela é responsável por enviar o sinal a partir do pino que conectamos o trigger, e receber o sinal no pino que conectamos o echo. Nesta função então estamos configurando os pinos do trigger e do echo, sendo o trigger uma saída e o echo uma entrada. Além disso estamos enviando o sinal do trigger, e recebendo o sinal a partir do pino do echo.

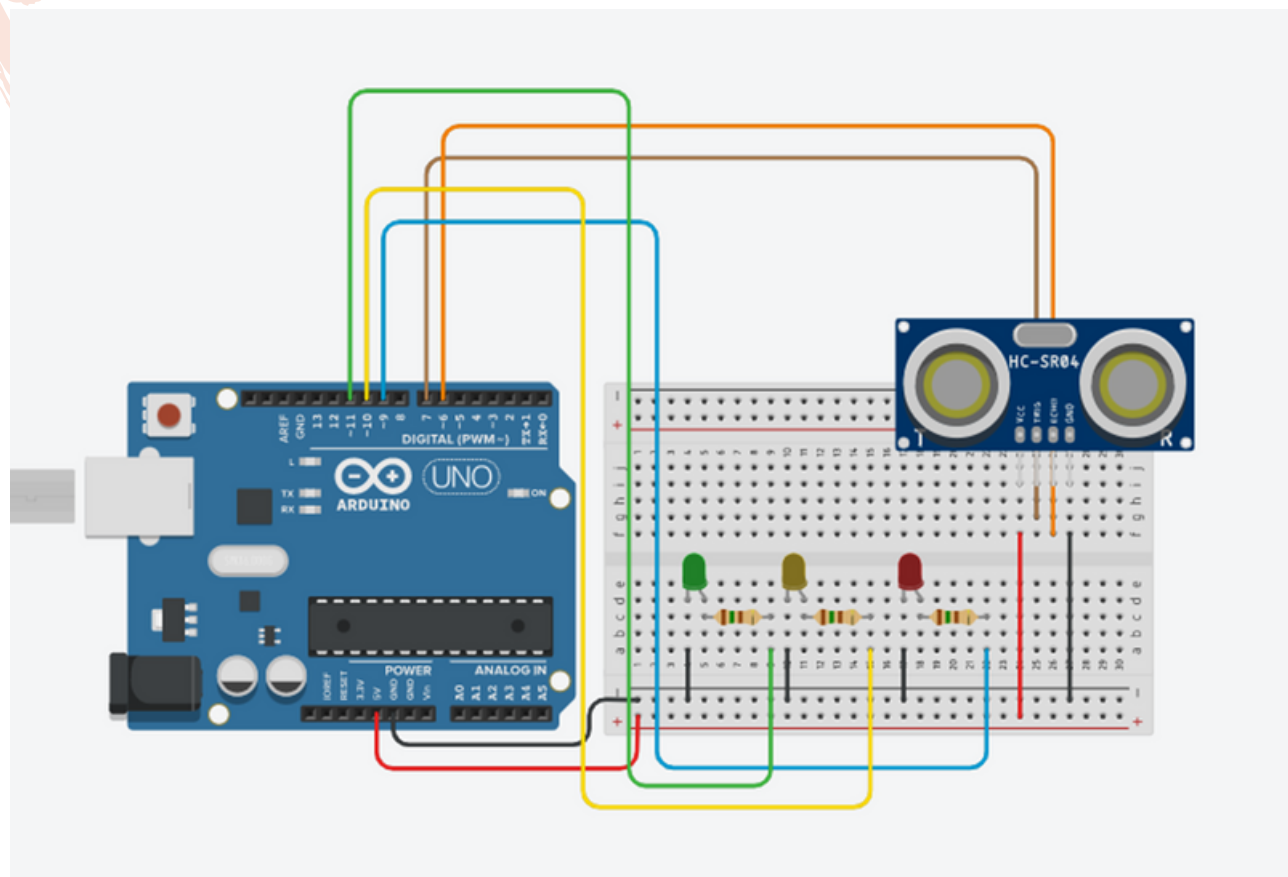
Por isso é interessante simular no Tinkercad antes de partir direto para o físico, pois a plataforma nos fornece a estrutura do código em C++ onde posteriormente podemos remodelar para o projeto que estamos fazendo.

Agora que já temos o código e o sistema físico montados, basta carregar o código e ver no Monitor Serial a distância do sensor ao objeto!

Sensor de aproximação com três LEDs no sistema físico

Por último, vamos montar o sistema mais complexo de hoje, o nosso sensor de aproximação com os LEDs verde, amarelo e vermelho!

Monte no sistema físico o seguinte esquema:



Podemos pegar o código que temos no Tinkercad para o último problema que solucionamos na simulação, colocar no Arduino IDE e carregar direto na placa! Para isso basta ir na aba “Blocos + texto”, copiar o código em C++ e colocar no Arduino IDE para enviar para a placa



```
Parte_3
// C++ code
//
long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT); // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}

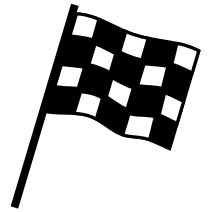
void setup()
{
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
}

void loop()
{
  if (0.01723 * readUltrasonicDistance(7, 6) > 80) {
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, HIGH);
  } else {
    if (0.01723 * readUltrasonicDistance(7, 6) > 50) {
      digitalWrite(10, HIGH);
      digitalWrite(9, LOW);
      digitalWrite(11, LOW);
    } else {
      digitalWrite(11, LOW);
      digitalWrite(9, HIGH);
      digitalWrite(10, LOW);
    }
  }
  delay(10); // Delay a little bit to improve simulation performance
}
```



PARABÉNS!

Você construiu mais um degrau no conhecimento de robótica!



DESAFIO

Para os próximos passos vamos pesquisar sobre o funcionamento do Display LCD, onde poderemos nos comunicar melhor com o usuário! Nele pode-se "imprimir" mensagens e até desenhos! No próximo projeto vamos vê-lo funcionando!



Apêndice M

Questionario qualitativo alunos

Qual ano você frequenta?

- 9o ano
- 12o ano

Como você acha que suas habilidades e conhecimentos em robótica mudaram desde o início das atividades? *

Long answer text

Os projetos atenderam às suas expectativas? Por favor explique *

Long answer text

Você achou as atividades desafiadoras o suficiente? Por que ou por que não? *

Long answer text

Os materiais das atividades (guião, módulos e Kahoot!) foram úteis e fáceis de entender? *

Long answer text

Qual o projeto que achou mais útil? Por quê? *

Long answer text

Você sentiu que as atividades ajudaram a desenvolver novas habilidades ou mesmo aprimorar as já existentes? *

Long answer text

Você achou benéfico o trabalho ser em grupo? Em caso afirmativo, como? *

Long answer text

Houve algum aspecto das atividades que você achou particularmente difícil ou frustrante? Caso sim, poderia explicar por favor? *

Long answer text

Você recomendaria essas atividades para outros colegas? Por que sim ou por que não? *

Long answer text

Houve algum aspecto negativo nas atividades? Se sim, como eles afetaram sua experiência de aprendizado? *

Long answer text

Apêndice N

Questionario quantitativo alunos

Muito ruim Ruim Neutro Bom Muito bom

Qual era seu nível de conhecimento prévio em robótica e programação?

Suas habilidades e conhecimentos em robótica mudaram desde o início das atividades?

Os projetos atenderam às suas expectativas?

Você achou as atividades desafiadoras o suficiente?

Os materiais das atividades (guião e módulos) foram úteis e fáceis de entender?

Você sentiu que as atividades ajudaram a desenvolver novas habilidades ou mesmo aprimorar as já existentes?

Você achou benéfico o trabalho ser em grupo?

O método de avaliação (Kahoot!) foi adequado?

Você achou o tempo dos projetos adequados?

Você recomendaria essas atividades para outros colegas?