

ESCOLA ESTADUAL DE TEMPO INTEGRAL 11 DE AGOSTO
14ªDIREC

**THE WORLD OF ARDUINO: Guia pedagógico para a utilização do Arduino em
ambientes educacionais, da teoria à prática.**

Área de Pesquisa: tecnologia educacional.

Orientador: José Everton Pinheiro Monteiro.

Autores: Arthur Matias Morais Silva, Pedro
Ryan Nunes de Oliveira.

Período de desenvolvimento do projeto: 4
meses.

UMARIZAL-RN

2024

AGRADECIMENTOS

Gostaríamos de expressar nossa sincera gratidão à Feira de Exposição Regional Acadêmica (FERA), realizada pela Universidade Federal Rural do Semi-Árido (UFERSA), campus Caraúbas-RN, pela oportunidade e apoio ao nosso projeto. A iniciativa FERA Nexus, em particular, desempenhou um papel fundamental na mentoria e aprimoramento do nosso relatório, proporcionando valiosas orientações que nos ajudaram a estruturar e fortalecer a seção de resultados de forma significativa.

Nosso agradecimento especial vai à equipe do FERA Nexus, por nos guiar ao longo desse processo e por oferecer um suporte contínuo e dedicado. Seu feedback foi essencial para o sucesso do nosso trabalho, e somos profundamente gratos por essa parceria. A todos os envolvidos, nosso muito obrigado!

RESUMO

O projeto "The World of Arduino" foi desenvolvido com o objetivo de preparar os alunos para enfrentar os desafios de um mercado de trabalho em constante evolução por meio da educação tecnológica. A iniciativa visa suprir a carência de recursos didáticos estruturados e a falta de conhecimento de muitos educadores sobre o Arduino, uma plataforma de prototipagem eletrônica de código aberto. O guia pedagógico proposto é direcionado aos docentes, abrangendo tanto aspectos teóricos quanto práticos, facilitando a compreensão e a aplicação dos conceitos de programação e robótica. Metodologicamente, o estudo foi estruturado como um projeto de intervenção educacional, aplicando um conjunto de aulas práticas e teóricas sobre Arduino, divididas em sete encontros. A pesquisa foi conduzida na Escola Estadual de Tempo Integral 11 de Agosto, em Umarizal/RN, com foco nos docentes do ensino médio. Os resultados mostraram uma evolução significativa no desempenho dos alunos, conforme evidenciado pela comparação entre avaliações diagnósticas e finais. As aulas proporcionaram uma melhoria notável nos conhecimentos sobre Arduino e prototipagem eletrônica, confirmando a eficácia do material didático desenvolvido. Entretanto, foram identificadas algumas limitações, como a necessidade de ajustar o ritmo das aulas para diferentes níveis de conhecimento prévio dos alunos e a disponibilidade de kits de Arduino nas instituições de ensino.

Palavras-chave: Arduino, Educação Tecnológica, Programação, Robótica, Guia Pedagógico.

ABSTRACT

The "The World of Arduino" project was developed with the aim of preparing students to face the challenges of an ever-evolving job market through technological education. The initiative aims to address the lack of structured teaching resources and the lack of knowledge among many educators about Arduino, an open-source electronic prototyping platform. The proposed pedagogical guide is directed at teachers, covering both theoretical and practical aspects, facilitating the understanding and application of programming and robotics concepts. Methodologically, the study was structured as an educational intervention project, applying a set of practical and theoretical Arduino lessons, divided into seven meetings. The research was conducted at Escola Estadual de Tempo Integral 11 de Agosto, in Umarizal/RN, focusing on high school teachers. The results showed a significant improvement in student performance, as evidenced by the comparison between diagnostic and final assessments. The lessons provided a remarkable improvement in knowledge about Arduino and electronic prototyping, confirming the effectiveness of the developed teaching material. However, some limitations were identified, such as the need to adjust the pace of lessons for different levels of prior knowledge among students and the availability of Arduino kits in educational institutions.

Keywords: Arduino, Technological Education, Programming, Robotics, Pedagogical Guide.

LISTA DE FIGURAS

Figura 1	–	Interface inicial do Google site.....	7
----------	---	---------------------------------------	---

LISTA DE GRÁFICOS

Gráfico 1	–	Distribuição das notas da prova diagnóstica	8
Gráfico 2	–	Distribuição das notas da prova final	9

SUMÁRIO

1. INTRODUÇÃO	1
2. OBJETIVO	2
3. MATERIAL E MÉTODOS	3
4. RESULTADOS E DISCUSSÃO	8
5. CONCLUSÕES	10
REFERÊNCIAS	12
APÊNDICES	14
ANEXO	55

1. INTRODUÇÃO

A educação utilizando a tecnologia tem se tornado cada vez mais essencial no mundo contemporâneo. Nosso projeto visa preparar os alunos para enfrentar os desafios de um mercado de trabalho em constante evolução. O Arduino é uma plataforma de prototipagem eletrônica de código aberto (Banzi, 2011), que se destaca como uma ferramenta poderosa e acessível para o ensino de conceitos de eletrônica e programação. No entanto, muitos educadores enfrentam dificuldades para integrar de forma eficaz o Arduino em suas aulas devido à falta de conhecimento sobre o assunto e à escassez de recursos didáticos estruturados.

O projeto "The World of Arduino" visa preencher essa lacuna desenvolvendo um guia pedagógico completo direcionado aos docentes para a utilização do Arduino em ambientes educacionais. Este guia abrange tanto aspectos teóricos quanto práticos do Arduino, facilitando a compreensão e a aplicação dos conceitos de programação e robótica oferecidos por essa plataforma. O guia consiste em sete aulas, sendo que a sétima é especialmente voltada para a iniciação científica, permitindo aos alunos aplicarem os conhecimentos adquiridos em projetos de pesquisa.

Dando a devida atenção ao método de ensino para que o guia seja aplicado da melhor forma possível, estamos utilizando a união da teoria e da prática.

Souza (2019, p. 42) relata que

Em meio às diversas metodologias de ensino que podem ser escolhidas para a mediação dos conteúdos curriculares inerentes a essa área do conhecimento, cita-se a experimentação a qual traz consigo a possibilidade de unir teoria e prática, permitindo que os alunos participem de forma ativa das aulas, motivando-os, tornando as abordagens teóricas mais atrativas, de modo a despertar o interesse pela ciência e o método pelo qual é construída.

As aulas foram projetadas para serem baixadas e utilizadas por qualquer docente interessado em ensinar utilizando a placa Arduino, sendo disponibilizadas em nosso site de fácil acesso, "theworldofarduino.online".

O projeto "The World of Arduino" contribui significativamente para a melhoria do ensino de tecnologia nas escolas, proporcionando aos educadores uma ferramenta valiosa para despertar o interesse dos alunos e promover uma aprendizagem ativa e interativa.

2. OBJETIVO

2.1 Objetivo geral:

Desenvolver um guia pedagógico gratuito e direcionado aos docentes para a efetiva utilização do Arduino em ambientes educacionais, visando capacitar os educadores a integrar de forma teórica e prática no ensino com o Arduino

2.2 Objetivos específicos:

- **Identificar os principais conceitos teóricos do Arduino:** Definir e descrever os principais componentes, funcionalidades e princípios de funcionamento da plataforma Arduino, incluindo a definição da placa Arduino, sua linguagem de programação e sua origem.
- **Desenvolver a estrutura do guia pedagógico:** Elaborar um plano detalhado de aulas e atividades práticas que integrarão teoria e prática utilizando o Arduino como ferramenta educacional.
- **Criar materiais de suporte para os docentes:** Produzir slides e planejamentos de aula que auxiliem os docentes na implementação das aulas, facilitando a compreensão e aplicação dos conceitos por parte dos alunos.
- **Testar e validar o guia pedagógico:** Realizar testes das aulas desenvolvidas em ambiente educacional real, coletando feedback dos docentes e alunos para ajustes e melhorias.
- **Disseminar o guia pedagógico:** Disponibilizar o guia completo e seus materiais associados em um site acessível para docentes interessados, sem nenhum custo ou mensalidade, promovendo sua utilização e coletando feedback contínuo.

3. MATERIAL E MÉTODOS

O objeto de estudo do presente projeto foi o uso da plataforma Arduino para o ensino de conceitos básicos de programação e robótica, apresentado em formato de um guia direcionado aos docentes. O guia integra teoria e prática como metodologia, sendo aplicado a alunos do ensino médio. A pesquisa foi realizada na Escola Estadual de Tempo Integral 11 de Agosto, localizada na cidade de Umarizal/RN. A sala de informática da escola, equipada com computadores e acesso à internet, foi o local onde todas as aulas práticas e teóricas conduzidas por nós foram realizadas.

O estudo foi estruturado como um projeto de intervenção educacional, com a implementação de um conjunto de aulas práticas e teóricas sobre Arduino, divididas em sete encontros. Cada aula tendo como objetivo abordar diferentes aspectos da programação e eletrônica utilizando o Arduino, sendo que o sétimo encontro foi totalmente direcionado à iniciação científica. A população-alvo para o presente projeto são os docentes, com o intuito de prepará-los para ensinar com o Arduino. Vieira, Meirelles, Rodrigues (2011, p. 2) definem que "é impossível pensar em um processo de ensino-aprendizagem que não integre os recursos tecnológicos e a prática educativa." A partir dessa premissa, e também observando a grade curricular do ensino médio e a falta de aulas utilizando essa metodologia, surgiu a ideia da confecção do guia.

Para a confecção de cada encontro, inicia-se sempre com o material de estudo destinado ao professor. Pesquisam-se informações e conteúdos através da leitura de revistas e artigos sobre o Arduino, que são posteriormente compilados em um único documento contendo todo o repertório necessário para que o professor conduza a aula. Em seguida, é elaborado o planejamento detalhado da aula, explicando como cada etapa deve ser conduzida. Por fim, desenvolvem-se os slides para o docente e disponibiliza-se todo o material no site destinado aos conteúdos.

Confecção dos encontros:

Para cada aula, o nosso guia possui um planejamento de aula, um material de estudo para o docente e um slide pronto para a aplicação da aula. Temos um modelo de planejamento de aula, um modelo do material de estudo do professor e um modelo dos slides, que aplicamos em todos os encontros. A maneira que escolhemos para que os professores interessados em nosso guia tivessem acesso a esse conteúdo foi disponibilizá-lo em um site, onde todas as aulas completas estão disponíveis para o docente.

Modelo de material de estudo do professor

De início, para a confecção do material de estudo, deve-se definir o assunto que será abordado em cada aula. Na primeira aula do guia, escolhemos fazer uma introdução ao Arduino, abordando sua definição, história e componentes básicos. Na segunda aula, resolvemos abordar a temática de circuitos do Arduino, onde falamos sobre o papel do Arduino em um circuito eletrônico, incluindo o conceito de circuito e a proposta de uma atividade prática que é a montagem de um semáforo com o Arduino.

Na terceira aula, abordamos a programação, explicando sua lógica e o funcionamento do código na prática, utilizando um LED. Na quarta aula, o tema escolhido foi sensores e entradas do Arduino, com uma introdução teórica aos principais sensores disponíveis para o Arduino, explicando o conceito de input e output, além de uma proposta de exemplo prático.

Na quinta aula, tratamos da comunicação serial, com explicação teórica, e na sexta aula, fornecemos um exemplo prático com o leitor de biometria e o software PLX-DAQ.

Parallax Data Acquisition tool é uma ferramenta de software add-in gratuita para a Microsoft Excel. Adquire até 26 canais de dados recolhidos por microcontroladores e organiza os números em colunas dentro de uma planilha em tempo real. com o Microsoft Excel (Silva; Melo; Camilo; Galindo; Viana, 2014, p. 3).

Na sétima aula, reunimos todo o conhecimento sobre iniciação científica, com o objetivo de encorajar os alunos a usarem todo o conhecimento adquirido ao longo do nosso guia para a confecção de um projeto científico.

Embora tenhamos separado as aulas em uma ordem numérica para fins de organização, cabe ao docente escolher por qual aula começar, qual aplicar e qual não aplicar, dando a liberdade para adaptar o guia à sua realidade e à realidade da instituição de ensino como um todo. O material de estudo do professor reúne todas as informações necessárias para ministrar a aula. Estamos utilizando o Microsoft Word para a confecção do material de estudo, mantendo cada documento organizado em arquivos separados.

Usando a primeira aula como exemplo, para a confecção do material de estudo da primeira aula, usamos como base teórica o livro “Arduino em Ação” dos autores Martin Evans, Joshua Noble e Jordan Hochenbaum, que aborda a história do Arduino.

O Arduino teve seu início no Interaction Design Institute na cidade de Ivrea, na Itália, em 2005. O professor Massimo Banzi procurava um meio barato de tornar mais fácil para os estudantes de design trabalhar com tecnologia. Ele discutiu seu problema com David Cuartielles, um

pesquisador visitante da Universidade de Malmö, na Suécia, que estava procurando uma solução semelhante, e o Arduino nasceu (Evans; Noble; Hochenbaum, 2013, p. 25).

Assim, abordamos o que é o Arduino, explicando quais são e como funcionam seus componentes básicos: protoboard, jumpers, LEDs, resistores, potenciômetro, botão e sensor de proximidade ultrassônico (Ricardo et al., 2024). Fornecemos um material de estudo completo para o professor, permitindo que o docente esteja bem preparado para aplicar a aula de maneira eficaz e capacitado para responder a possíveis dúvidas dos alunos.

Esse processo é realizado em todas as sete aulas do nosso guia. Consiste em buscar o conteúdo, ler o conteúdo e reunir esse conteúdo em um documento no Word, deixando o material pronto para o estudo do docente e reunindo todo o conhecimento necessário em apenas um local.

Modelo do planejamento de aula

O planejamento da aula “Consiste em orientar para a ação, sendo que o plano é uma estrutura de decisões quanto aos fins e meios, apresentando objetivos e metodologia.” (TAKAHASHI; FERNANDES, 2004, p. 2). Partindo dessa premissa, o plano de aula deve ser estruturado com base nos princípios de clareza, organização e objetividade, como necessário para um artigo científico.

Planejamento de Aula

Título da Aula: A partir da escolha do conteúdo que vai ser abordado é de extrema importância escolher um bom título para a aula, para engajar os alunos.

Introdução: Fazer uma boa introdução sempre será importante para o decorrer da aula. Sendo assim, inserimos em nossos planejamentos a orientação para que o professor inicie a aula explicando o que será feito e qual é o assunto da aula.

Duração: Todas as nossas aulas foram planejadas para ter uma duração de 1 hora e 40 minutos, divididas em duas aulas de 50 minutos. No entanto, o docente pode se adaptar à sua realidade.

Materiais Necessários: Informar no planejamento de aula os itens necessários para que o encontro aconteça.

Metodologia:

Exposição Teórica: Explicar todo o conteúdo e os conceitos fornecidos no material de estudo do professor por meio dos slides disponíveis no site.

Atividade Prática: Implementar um exercício onde os alunos coloquem seu conhecimento na prática, para despertar maior interesse neles. Assim, Bartzik; Daniele, et al. (2017) afirmam que as atividades práticas permitem que os alunos adquiram um conhecimento que a teoria por si só não abrange, sendo dever do professor e da escola oferecer essas atividades aos alunos. Nosso projeto visa fortalecer essa afirmação.

Avaliação: Implementamos uma avaliação diagnóstica na terceira aula do nosso guia, com o objetivo de avaliar o conhecimento antecipadamente dos alunos. Após a aplicação da terceira aula, realizamos uma prova final para quantificar a evolução dos alunos. Ambas as avaliações disponíveis estão no site do projeto.

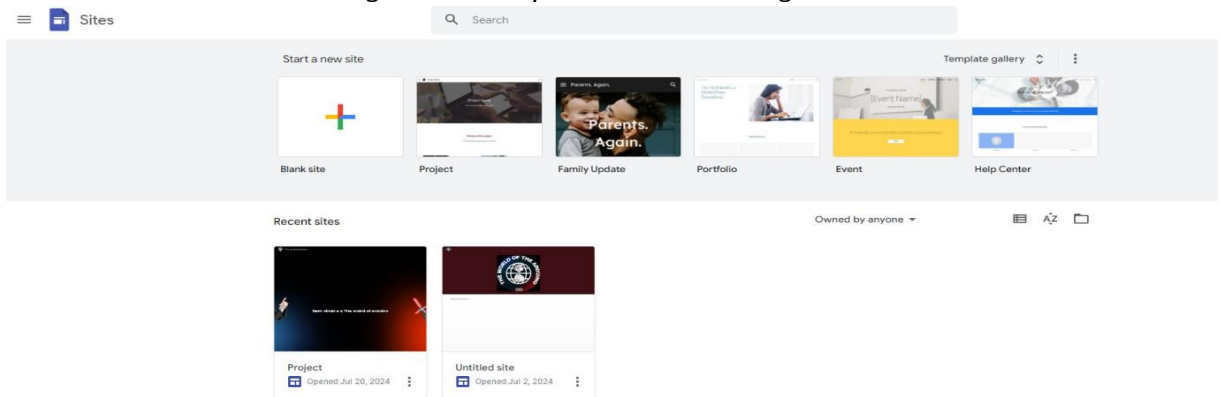
Referencias: Para credibilizar nosso projeto, incluímos todos os artigos, livros e sites que utilizamos para a preparação do material de estudo do professor e do planejamento de aula no final do planejamento de todos os encontros.

Confecção do site:

Para a confecção do site, iniciamos com uma busca aprofundada na internet à procura de ferramentas para a criação de um site. Procuramos também por sites de venda de domínios. “O Sistema de Nomes de Domínio (DNS) é um sistema hierárquico de nomenclatura distribuído que traduz entre endereços IP numéricos e nomes de domínio alfanuméricos usados para acessar sites.” (Kenneth, 2016, p. 89-106).

Fizemos a compra de um domínio, escolhendo o nome “theworldofarduino.online”. Após a compra do domínio, optamos por criar o site. Procuramos um criador de sites realmente confiável e seguro e escolhemos o criador de sites do Google, denominado Google sites. A ferramenta Google Sites é uma aplicação da Web 2.0 que possibilita a criação de sites na internet por qualquer usuário, mesmo aqueles sem grandes conhecimentos de programação. Ela oferece um ambiente simples para a criação e edição de páginas (Bottentuit Junior e Coutinho, 2016). Podemos observar a interface do google sites (Figura 1).

Figura 1: Interface inicial do Google site.



Fonte: Captura de tela da interface do Google Sites, tirada em 27/07/2024.

Para compartilhar as aulas, utilizamos o MEGA em uma página do Google Sites. O processo seguiu estas etapas:

- ☐ Carregar Arquivos no MEGA
- ☐ Crie uma conta no MEGA em mega.nz.
- ☐ Faça login e carregue seus arquivos clicando no ícone de "Upload".
- ☐ Obtenha o Link Compartilhável
- ☐ Encontre o arquivo carregado, clique com o botão direito e selecione "Obter link".
- ☐ Copiar ou gerar link.
- ☐ Adicione o link no Google Sites
- ☐ Acesse o Google Sites e edite a página desejada.
- ☐ Insira uma caixa de texto e transforme o texto em um link, colando o link do MEGA.
- ☐ Publicar e Testar

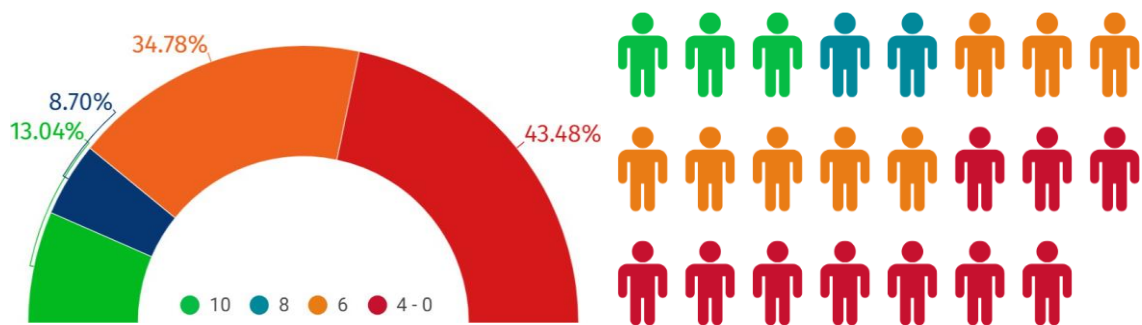
Publique a página e teste o link para garantir que funciona corretamente.

4. RESULTADOS E DISCUSSÃO

Desempenho na Prova Diagnóstica

Na aula número três do nosso guia pedagógico, denominada “Programação Básica”, aplicamos uma avaliação diagnóstica com 23 alunos da eletiva “Metodologia Científica”. Os alunos fizeram a prova diagnóstica sabendo de sua finalidade e propósito, que é identificar seu conhecimento prévio sobre o Arduino e prototipagem eletrônica. A distribuição das notas dessa avaliação pode ser visualizada no Gráfico 1.

Gráfico 1: Distribuição das notas da prova diagnóstica.



Fonte: FERA (2024).

A análise dos resultados mostrou que:

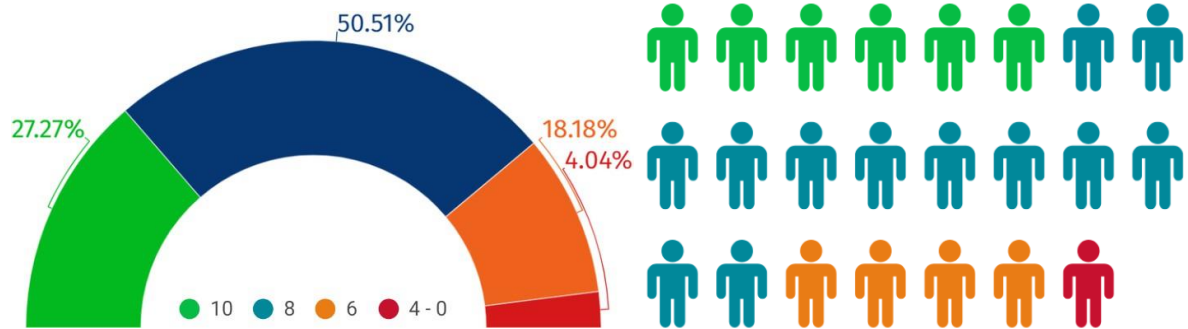
- 43,5% dos alunos obtiveram notas entre 0 e 4.
- 34,8% dos alunos alcançaram nota 6.
- 8,7% dos alunos obtiveram nota 8.
- 13% dos alunos atingiram a nota máxima de 10.

Esses dados indicam que a maioria dos alunos tinha um conhecimento inicial limitado sobre o tema, com quase metade da turma obtendo notas abaixo de 6.

Desempenho na Prova Final

Após a aplicação da terceira aula do nosso guia, foi realizada uma prova final para quantificar a evolução do conhecimento dos alunos. A distribuição das notas dessa prova está representada no Gráfico 2.

Gráfico 2: Distribuição das notas da prova final.



Fonte: FERA (2024).

Os resultados da prova final revelam uma melhoria significativa no desempenho dos alunos:

- Apenas 4,04% dos alunos obtiveram notas entre 0 e 5.
- 18,2% dos alunos alcançaram nota 6.
- 50,5% dos alunos obtiveram nota 8.
- 27,3% dos alunos atingiram a nota máxima de 10.

Comparando as duas avaliações, observa-se um aumento considerável no número de alunos com notas altas. A porcentagem de alunos com notas entre 0 e 5 caiu drasticamente de 43,5% para 4,5%, enquanto a porcentagem de alunos com notas 8 e 10 aumentou de 21,7% para 77,3%.

Discussão dos Resultados

Os resultados demonstraram que a aplicação do guia contribuiu significativamente para o aprendizado dos alunos, conforme evidenciado pela comparação entre as avaliações diagnósticas e finais.

Os alunos apresentaram uma evolução notável em seus conhecimentos sobre Arduino e prototipagem eletrônica, confirmando a eficácia do material didático desenvolvido e da metodologia aplicada. No entanto, algumas limitações foram identificadas, como a necessidade de ajustes no ritmo das aulas para atender diferentes níveis de conhecimento prévio dos alunos. Por isso, deixamos nas mãos dos docentes a escolha da ordem das aulas. Além disso, a disponibilidade de kits de Arduino nas instituições de ensino ainda é um desafio que pode limitar a abrangência do projeto.

Futuras pesquisas poderiam explorar a aplicação do guia em diferentes contextos educacionais, bem como a inclusão de módulos adicionais que abordem tópicos mais

avançados em eletrônica e programação. Outra linha de investigação interessante seria a adaptação do material para diferentes faixas etárias e níveis de ensino, além de iniciativas de aquisição de kits de Arduino para as escolas, incentivando gestores dessas instituições a adquirir o Arduino e os kit.

5. CONCLUSÕES

Este projeto de iniciação científica teve como objetivo desenvolver e aplicar um guia didático para o ensino de Arduino, visando facilitar a compreensão e utilização dessa plataforma de prototipagem eletrônica por docentes em ambientes educacionais. Ao longo do projeto, foi possível elaborar um conjunto de aulas estruturadas que abordam desde os conceitos básicos sobre o Arduino até a programação na prática. Os resultados demonstraram que a aplicação do guia contribuiu significativamente para o aprendizado dos alunos, conforme evidenciado pela comparação entre as avaliações diagnósticas e finais. Egea, Rosa e Rodríguez et al. (2017) indicam que o aprendizado dos alunos é significativamente melhorado em aulas de laboratório de eletrônica com o uso do Arduino, mostrando uma melhoria de 30% em relação aos métodos de ensino convencionais. Os alunos apresentaram uma evolução notável em seus conhecimentos sobre Arduino e prototipagem eletrônica, confirmando a eficácia do material didático desenvolvido e da metodologia aplicada. No entanto, algumas limitações foram identificadas. Foi observada a necessidade de ajustes no ritmo das aulas para atender aos diferentes níveis de conhecimento prévio dos alunos. Para contornar essa questão, deixamos aos docentes a decisão sobre a ordem das aulas a ser seguida. Além disso, a disponibilidade de kits de Arduino nas instituições de ensino continua a ser um desafio, o que pode limitar a abrangência do projeto.

Futuras investigações poderiam explorar a aplicação do guia em diferentes contextos educacionais, analisando a eficácia em variações de ambientes e perfis de alunos. Além disso, a inclusão de módulos adicionais que abordem tópicos mais avançados em eletrônica e programação pode proporcionar uma profundidade maior ao aprendizado e expandir o alcance do guia. Outra linha de pesquisa interessante seria a adaptação do material para diferentes faixas etárias e níveis de ensino. Adicionalmente, uma iniciativa para a aquisição de kits de Arduino para escolas seria benéfica. Motivar gestores educacionais a investir na compra desses kits pode ampliar o acesso à tecnologia e permitir a implementação do guia em um maior número de instituições.

Em suma, este projeto demonstrou que o uso de um guia didático estruturado pode ser uma ferramenta valiosa no ensino de Arduino, promovendo um aprendizado mais eficaz e engajador para os alunos. Acreditamos que nossas contribuições possam servir como base para futuras iniciativas educacionais e pesquisas na área de ensino de tecnologia e eletrônico

REFERÊNCIAS

BANZI, M. **Primeiros Passos com o Arduino**. 2. ed. São Paulo: Novatec, 2011. 152 p.

BARTZIK, F.; ZANDER, L. D. **A importância das aulas práticas de ciências no ensino fundamental**. @rquivo Brasileiro de Educação, v. 4, n. 8, p. 31-38, 26 mar. 2017.

BOTTENTUIT JUNIOR, João Batista; COUTINHO, Clara Pereira. **A integração do Google Sites no processo de ensino e aprendizagem: um estudo com alunos de licenciatura em matemática da Universidade Virtual do Maranhão**. In: VI Conferência Internacional de TIC na Educação, 2016, p. 387-390.

EGEA, R.; ROSA, R.; RODRÍGUEZ, D. **Uma plataforma aberta de prototipagem eletrônica como recurso para ensino**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2017. p. 1739. Disponível em: <https://doi.org/10.5753/cbie.sbie.2017.1739>. Acesso em: 20 jul. 2024.

EVANS, M.; NOBLE, J.; HOCHENBAUM, J. **Arduino em ação**. 1. ed. São Paulo: Novatec, 2013. 424 p.

FERA - FEIRA DE EXPOSIÇÃO REGIONAL ACADÊMICA. **Distribuição das notas da prova diagnóstica**. 27 set. 2024. Gráfico. Todos os direitos cedidos ao projeto THE WORLD OF ARDUINO: Guia pedagógico para a utilização do Arduino em ambientes educacionais, da teoria à prática.

FERA - FEIRA DE EXPOSIÇÃO REGIONAL ACADÊMICA. **Distribuição das notas da prova final**. 01 out. 2024. Gráfico. Todos os direitos cedidos ao projeto THE WORLD OF ARDUINO: Guia pedagógico para a utilização do Arduino em ambientes educacionais, da teoria à prática.

LOUSADA, Ricardo. **Componentes Básicos do Arduino – O que é Resistor, LED, Potenciômetro, Push Button**. Blog Elegrade, 2024. Disponível em: <https://blog.eletragate.com/componentes-basicos-do-arduino-o-que-e-resistor-led-potenciometro-push-button/>. Acesso em: 12 jun. 2024.

MERRILL, K. (2016). **Domains of Control: Governance of and by the Domain Name System**. Palgrave Macmillan US, New York, p. 89-106.

SILVA, J. L. D. S.; MELO, M. C.; CAMILO, R. D. S.; GALINDO, A. L.; VIANA, E. C. Plataforma Arduino integrado ao PLX-DAQ: **análise e aprimoramento de sensores com ênfase no LM35**. In: ESCOLA REGIONAL DE COMPUTAÇÃO BAHIA, ALAGOAS E SERGIPE (ERBASE), XIV, 2014, p. 1-10.

SOUZA, T. G. S. **Ensino de ciências e experimentação nos anos iniciais: da teoria à prática**. Revista Pró-Discente, Vitória, v. 25, n. 1, p. 41-53, 2019.

TAKAHASHI, R. T.; FERNANDES, M. F. P. **Plano de aula: conceitos e metodologia**. Acta Paulista de Enfermagem, v. 17, n. 1, p. 114-118, 2004.

VIEIRA, E.; MEIRELLES, R. M. S.; RODRIGUES, D. C. G. A. **O uso de tecnologias no ensino de química: a experiência do laboratório virtual Química Fácil**. In: Encontro Nacional de Pesquisa em Educação em Ciências, 2011, v. 8, p. 1-10.

APÊNDICES

LISTA DE EXERCÍCIOS (METODOLOGIA MAKER)

Nível 1 – Básico

1. Piscar LED com temporizador

Objetivo: Fazer um LED piscar com intervalos de 1 segundo.

2. Controle de LED com botão

Objetivo: Acender e apagar um LED pressionando um botão.

3. Controle de brilho de LED com potenciômetro

Objetivo: Ajustar o brilho do LED usando um potenciômetro.

4. Botão de alternância

Objetivo: Alternar entre ligado/desligado de um LED com um único botão (cada clique inverte o estado do LED).

5. Sequência de LEDs (Knight Rider)

Objetivo: Criar um efeito de LED piscando em sequência (como no carro do Knight Rider).

6. Botão para inverter sequência de LEDs

Objetivo: Controlar a direção da sequência de LEDs com um botão (esquerda/direita).

7. LED de alerta com botão duplo

Objetivo: Usar dois botões: um para acender e outro para apagar o LED.

8. Piscar LED com diferentes intervalos de tempo

Objetivo: Controlar o tempo que um LED fica ligado e desligado usando um botão para alternar entre tempos.

Nível 2 – Intermediário

9. Sensor ultrassônico com LED de proximidade

Objetivo: Acender um LED quando um objeto estiver a uma certa distância do sensor ultrassônico.

10. Controle de LED por tempo com botão

Objetivo: Acender um LED por um período fixo (ex: 5 segundos) após pressionar um botão.

11. Potenciômetro e LED piscando em diferentes frequências

Objetivo: Usar o potenciômetro para alterar a frequência com que o LED pisca.

12. Ultrassônico com buzzer (Alarme de proximidade)

Objetivo: Tocar o buzzer quando um objeto estiver próximo do sensor ultrassônico.

13. Jogo interativo utilizando o LCD Keypad Shield

Objetivo: Criar um jogo onde o usuário possa mover um pixel no display LCD Keypad Shield.

14. Controle de brilho de LED com 2 botões

Objetivo: Usar dois botões, um para aumentar e outro para diminuir o brilho do LED.

15. Sensor ultrassônico com LEDs (Barra de proximidade)

Objetivo: Usar múltiplos LEDs para indicar a distância de um objeto do sensor ultrassônico (quanto mais próximo, mais LEDs acendem).

16. Buzzer com ajuste de volume usando potenciômetro

Objetivo: Usar o potenciômetro para controlar a intensidade do som do buzzer.

17. Controle de LED com Sensor de Luz.

Objetivo: Acender um LED quando a luz ambiente estiver abaixo de um determinado nível

18. Controle de núcleos do LED RGB com Potenciômetro

Objetivo:

Controlar a cor do LED RGB usando um potenciômetro, mudando a cor de acordo com a posição do potenciômetro.

19. Controle de velocidade de transição de cores do LED RGB com potenciômetro

Objetivo: Mudar a cor do LED RGB em loop e controlar a velocidade com o potenciômetro.

20. Leitor de biometria: Captura de impressão digital

Objetivo: Usar o leitor de biometria para capturar e armazenar impressões digitais.

21. Controle de acesso com leitor de biometria e LED

Objetivo: Acender um LED se a impressão digital capturada pelo leitor de biometria for reconhecida.

22. Sistema de segurança com sensor ultrassônico e leitor de biometria

Objetivo: Tocar um alarme (buzzer) se o sensor ultrassônico detectar movimento e a impressão digital não for reconhecida.

23. Display mostrando leitura do potenciômetro

Objetivo: Mostrar o valor lido pelo potenciômetro em um display de 7 segmentos.

24. Sistema de autenticação com biometria e botão

Objetivo: Usar o leitor de biometria para autenticação, acendendo um LED se a impressão digital for válida, e usando um botão para tentar novamente.

25. Sensor ultrassônico com 3 LEDs (Semáforo de proximidade)

Objetivo: Usar o sensor ultrassônico para acender 3 LEDs como um semáforo (verde, amarelo e vermelho) de acordo com a distância do objeto.

26. Display de 7 segmentos com contagem automática

Objetivo: Criar um contador que se atualiza automaticamente no display, sem usar botões.

27. Controle de servos com leitor de biometria

Objetivo: Usar o leitor de biometria para acionar um servo motor, movendo-o quando uma impressão digital for reconhecida (se você adquirir um servo futuramente).

28. Alarme sonoro com leitor de biometria e botão de cancelamento

Objetivo: Ativar um alarme com o buzzer se o leitor de biometria não reconhecer a impressão digital, e usar um botão para cancelar o alarme.

29. Potenciômetro ajustando tom do buzzer

Objetivo: Usar o potenciômetro para controlar o tom gerado pelo buzzer.

30. Sistema de medição de distância com ajuste de precisão

Objetivo: Usar o sensor ultrassônico para medir a distância e ajustar a precisão da leitura usando o potenciômetro, exibindo o resultado no display.

RESOLUÇÃO

EXERCÍCIO 1:

Objetivo: fazer um LED piscar com intervalos de 1 segundo.

Materiais:

- 1 LED
- 1 resistor (220Ω)
- 1 Arduino (Uno, Nano, etc.)
- Cabos jumpers
- Protoboard

Passo a Passo:

Montagem do circuito:

- Conecte o terminal longo (positivo) do LED a um dos pinos digitais do Arduino (ex.: pino 13).
- Conecte o terminal curto (negativo) do LED a um resistor de 220Ω.
- O outro terminal do resistor deve ser conectado ao **GND** do Arduino.

Código para o Arduino: disponível no site The World of Arduino.

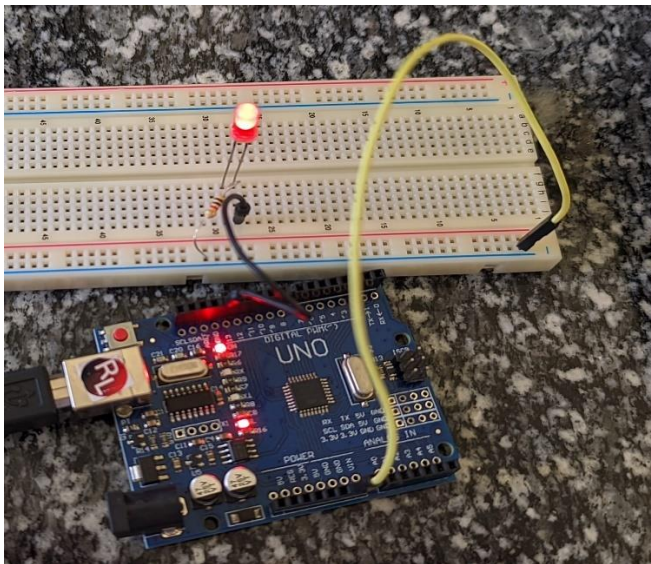
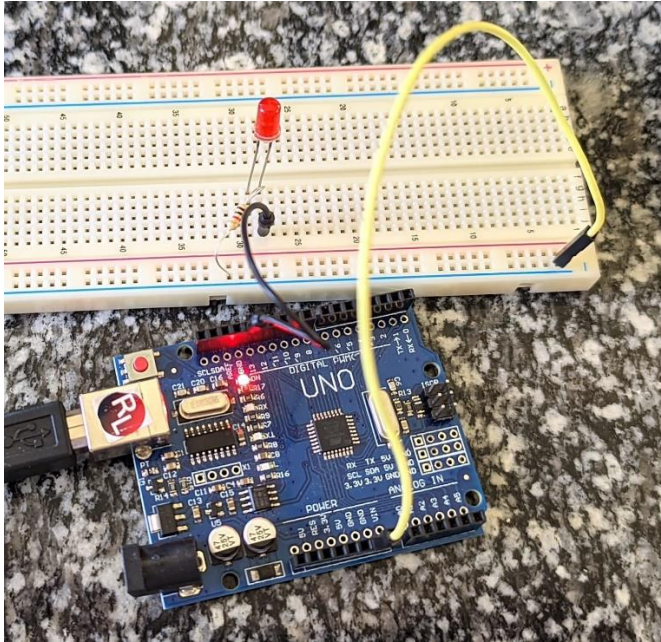
Explicação do código:

- **setup():** Aqui você define o pino 13 como saída, o que permite controlar o LED.
- **loop():** O código nesta função será repetido indefinidamente. O LED fica ligado por 1 segundo (1000 milissegundos) e desligado por 1 segundo, criando o efeito de piscar.

Carregar do código:

- Conecte o Arduino ao seu computador via cabo USB.
- Abra o Arduino IDE, copie o código acima e cole no IDE.
- Selecione a placa e a porta correta no menu Ferramentas.
- Clique em “Upload” para enviar o código ao Arduino.

Resultado esperado: o LED piscará com intervalos de 1 segundo, ligando e desligando continuamente.



EXERCÍCIO 2:

Objetivo: acender e apagar um LED pressionando um botão.

Materiais:

- 1 LED
- 1 resistor (220Ω para o LED)
- 1 resistor ($10k\Omega$ para o botão)
- 1 botão (push button)
- 1 Arduino (Uno, Nano, etc.)
- Cabos jumpers
- Protoboard

Passo a Passo:

Montagem do circuito:

- Conecte o terminal longo (positivo) do LED ao pino digital **13** do Arduino.
- O terminal curto (negativo) do LED deve ser conectado a um resistor de **220Ω**.
- O outro terminal do resistor vai para o **GND** do Arduino.
- Conecte um terminal do botão ao **+5V** do Arduino.
- O outro terminal do botão será conectado ao pino digital **7** do Arduino.
- Conecte um resistor de **10kΩ** entre o pino **7** e o **GND** para garantir que a leitura do botão seja estável quando não pressionado.

Código para o Arduino: disponível no site The World of Arduino.

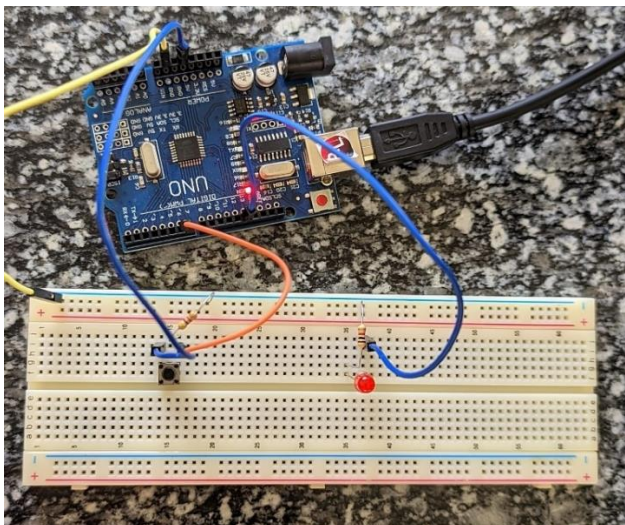
Explicação do código:

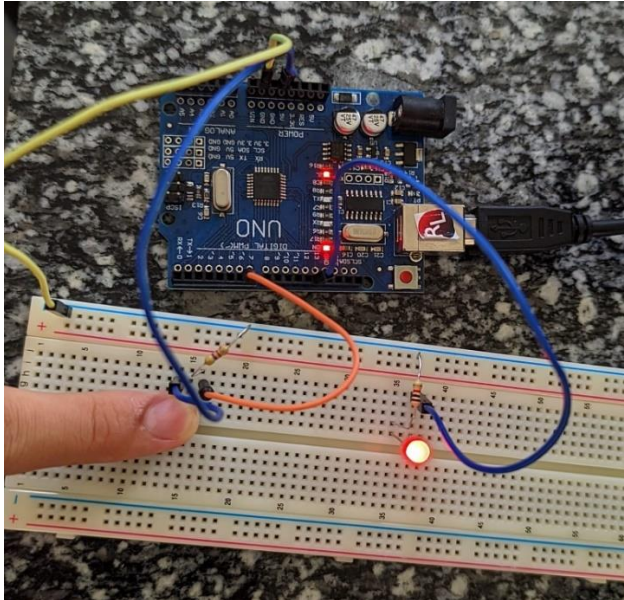
- **setup():** Aqui você define o pino 13 como saída (LED) e o pino 7 como entrada (botão).
- **loop():** O código lê continuamente o estado do botão:
- Se o botão for pressionado (HIGH), o LED acende.
- Se o botão não for pressionado (LOW), o LED apaga.

Upload do código:

- Conecte o Arduino ao computador via cabo USB.
- Abra o Arduino IDE, copie o código acima e cole na IDE.
- Selecione a placa e a porta corretas no menu Ferramentas.
- Clique em "Upload" para enviar o código ao Arduino.

Resultado esperado: ao pressionar o botão, o LED acenderá. Quando soltar o botão, o LED apagará.





EXERCÍCIO 3:

Objetivo: ajuste o brilho do LED usando um potenciômetro.

Materiais:

- 1 LED
- 1 resistor (220Ω para o LED)
- 1 potenciômetro ($10k\Omega$)
- 1 Arduino (Uno, Nano, etc.)
- Cabos jumpers
- Protoboard

Passo a Passo:

Montagem do circuito:

- Conecte o terminal longo (positivo) do LED ao pino digital **9** do Arduino.
- O terminal curto (negativo) do LED deve estar conectado a um resistor de **220Ω** .
- O outro terminal do resistor vai para o **GND** do Arduino.
- Conecte o terminal do meio do potenciômetro ao pino analógico **A0** do Arduino.
- Conecte um dos terminais externos do potenciômetro ao **+5V** do Arduino e o outro terminal externo ao **GND** do Arduino.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do código:

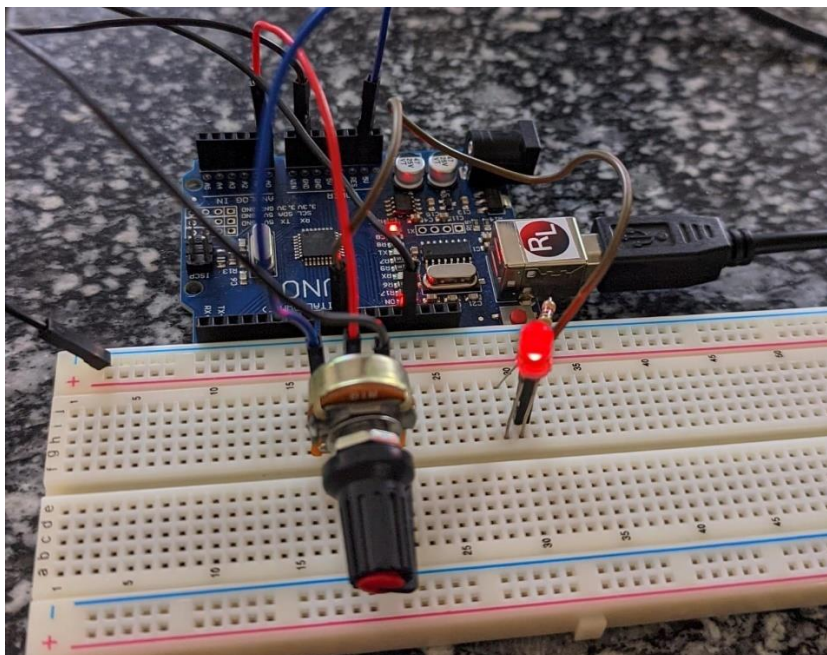
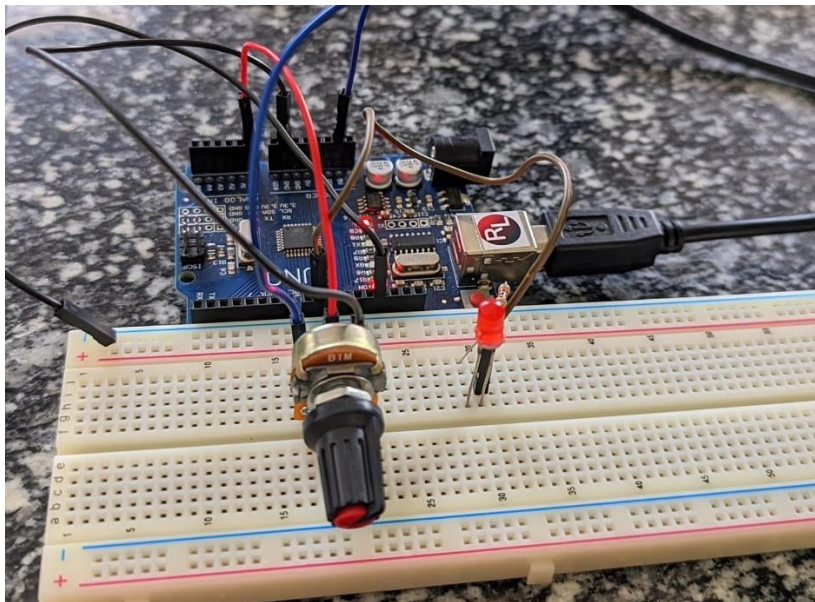
- `setup()`: Define o pino 9 como saída (para o LED).
- `loop()`: Lê o valor do potenciômetro (0 a 1023) e o mapa para o intervalo de 0 a 255, que é o intervalo de PWM do LED. O `analogWrite()` ajusta o brilho do LED com base nesse valor.

Carregar do código:

- Conecte o Arduino ao computador via cabo USB.
- Abra o Arduino IDE, copie o código acima e cole no IDE.
- Selecione a placa e a porta correta no menu Ferramentas.
- Clique em “Upload” para enviar o código ao Arduino.

Resultado esperado:

Ao girar o potenciômetro, o brilho do LED deve aumentar ou diminuir, dependendo da posição do potenciômetro.

**EXERCÍCIO 4:**

Objetivo: alternar entre ligado e desligado de um LED com um único botão (cada clique invertido

o estado do LED).

Materiais:

- 1 LED
- 1 resistor (220Ω para o LED)
- 1 botão (push button)
- 1 resistor (10kΩ para o botão)
- 1 Arduino (Uno, Nano, etc.)
- Cabos jumpers
- Protoboard

Passo a Passo:**Montagem do circuito:**

- Conecte o terminal longo (positivo) do LED ao pino digital **13** do Arduino.
- O terminal curto (negativo) do LED deve estar conectado a um resistor de **220Ω**.
- O outro terminal do resistor vai para o **GND** do Arduino.
- Conecte um terminal do botão ao **+5V** do Arduino.
- O outro terminal do botão será conectado ao pino digital **7** do Arduino.
- Conecte um resistor de **10kΩ** entre o pino **7** e o **GND** para garantir uma leitura estável quando o botão não estiver pressionado.

Código para o Arduino: disponível no site The World of Arduino.

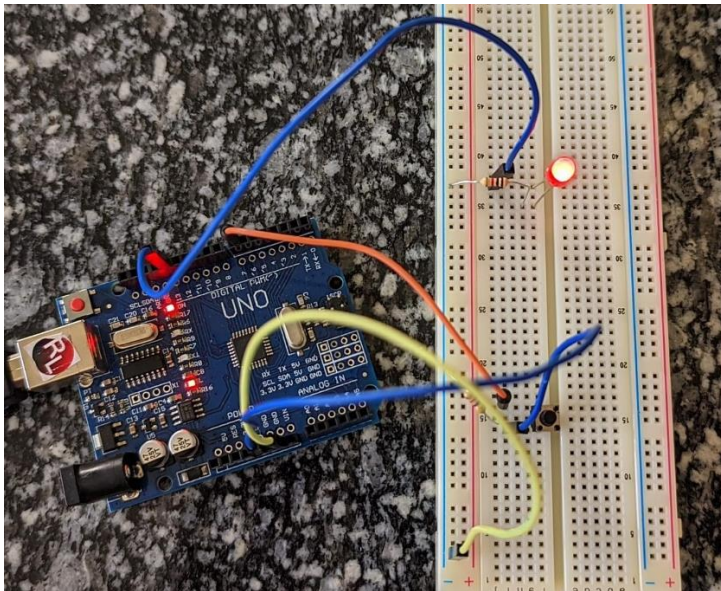
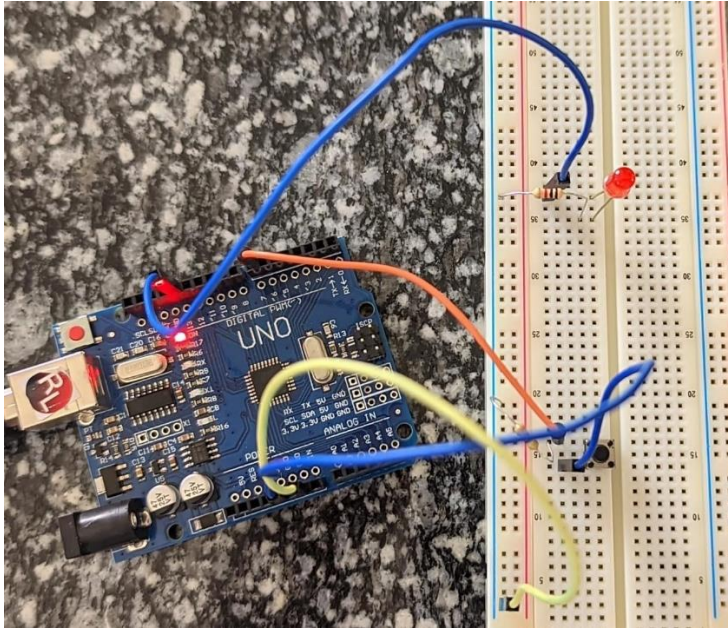
Explicação do código:

- **setup():** Define o pino 13 como saída (LED) e o pino 7 como entrada (botão).
- **loop():** O código verifica o estado do botão:
- Se o botão mudar de LOW para HIGH (pressionado), o estado do LED é invertido.
- O estado atual do botão é salvo para verificar a transição na próxima leitura.

Carregar do código:

- Conecte o Arduino ao computador via cabo USB.
- Abra o Arduino IDE, copie o código acima e cole no IDE.
- Selecione a placa e a porta correta no menu Ferramentas.
- Clique em “Upload” para enviar o código ao Arduino.

Resultado esperado: cada vez que você abrir o botão, o LED deverá alternar entre aceso e apagado.



EXERCÍCIO 5:

Objetivo: crie um efeito de LED piscando em sequência (como no carro do Knight Rider).

Materiais:

- 8 LEDs
- 8 resistores (220Ω para cada LED)
- 1 Arduino (Uno, Nano, etc.)
- Cabos jumpers
- Protoboard

Passo a Passo:

Montagem do circuito:

- Conecte os terminais longos (positivos) de cada LED aos pinos digitais do Arduino de 2 a 9 (ou conforme desejado).
- Conecte os terminais curtos (negativos) de cada LED a um resistor de 220Ω .
- O outro terminal de cada resistor deve ser conectado ao **GND** do Arduino.

Código para o Arduino: disponível no site [The World of Arduino](#).

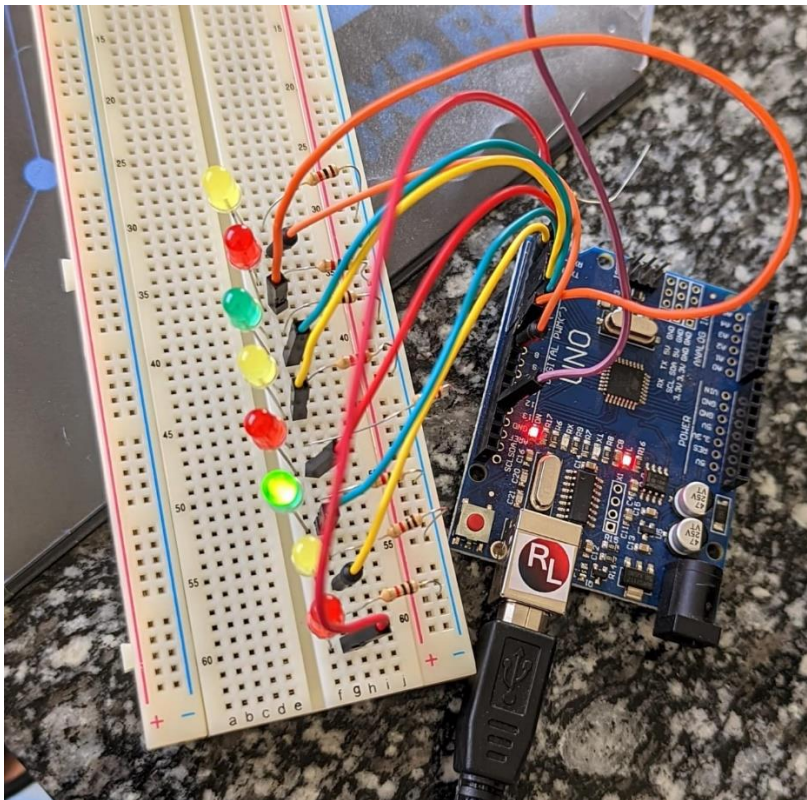
Explicação do código:

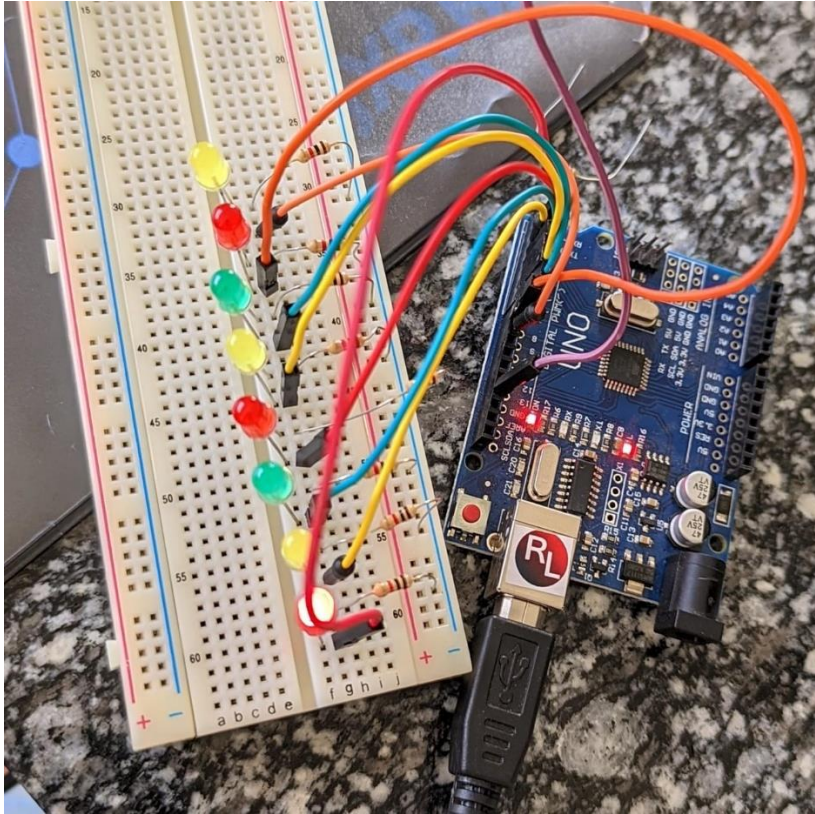
- **setup():** Define os pinos dos LEDs como saída.
- **loop():** O código realiza dois loops:
 - O primeiro acende os LEDs da esquerda para a direita.
 - O segundo acende os LEDs da direita para a esquerda.
- Em cada iteração, há um pequeno atraso (`delay(100)`) para criar a ilusão de movimento.

Carregar do código:

- Conecte o Arduino ao computador via cabo USB.
- Abra o Arduino IDE, copie o código acima e cole no IDE.
- Selecione a placa e a porta correta no menu Ferramentas.
- Clique em “Upload” para enviar o código ao Arduino.

Resultado esperado: os LEDs devem acender em sequência, imitando o efeito do carro de Knight Rider, alternando a direção da sequência.





EXERCÍCIO 6:

Objetivo: controle a direção da sequência de LEDs com um botão (esquerda/direita).

Materiais:

- 8 LEDs
- 8 resistores (220Ω para cada LED)
- 1 botão (push button)
- 1 resistor ($10k\Omega$ para o botão)
- 1 Arduino (Uno, Nano, etc.)
- Cabos jumpers
- Protoboard

Passo a Passo:

Montagem do circuito:

- Conecte os terminais longos (positivos) de cada LED aos pinos digitais do Arduino de **2 a 9** (ou conforme desejado).
- Conecte os terminais curtos (negativos) de cada LED a um resistor de **220Ω** .
- O outro terminal de cada resistor deve ser conectado ao **GND** do Arduino.
- Conecte um terminal do botão ao **+5V** do Arduino.
- Conecte o outro terminal do botão ao pino digital **10** do Arduino.
- Conecte um resistor de **$10k\Omega$** entre o pino **10** e o **GND** para garantir uma leitura estável quando o botão não estiver pressionado.

Código para o Arduino: disponível no site The World of Arduino.

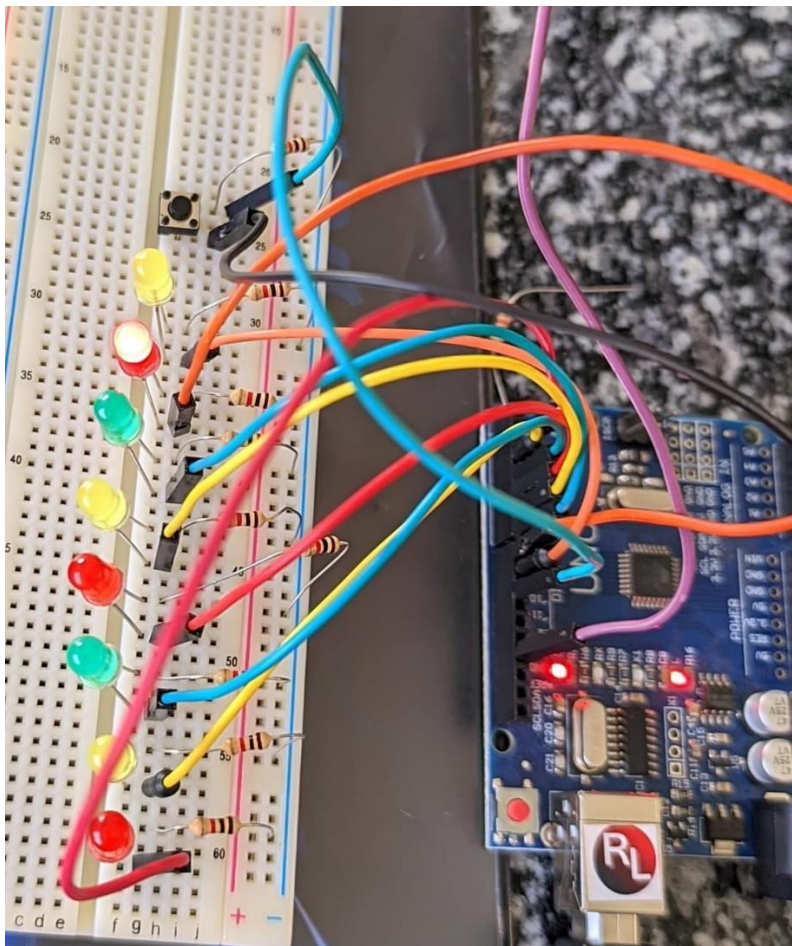
Explicação do código:

- **setup():** Define os pinos dos LEDs como saída e o pino do botão como entrada.
- **loop():** O código verifica o estado do botão:
- Se o botão for pressionados (HIGH), inverte a direção da sequência (variável reverse).
- Dependendo do estado da variável reverse, acenda os LEDs da esquerda para a direita ou da direita para a esquerda.
- Um pequeno atraso (delay(200)) é utilizado para evitar leituras falsas do botão.

Carregar do código:

- Conecte o Arduino ao computador via cabo USB.
- Abra o Arduino IDE, copie o código acima e cole no IDE.
- Selecione a placa e a porta correta no menu Ferramentas.
- Clique em “Upload” para enviar o código ao Arduino.

Resultado esperado: os LEDs devem lançar em sequência, e a direção da sequência deve mudar a cada suspensão do botão.



EXERCÍCIO 7:

Objetivo: usar dois botões: um para acender e outro para apagar o LED.

Materiais:

- 1 LED
- 1 resistor (220Ω para o LED)

- 2 botões (push button)
- 2 resistores (10k Ω para cada botão)
- 1 Arduino (Uno, Nano, etc.)
- Cabos jumpers
- Protoboard

Passo a Passo:

Montagem do circuito:

- Conecte o terminal longo (positivo) do LED ao pino digital **2** do Arduino.
- Conecte o terminal curto (negativo) do LED a um resistor de **220 Ω** e o outro terminal do resistor ao **GND** do Arduino.
- Conecte um terminal de um botão ao +5V do Arduino e o outro terminal ao pino digital **3** (para acender o LED).
- Conecte um resistor de **10k Ω** entre o pino **3** e o **GND**.
- Repita o processo para o segundo botão, conectando-o ao pino digital **4** (para desligar o LED) e usando outro resistor de **10k Ω** entre o pino **4** e o **GND**.

Código para o Arduino: disponível no site The World of Arduino.

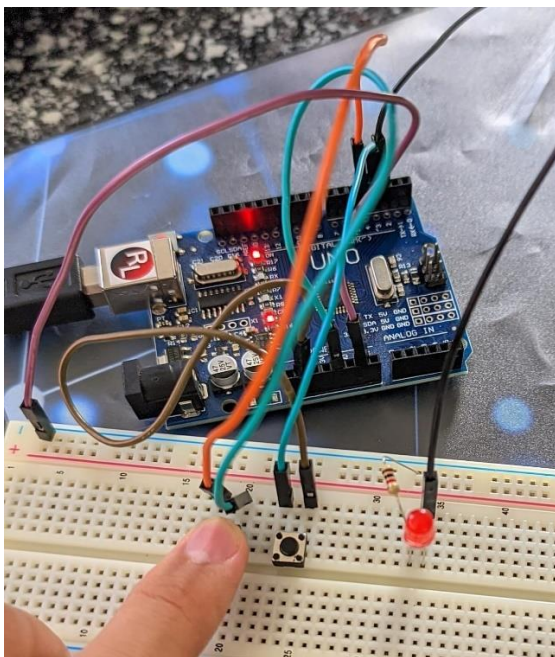
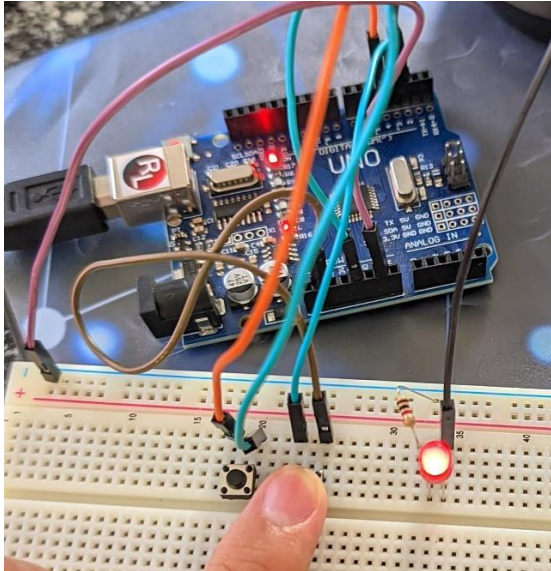
Explicação do código:

- **setup():** Define o pino do LED como saída e os pinos dos botões como entrada.
- **loop():** O código verifica continuamente o estado dos botões:
- Se o botão de acender para botões pressionados (HIGH), o LED acender.
- Se o botão de desligamento dos botões pressionados (HIGH), o LED apaga.

Carregar do código:

- Conecte o Arduino ao computador via cabo USB.
- Abra o Arduino IDE, copie o código acima e cole no IDE.
- Selecione a placa e a porta correta no menu Ferramentas.
- Clique em “Upload” para enviar o código ao Arduino.

Resultado esperado: o LED deve acender ao pressionar o botão designado para acendê-lo e desligar ao pressionar o botão designado para apagá-lo.



EXERCÍCIO 8:

Objetivo: controle o tempo que um LED fica ligado e desligado usando um botão para alternar entre tempos.

Materiais:

- 1 LED
- 1 resistor (220Ω para o LED)
- 1 botão (push button)
- 1 resistor ($10k\Omega$ para o botão)
- 1 Arduino (Uno, Nano, etc.)
- Cabos jumpers
- Protoboard

Passo a Passo:

Montagem do circuito:

- Conecte o terminal longo (positivo) do LED ao pino digital 2 do Arduino.

- Conecte o terminal curto (negativo) do LED a um resistor de **220Ω** e o outro terminal do resistor ao **GND** do Arduino.
- Conecte um terminal do botão ao **+5V** do Arduino e o outro terminal ao pino digital **3**.
- Conecte um resistor de **10kΩ** entre o pino **3** e o **GND**.

Código para o Arduino: disponível no site The World of Arduino.

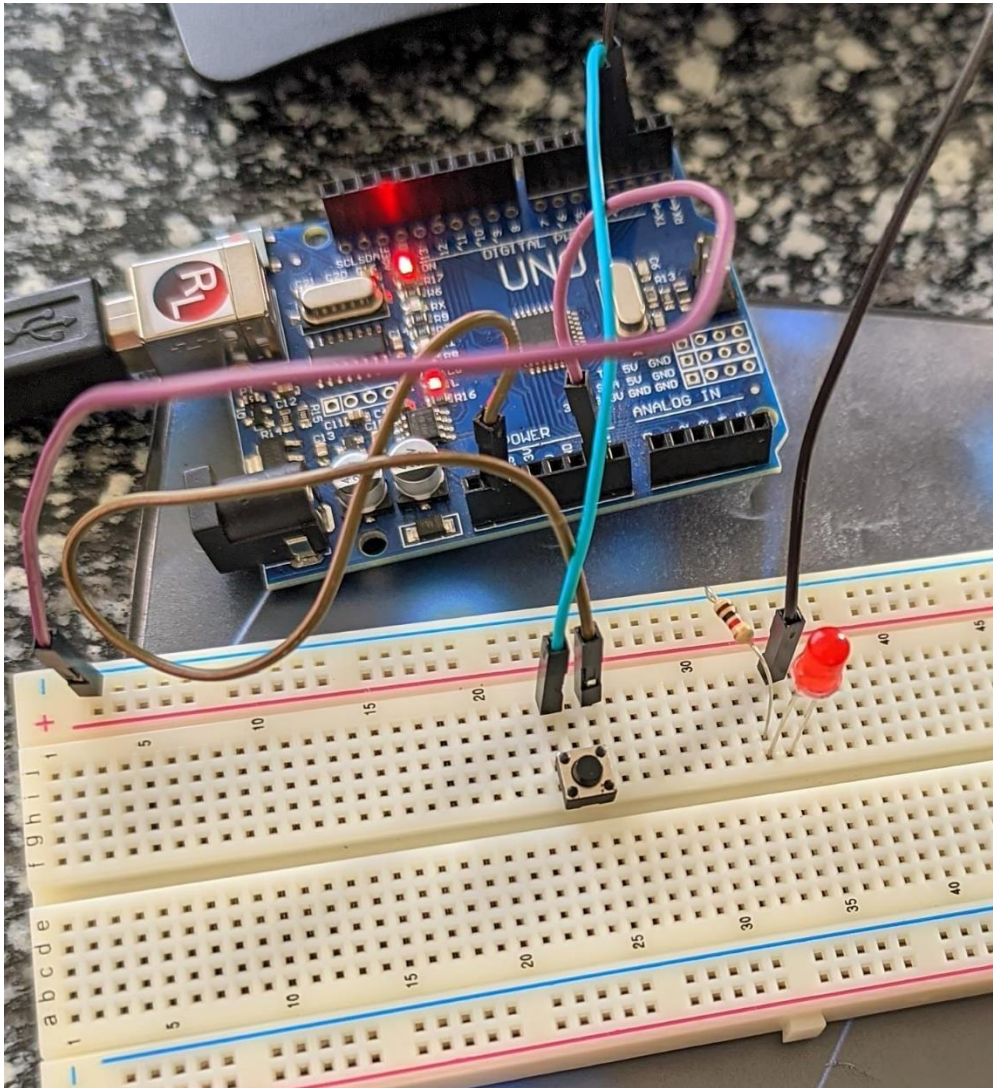
Explicação do código:

- **setup():** Define o pino do LED como saída e o pino do botão como entrada.
- **loop():** O código verifica continuamente o estado do botão:
- Quando o botão é pressionado, alterna entre dois intervalos (1 segundo e 500 milissegundos).
- Um controle de debounce é utilizado para evitar múltiplas leituras ao pressionar o botão.
- O LED pisca de acordo com o intervalo atual.

Carregar do código:

- Conecte o Arduino ao computador via cabo USB.
- Abra o Arduino IDE, copie o código acima e cole no IDE.
- Selecione a placa e a porta correta no menu Ferramentas.
- Clique em “Upload” para enviar o código ao Arduino.

Resultado esperado: o LED deve piscar com um intervalo de 1 segundo inicialmente, e ao pressionar o botão, deve alternar para um intervalo de 500 milissegundos e vice-versa.



EXERCÍCIO 9:

Objetivo: acenda um LED quando um objeto estiver a uma certa distância do sensor ultrassônico.

Materiais:

- 1 módulo de sensor ultrassônico (HC-SR04)
- 1 LED
- 1 resistor (220Ω para o LED)
- 1 Arduino (Uno, Nano, etc.)
- Cabos jumpers
- Protoboard

Passo a Passo:

Montagem do circuito:

Sensor Ultrassônico:

- Conecte o pino **VCC** do HC-SR04 ao **+5V** do Arduino.
- Conecte o pino **GND** do HC-SR04 ao **GND** do Arduino.
- Conecte o pino **Trig** do HC-SR04 ao pino digital **9** do Arduino.

- Conecte o pino **Echo** do HC-SR04 ao pino digital **10** do Arduino.

LIDERADO:

- Conecte o terminal longo (positivo) do LED ao pino digital **2** do Arduino.
- Conecte o terminal curto (negativo) do LED a um resistor de **220Ω** e o outro terminal do resistor ao **GND** do Arduino.

Código para o Arduino: disponível no site The World of Arduino.

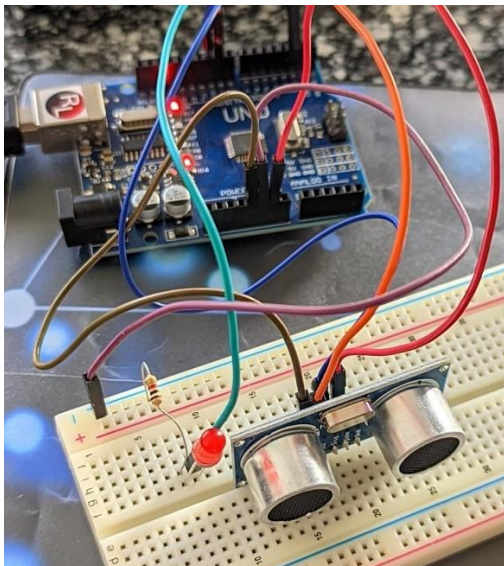
Explicação do código:

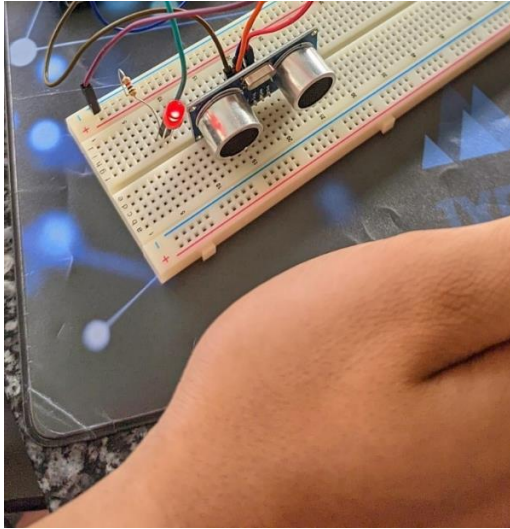
- **setup():** Inicializa os pinos do sensor e do LED, e configura a comunicação serial.
- **loop():** O código envia um pulso para o pino **Trig** , mede o tempo que o sinal leva para retornar ao pino **Echo** , calcula a distância e acende o LED se a distância for menor que 20 cm.

Carregar do código:

- Conecte o Arduino ao computador via cabo USB.
- Abra o Arduino IDE, copie o código acima e cole no IDE.
- Selecione a placa e a porta correta no menu Ferramentas.
- Clique em “Upload” para enviar o código ao Arduino.

Resultado esperado: quando um objeto estiver a menos de 20 cm do sensor ultrassônico, o LED deverá acender. Caso contrário, ele permanecerá desligado.





EXERCÍCIO 10:

Objetivo: acender um LED por um período fixo (ex: 5 segundos) após pressionar um botão.

Materiais Necessários:

- 1 LED
- 1 Resistor (220 ohms)
- 1 Botão (push button)
- Fios de conexão
- 1 Placa Arduino

Montagem do Circuito:

LIDERADO:

- Conecte o terminal positivo (anodo) do LED ao pino **9** do Arduino.
- Conecte o terminal negativo (catodo) ao **GND** através de um resistor de 220 ohms.

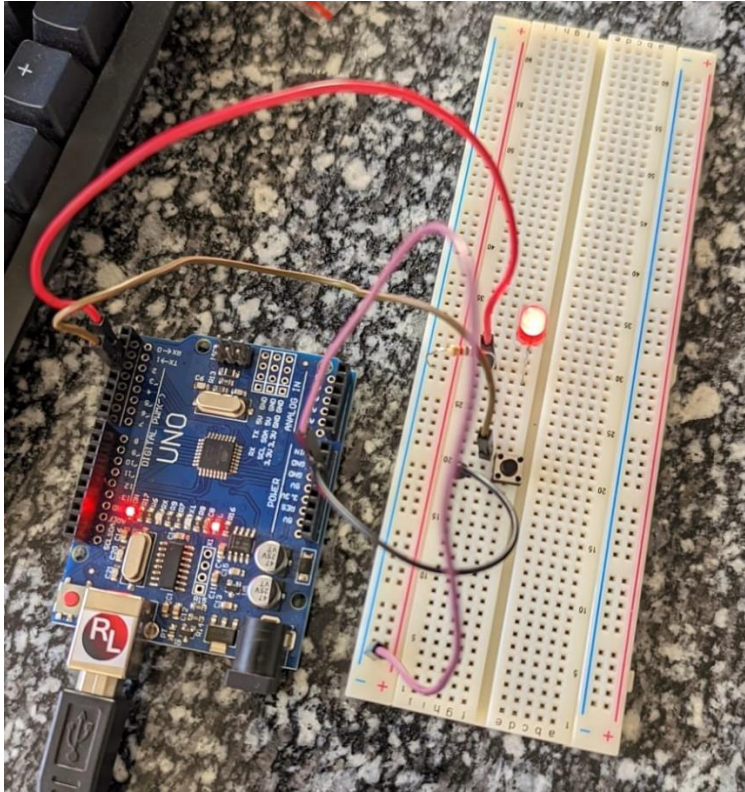
Botão:

- Conecte um lado do botão ao pino 7 do Arduino.
- Conecte o outro lado do botão ao GND .
- Conecte um resistor pull-up (de 10k ohms) entre o pino 7 e o 5V do Arduino para evitar falsos acionamentos.
- Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- **buttonPin:** Pino onde o botão está conectado.
- **ledPin:** Pino onde o LED está conectado.
- **digitalRead():** Verifique se o botão foi pressionado (estado HIGH).
- **digitalWrite():** Liga o LED ao iniciar o botão e o mantém ligado por 5 segundos usando a função delay().

Resultado Esperado: ao pressionar o botão, o LED acenderá por 5 segundos e depois desligará automaticamente. Se o botão for pressionado novamente, o LED repetirá o ciclo.



EXERCÍCIO 11:

Objetivo: usar o potenciômetro para alterar a frequência com que o LED pisca.

Materiais Necessários:

- 1 LED
- 1 Resistor de 220 ohms (para o LED)
- 1 Potenciômetro de 10k ohms
- Fios de conexão
- 1 Placa Arduino

Montagem do Circuito:

LIDERADO:

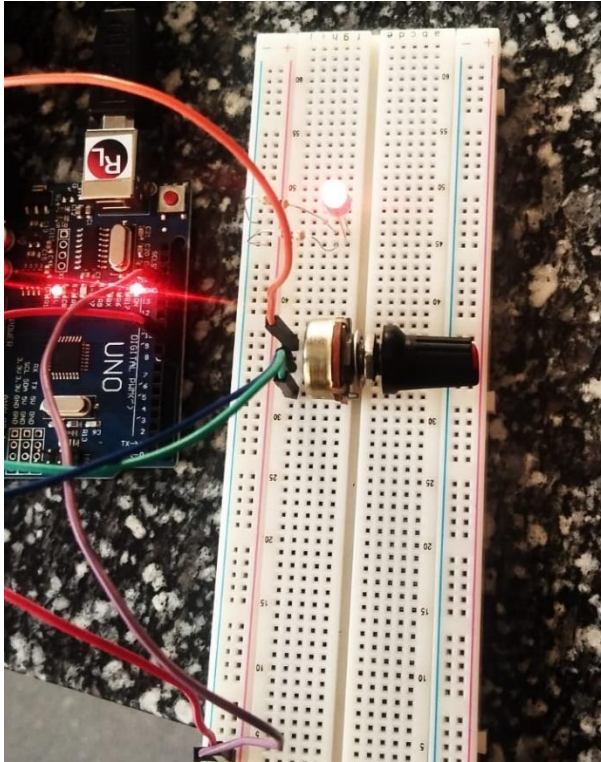
- O terminal longo (ânodo) do LED vai para o pino **9** do Arduino (com um resistor de 220 ohms em série).
- O terminal curto (cátodo) do LED vai para o **GND** do Arduino.

Potenciômetro:

- Conecte um terminal do potenciômetro ao **5V** do Arduino.
- Conecte o outro terminal do potenciômetro ao **GND** do Arduino.
- Conecte o terminal do meio do potenciômetro ao pino **A0** do Arduino.

Código para o Arduino: disponível no site The World of Arduino.

Resultado Esperado: o LED piscará em uma frequência que varia de acordo com a posição do potenciômetro. Quando o potenciômetro estiver em sua posição mínima, o LED piscaria rapidamente (a cada 50 ms). Quando estiver em sua posição máxima, o LED piscaria lentamente (a cada 1000 ms).



EXERCÍCIO 12:

Objetivo: tocar o buzzer quando um objeto estiver próximo do sensor ultrassônico.

Materiais Necessários:

- 1 Sensor Ultrassônico (HC-SR04)
- 1 campainha
- Fios de conexão
- 1 Placa Arduino

Montagem do Circuito:

- **Sensor Ultrassônico (HC-SR04):**
- Conecte o pino **VCC** ao **5V** do Arduino.
- Conecte o pino **GND** ao **GND** do Arduino.
- Conecte o pino **Trig** ao pino **7** do Arduino.
- Conecte o pino **Echo** ao pino **6** do Arduino.

Buzzer:

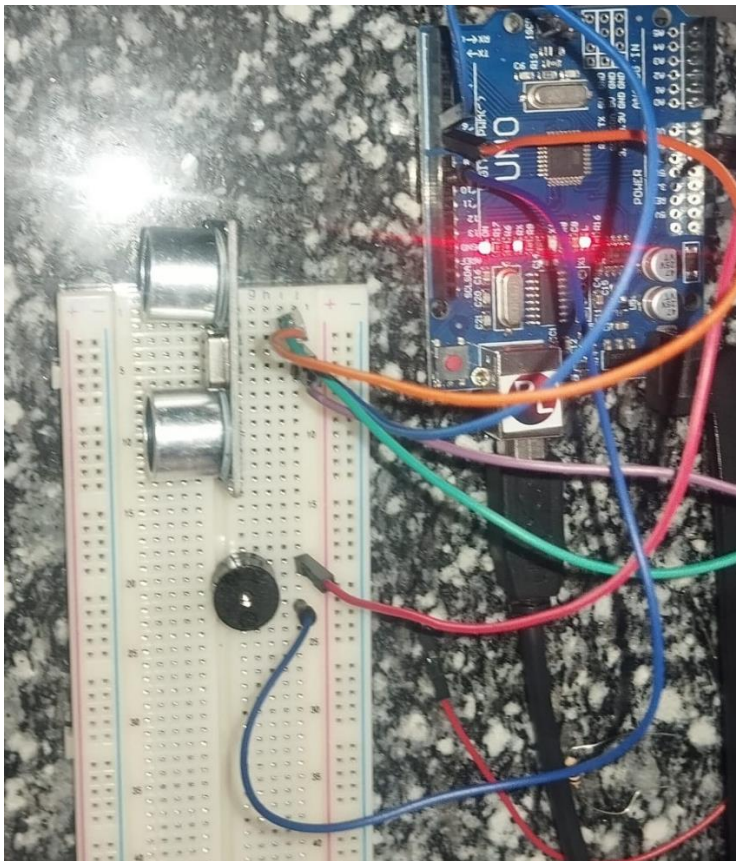
- Conecte o terminal positivo do buzzer ao pino **9** do Arduino.
- Conecte o terminal negativo do buzzer ao **GND** do Arduino.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- **trigPine echoPin:** Pinos usados para o sensor ultrassônico.
- **buzzerPin:** Pino usado para campainha.
- **duratione distance:** Variáveis para armazenar o tempo de duração do pulso ultrassônico e a distância calculada.
- **pulseIn():** Mede o tempo que o pulso ultrassônico leva para retornar.
- **distance:** A distância é calculada multiplicando o tempo de duração do pulso por 0,034 e dividindo por 2 (porque o pulso vai e volta).
- **tone():** Ativa o buzzer quando um objeto está a menos de 20 cm do sensor.
- **noTone():** Desliga o som quando a distância é maior que 20 cm.

Resultado Esperado: o buzzer emitirá um som quando um objeto estiver a menos de 20 cm do sensor ultrassônico, criando um "alarme" de proximidade. Se o objeto for afastado, o buzzer para emitir algo.



EXERCÍCIO 13:

Objetivo: Criar um jogo onde o usuário possa mover um pixel no display LCD Keypad Shield.

Materiais Necessários:

- Arduino Uno
- Display LCD Keypad Shield.

Montagem do Circuito:

- O LCD Keypad Shield se conecta diretamente ao Arduino, portanto não há necessidade de montagem extra.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- **#include <LiquidCrystal.h>:** Inclui uma biblioteca para controlar o LCD.
- **LiquidCrystal lcd(8, 9, 4, 5, 6, 7):** Define os pinos do LCD com base na configuração do Keypad Shield.
- **posX e posY:** Variáveis que armazenam na posição atual do pixel.
- **lcd.begin(16, 2):** Inicia o LCD com 16 colunas e 2 linhas.
- **lcd.clear():** Limpa o LCD a cada iteração do loop.
- **lcd.setCursor(posX, posY):** Define a posição do cursor no LCD.
- **lcd.print("*"):** Exibe o pixel na posição atual.
- **analogRead(A0):** Lê a entrada analógica do botão.
- **lógica de controle de movimento:** Verifica qual botão foi pressionado e ajusta a posição do pixel de acordo, incluindo a lógica para redefinir a posição.

Resultados Esperados: o usuário deve ser capaz de mover o pixel na tela usando os botões do LCD Keypad Shield. O pixel será representado pelo caractere "*".



EXERCÍCIO 14:

Objetivo: Usar dois botões, um para aumentar e outro para diminuir o brilho do LED.

Materiais Necessários:

- 1 Arduino Uno
- 1 LED
- 2 resistores (220Ω para o LED e 10kΩ para os botões)
- 2 botões
- Jumpers
- Protoboard

Montagem do Circuito:**LIDERADO:**

- O terminal longo (ânodo) do LED se conecta a um pino PWM (ex: pino 9) do Arduino.
- O terminal curto (cátodo) se conecta ao GND através de um resistor de 220Ω.

Botões:

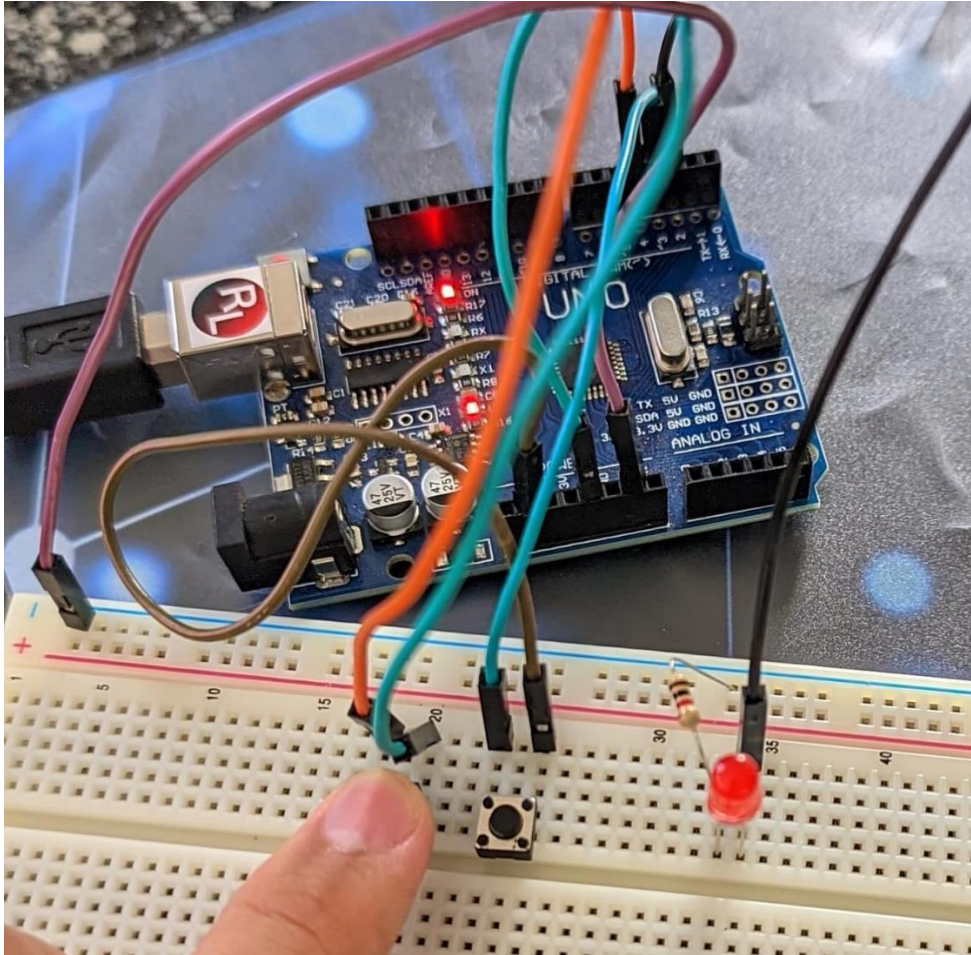
- Conecte um lado do primeiro botão a um pino digital (ex: pino 2) e o outro lado ao GND.
- Conecte um lado do segundo botão a outro pino digital (ex: pino 3) e o outro lado ao GND.
- Use resistores de pull-up internos ou externos para garantir que os botões funcionem corretamente.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- **const int ledPin = 9;** Definir o pino do LED.
- **const int buttonIncrease = 2;** Defina o pino do botão para aumentar o brilho.
- **const int buttonDecrease = 3;** Defina o pino do botão para diminuir o brilho.
- **int brightness = 0;** Nível de brilho inicial (0-255).
- **pinMode(ledPin, OUTPUT);** Define o pino do LED como saída.
- **pinMode(buttonIncrease, INPUT_PULLUP);** Defina o pino do botão de aumentar como entrada com pull-up interno.
- **pinMode(buttonDecrease, INPUT_PULLUP);** Defina o pino do botão de diminuição como entrada com pull-up interno.
- **Lógica de controle de brilho:** Aumenta ou diminui o brilho do LED com base no botão apertado, limitando o brilho entre 0 e 255.

Resultados Esperados: o usuário deve ser capaz de aumentar ou diminuir o brilho do LED usando os dois botões.



EXERCÍCIO 15

Objetivo: Usar múltiplos LEDs para indicar a distância de um objeto do sensor ultrassônico (quanto mais próximo, mais LEDs acendem).

Materiais Necessários:

- 1 Arduino Uno
- 1 Sensor Ultrassônico HC-SR04
- 5 LEDs
- 5 resistores (220Ω para os LEDs)
- Fios de conexão
- Protoboard

Montagem do Circuito:

Sensor Ultrassônico:

- O pino VCC do sensor se conecta ao 5V do Arduino.
- O pino GND do sensor se conecta ao GND do Arduino.
- O pino TRIG do sensor se conecta a um pino digital (ex: pino 9) do Arduino.
- O pino ECHO do sensor se conecta a um pino digital (ex: pino 10) do Arduino.

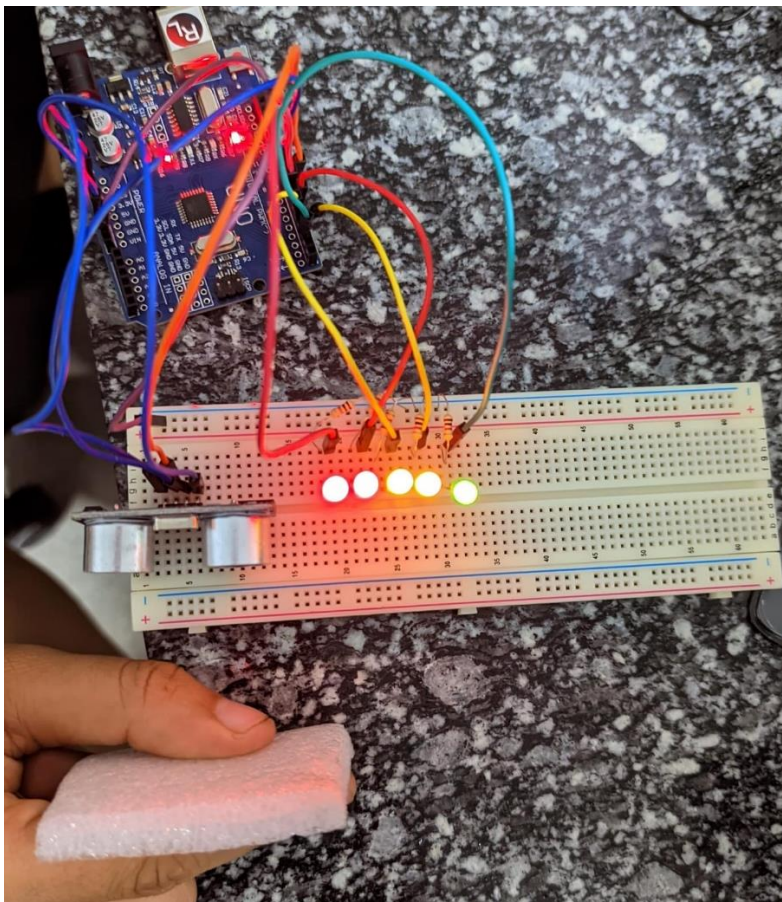
LEDs:

- Conecte o terminal longo (ânodo) de cada LED a um pino digital do Arduino (ex: pinos 2, 3, 4, 5 e 6).
- O terminal curto (cátodo) de cada LED se conecta ao GND através de um resistor de 220Ω.
- **Código para o Arduino:** disponível no site The World of Arduino.

Explicação do Código:

- **const int trigPin = 9;;** Define o pino do TRIG do sensor ultrassônico.
- **const int echoPin = 10;;** Define o pino do ECHO do sensor ultrassônico.
- **const int ledPins[] = {2, 3, 4, 5, 6};;** Define os pinos dos LEDs.
- **const int numLeds = 5;;** Número de LEDs.
- **pinMode(trigPin, OUTPUT);** Define o pino TRIG como saída.
- **pinMode(echoPin, INPUT);** Define o pino ECHO como entrada.
- **Lógica de leitura de distância:** Envia um pulso do sensor ultrassônico e calcula a distância com base no tempo que o eco leva para voltar.
- **Controle de LEDs:** Acende os LEDs com base na distância medida; cada LED se acende quando um objeto estiver a 20 cm ou menos.

Resultados Esperados: os LEDs devem acender em uma barra de proximidade, indicando a distância de um objeto em relação ao sensor ultrassônico. Quanto mais próximo o objeto, mais LEDs devem acender.



EXERCÍCIO 16:

Objetivo: usando um potenciômetro para ajustar o volume do buzzer.

Materiais Necessários:

- 1 Arduino Uno.
- 1 Buzzer piezoelétrico.
- 1 Potenciômetro.
- Fios de conexão.
- Protoboard.

Montagem do Circuito:**Buzzer:**

- O terminal positivo do buzzer se conecta a um pino digital (ex: pino 8) do Arduino.
- O terminal negativo do buzzer se conecta ao GND do Arduino.

Potenciômetro:

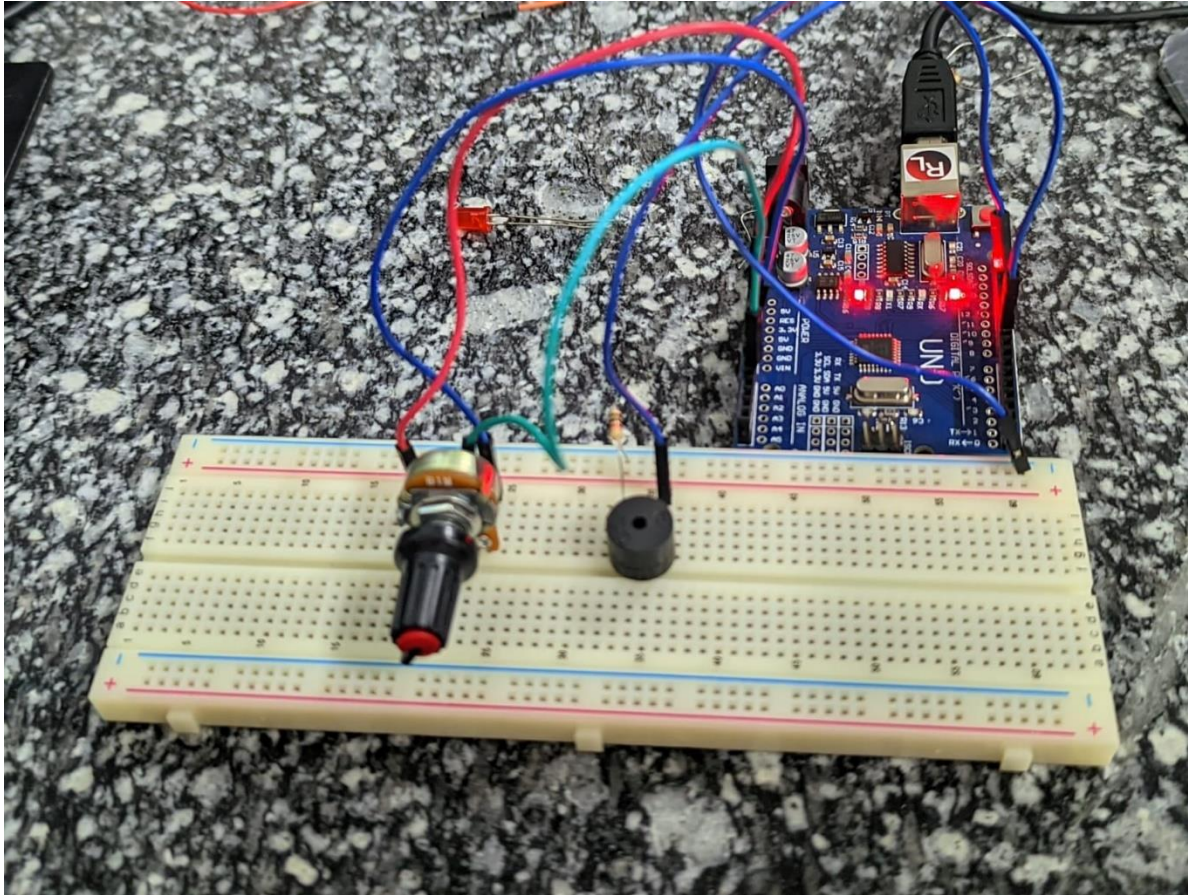
- O terminal central do potenciômetro se conecta ao pino analógico A0 do Arduino.
- Um dos terminais laterais do potenciômetro se conecta ao VCC (5V).
- O outro terminal lateral do potenciômetro se conecta ao GND.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- **#include <LiquidCrystal.h>**: Inclui uma biblioteca para controlar o LCD (não aplicável aqui, mas seria caso fosse usado).
- **const int buzzerPin = 8;**: Defina o pino onde a campainha está conectada.
- **const int potPin = A0;**: Defina o pino onde o potenciômetro está conectado.
- **analogRead(potPin);**: Lê o valor do potenciômetro (0 a 1023).
- **map(potValue, 0, 1023, 100, 1000);**: Mapeia o valor lido do potenciômetro para uma faixa de frequência (100 a 1000 Hz).
- **tone(buzzerPin, frequency);**: Emite um tom no buzzer com a frequência ajustada pelo potenciômetro.
- **delay(50);**: Aguarde 50 milissegundos antes de repetir a leitura.

Resultados Esperados: o buzzer deve emitir um som cuja frequência varia de acordo com a posição do potenciômetro, permitindo ajustar o “volume” ou o tom.



EXERCÍCIO 17

Objetivo: acender um LED quando a luz ambiente estiver abaixo de um determinado nível, utilizando um sensor de luz (LDR).

Materiais Necessários:

- 1 Arduino Uno
- 1 LDR (Resistor Dependente de Luz)
- 1 LED
- 1 resistor (220 Ω para o LED e 10k Ω para o LDR)
- Fios de conexão
- Protoboard

Montagem do Circuito:

LIDERADO:

- O terminal longo (ânodo) do LED se conecta a um pino digital (ex: pino 9) do Arduino.
- O terminal curto (cátodo) se conecta ao GND através de um resistor de 220 Ω .

Esquerda Longa:

- Conecte um terminal do LDR a 5V do Arduino.

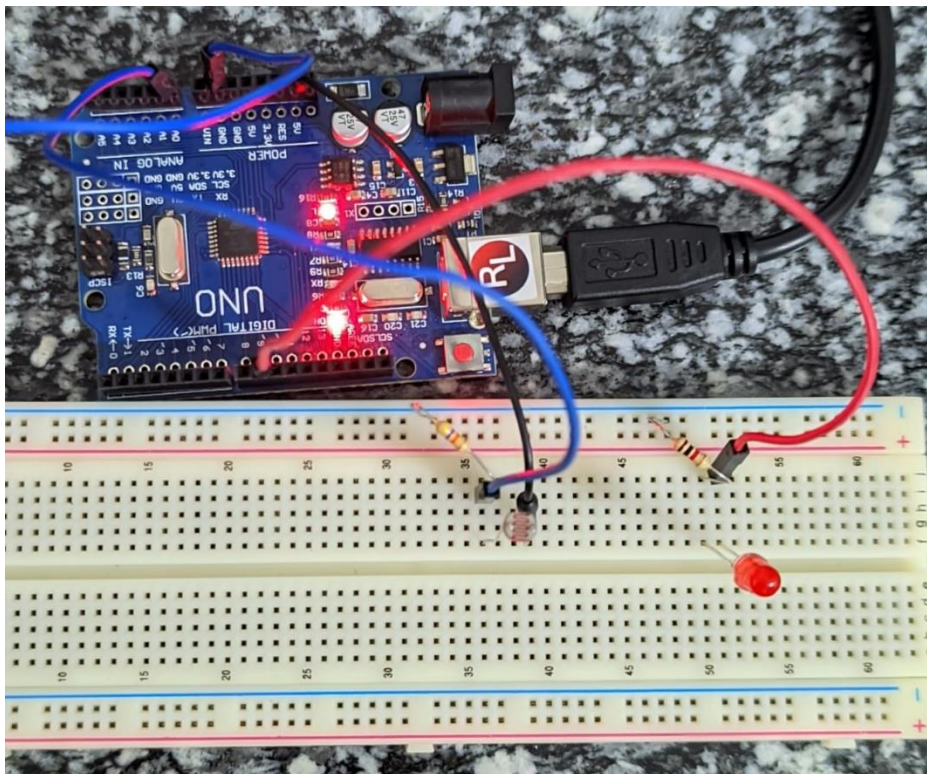
- O outro terminal do LDR se conecta a um pino analógico (ex: A0) e também ao GND através de um resistor de 10k Ω (configuração de divisor de tensão).

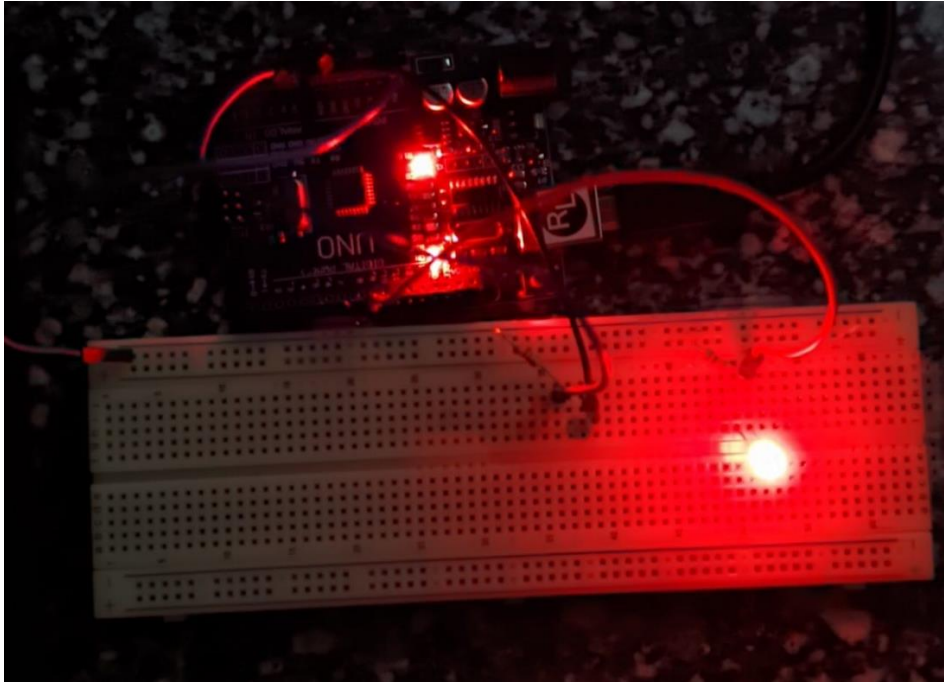
Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- `const int ldrPin = A0;` Define o pino analógico onde o LDR está conectado.
- `const int ledPin = 9;` Define o pino digital onde o LED está conectado.
- `ldrValue = analogRead(ldrPin);` Lê o valor do sensor de luz.
- O código verifica se o valor do LDR está abaixo de um determinado nível (500 neste exemplo) e acenda o LED se a luz estiver fraca.

Resultados Esperados: o LED deve acender quando a luz ambiente estiver baixa e apagar quando a luz estiver acima do nível definido.





EXERCÍCIO 18

Objetivo: controlar a cor do LED RGB usando um potenciômetro, mudando a cor de acordo com a posição do potenciômetro.

Materiais Necessários:

- Arduino Uno
- LED RGB (anodo comum ou gato comum)
- 3 resistores (220Ω)
- Potenciômetro
- Protoboard
- Fios de conexão

Montagem do Circuito:

LED RGB:

- Se o LED para **cato comum** , conecte o pino comum ao **GND** . Se for **anodo comum** , conecte o pino comum a **5V** .
- Conecte o pino do **vermelho** (R) ao pino **9** do Arduino.
- Conecte o pino do **verde** (G) ao pino **10** do Arduino.
- Conecte o pino do **azul** (B) ao pino **11** do Arduino.
- Utilize resistores de **220Ω** para cada pino do LED.

Potenciômetro:

- Conecte as extremidades do potenciômetro ao **5V** e **GND** .
- Conecte o pino central (wiper) ao pino **A0** do Arduino.

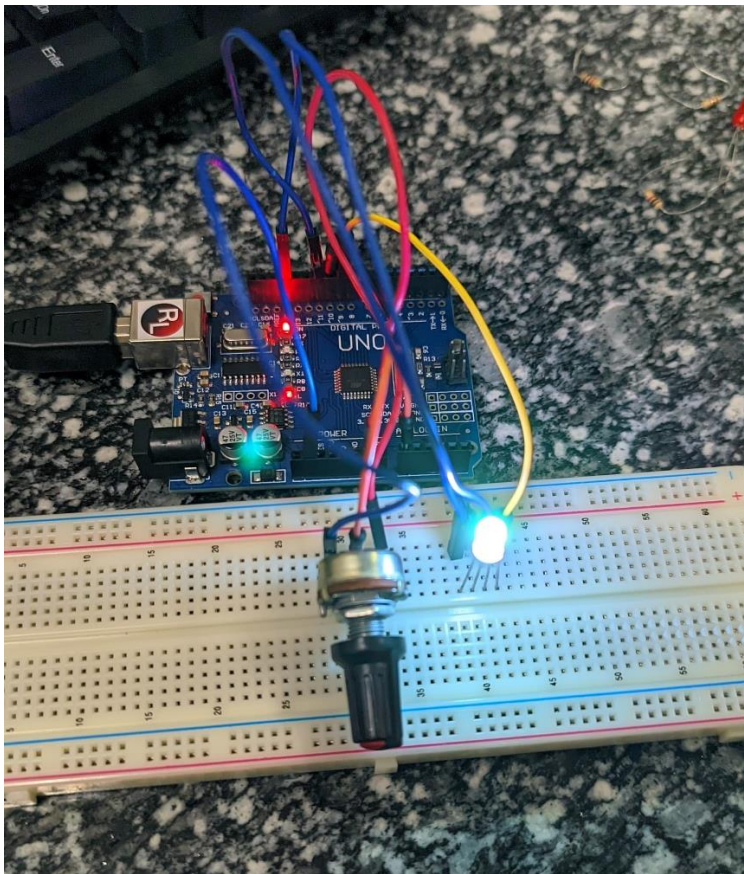
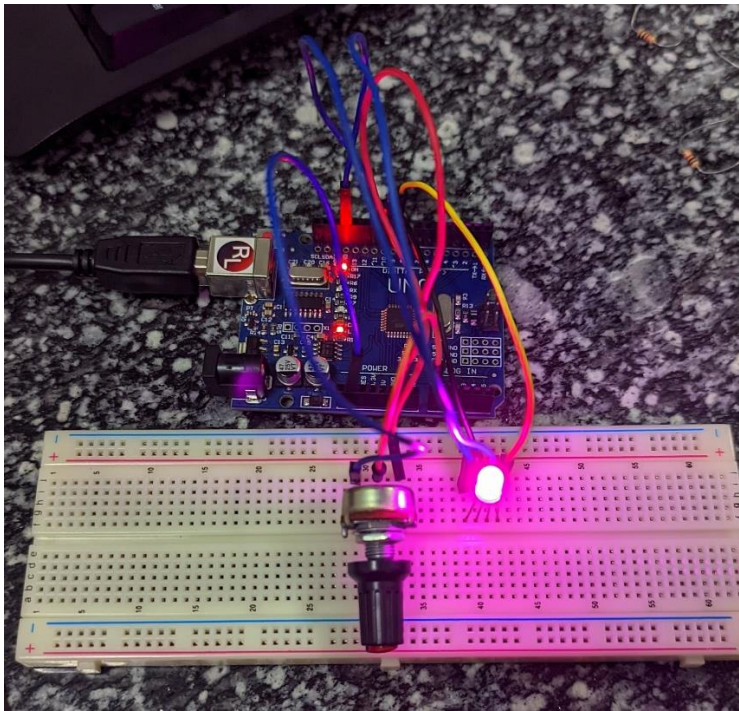
Código para o Arduino:

Explicação do Código:

- **analogRead(A0)** : O valor do potenciômetro e ajusta entre 0 e 1023.

- **map()** : Mapeia os valores do potenciômetro para controlar a intensidade de cada cor (RGB) no intervalo de 0 a 255.
- **analogWrite()** : Define a intensidade de cada canal do LED RGB de acordo com o valor lido do potenciômetro.

Resultados Esperados: o LED RGB mudará de cor de acordo com a posição do potenciômetro, variando entre os núcleos vermelho, verde e azul. Quanto mais você girar o potenciômetro, mais a cor mudará.



Objetivo: mudar a cor do LED RGB em loop e controlar a velocidade com o potenciômetro.

Materiais Necessários:

- 1 Arduino Uno
- 1 LED RGB comum (ou LED RGB com pinos separados)
- 1 Potenciômetro (10kΩ)
- Fios de conexão
- Protoboard

Montagem do Circuito:

LED RGB:

- Conecte o terminal ânodo comum do LED RGB ao VCC (5V).
- O pino do vermelho do LED RGB deve estar conectado ao pino PWM 9 do Arduino.
- O pino do verde do LED RGB deve estar conectado ao pino PWM 10 do Arduino.
- O pino do azul do LED RGB deve estar conectado ao pino PWM 11 do Arduino.
- Os terminais cátodos (separados) devem ser conectados ao GND através de resistores de 220Ω.

Potenciômetro:

- Um terminal do potenciômetro deve estar conectado ao VCC (5V).
- O outro terminal deve estar conectado ao GND.
- O pino central (wiper) do potenciômetro deve estar conectado ao pino analógico A0 do Arduino.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código

Objetivo: Mudar a cor do LED RGB em loop e controlar a velocidade com um potenciômetro.

Definição de pinos:

- redPin, greenPin, bluePin: Pinos de LED RGB.
- potPin: Pino do potenciômetro.

Configurar:

Configure os pinos do LED RGB como saídas.

Laço:

- Lê o valor do potenciômetro e mapeia para um tempo de atraso entre 10 ms e 200 ms.

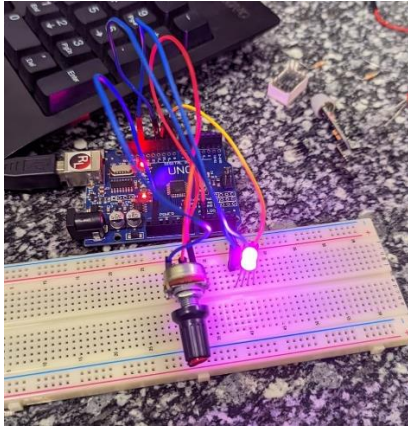
Transições de cores:

- Vermelho para Verde: Aumenta o vermelho, o diminui verde.
- Verde para Azul: Aumenta o azul, o diminui verde.
- Azul para Vermelho: Aumenta o vermelho, diminui o azul.

Função setColor():

- Defina a intensidade dos núcleos do LED RGB usando analogWrite().

Resultados Esperados: o LED RGB deve mudar de cor em um loop contínuo, e a velocidade dessa transição deve ser ajustável com o potenciômetro. Quanto maior o valor do potenciômetro, mais lenta será uma transição de núcleos.

**EXERCÍCIO 20**

Objetivo: usar o leitor de biometria para capturar e armazenar amostras digitais.

Materiais Necessários:

- 1 Arduino Uno
- 1 Módulo de Leitor de Impressão Digital (ex: GT-511C3 ou similar)
- Fios de conexão
- Protoboard (opcional)

Montagem do Circuito:**Conexões do Módulo de Impressão Digital:**

- **GND** do módulo → **GND** no Arduino
- **RX** do módulo → **Pino 2** no Arduino (Recepção)
- **TX** do módulo → **Pino 3** no Arduino (Transmissão)
- **3V3** do módulo → **5V** no Arduino (ou **3.3V** se o módulo não aceita 5V)

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:**Inclusão de Bibliotecas:**

- `#include <Adafruit_Fingerprint.h>`: Importe a biblioteca necessária para controlar o módulo de biometria.
- `#include <SoftwareSerial.h>`: Permite a comunicação serial em pinos digitais, útil para módulos que não utilizam os pinos de serial padrão.

Configuração dos Pinos:

- `SoftwareSerial mySerial(2, 3);`: Defina os pinos 2 e 3 como RX e TX para comunicação com o leitor de impressão digital.

- `Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);`:: Inicializa o objeto `finger` que representará o leitor.

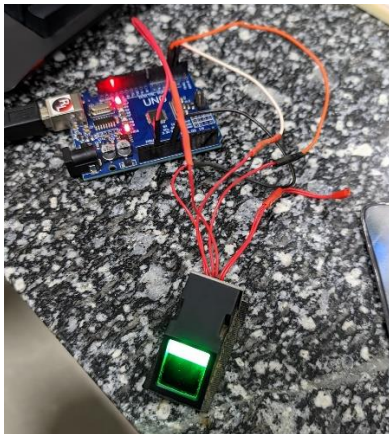
Função `setup()`:

- `Serial.begin(57600);`:: Inicia a comunicação serial para monitoramento.
- `mySerial.begin(57600);`:: Inicia a comunicação com o módulo de impressão digital.
- A função verifica se o leitor de flores digitais foi encontrada e informada através do monitor serial. Se não for encontrado, o programa para.

Função `loop()`:

- Chama a função `getFingerprintID()` para capturar e processar a impressão digital.
- Se um ID válido for retornado, ele é impresso no monitor serial.
- **Função `getFingerprintID()`:**
- `finger.getImage()`: Captura a imagem da impressão digital.
- Se a captura for bem-sucedida, a imagem é convertida em um template com `finger.image2Tz()`.
- Em seguida, a função busca a impressão na base de dados usando `finger.fingerFastSearch()`.
- Se a impressão for encontrada, retorna o ID da impressão digital. Se houver erro em qualquer etapa, exibe uma mensagem de erro e retorna -1.

Resultados Esperados: o usuário deve ser capaz de capturar e armazenar rios digitais no módulo, e o feedback será exibido na comunicação serial.



EXERCÍCIO 21

Objetivo: acender um LED se a impressão digital capturada pelo leitor de biometria para confiança.

Materiais Necessários:

- Arduino Uno
- Leitor de biometria (ex.: módulo de impressão digital)
- LIDERADO
- Resistor (220Ω para LED)
- Fios de conexão
- Protoboard (opcional)

Montagem do Circuito:**Leitor de Biometria:**

- GND -> GND do Arduino
- RX -> pino digital 2 do Arduino
- TX -> pino digital 3 do Arduino
- 3V3 -> 3.3V do Arduino (ou 5V, dependendo do modelo do leitor)

LIDERADO:

- O ânodo (perna longo) do LED deve estar conectado a um pino digital (por exemplo, pino 10).
- O cátodo (perna curta) deve ser conectado ao GND através de um resistor de 220Ω.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- O código inicializa o leitor de biometria e espera que uma impressão digital seja capturada.
- Se a impressão digital for confiável, o LED aceso, reduza o acesso permitido. Caso contrário, uma mensagem é exibida no monitor serial.

Resultado Esperado:

O LED acende quando uma impressão digital válida é lida e reconhecida. Se não for confiável, o LED permanece apagado e uma mensagem de falha é exibida.

EXERCÍCIO 22

Objetivo: Tocar um alarme (buzzer) se o sensor ultrassônico detectar movimento e a impressão digital não for reconhecida.

Materiais Necessários:

- Arduino Uno
- Sensor ultrassônico (HC-SR04)
- Leitor de biometria (ex.: módulo de impressão digital)
- Buzzer
- Fios de conexão
- Protoboard (opcional)

Montagem do Circuito:**Sensor Ultrassônico:**

- VCC -> 5V do Arduino
- GND -> GND do Arduino
- Trig -> pino digital 9 do Arduino
- Echo -> pino digital 10 do Arduino

Leitor de Biometria:

- GND -> GND do Arduino
- RX -> pino digital 2 do Arduino
- TX -> pino digital 3 do Arduino
- 3V3 -> 3.3V do Arduino (ou 5V, dependendo do modelo do leitor)

Buzzer:

- Um terminal do buzzer deve estar conectado ao pino digital 8 do Arduino.
- O outro terminal deve estar conectado ao GND.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- O código utiliza um sensor ultrassônico para medir a distância. Se a distância for menor que 20 cm, ele tenta ler uma impressão digital.
- Se a impressão digital não for reconhecida, o buzzer toca, indica que um movimento foi detectado sem autenticação.

Resultado Esperado: o buzzer toca se um movimento é detectado a menos de 20 cm e a impressão digital não é reconhecida. Se a impressão digital for confiável, o alarme é desativado.

EXERCÍCIO 23

Objetivo: mostrar o valor lido pelo potenciômetro em um display de 7 segmentos.

Materiais Necessários:

- Arduino Uno
- Potenciômetro
- Exibição de 7 segmentos
- Resistores (220Ω para display)
- Fios de conexão
- Protoboard (opcional)

Montagem do Circuito:**Potenciômetro:**

- Um terminal ao VCC (5V)
- Outro terminal ao GND
- O terminal do meio ao pino analógico A0 do Arduino.

Exibição de 7 segmentos:

- Um (pino 2)
- B (pino 3)

- C (pino 4)
- D (pino 5)
- E (pino 6)
- F (pino 7)
- Sol (pino 8)
- DP (pino 9, se necessário)
- Conecte os pinos correspondentes a um resistor de 220Ω ao GND.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- O código lê o valor do potenciômetro e o mapeia para um número entre 0 e 9, que será exibido no display de 7 segmentos.
- A função `displayDigit`` acende os segmentos corretos do display para mostrar o dígito correspondente.

Resultado esperado: O display de 7 segmentos mostra o valor atual do potenciômetro, variando de 0 a 9 conforme o potenciômetro é ajustado.

EXERCÍCIO 24

Objetivo: criar um sistema de autenticação que utilize um leitor biométrico e um botão para permitir ou negar o acesso.

Materiais Necessários:

- Arduino Uno
- Leitor biométrico
- Botão

LIDERADO

- Resistor de 220Ω
- Fios de conexão

Esquema de Ligação:

- Conecte o leitor biométrico aos pinos RX e TX do Arduino.
- Conecte o LED a um pino digital.
- Conecte o botão a outro pino digital.

Código para o Arduino: disponível no site The World of Arduino

Explicação do Código:

- **Biblioteca Biométrica:** Importe a biblioteca necessária para trabalhar com o leitor biométrico.
- **Definições de pinos:** Define o pino do leitor biométrico, do LED e do botão.

- **Setup:** Inicializa a comunicação serial, configura os pinos como saída (LED) ou entrada (botão) e inicia o leitor biométrico.
- **Loop:** Verifique se a biometria é reconhecida. Se sim, acender o LED; caso contrário, apaga. O botão, quando pressionado, limpa a biometria.

Resultados Esperados:

- Quando a biometria é reconhecida, o LED acende, diminuindo o acesso concedido.
- Quando não reconhecido, o LED permanece apagado.
- O botão pode ser usado para limpar a biometria armazenada.

EXERCÍCIO 25

Objetivo: criar um sistema de semáforo que utilize um sensor ultrassônico para detectar proximidade e acender LEDs conforme a distância.

Materiais Necessários:

- Arduino Uno
- Sensor ultrassônico
- 3 LEDs (vermelho, amarelo e verde)
- Resistores de 220Ω
- Fios de conexão

Esquema de Ligação:

- Conecte o sensor ultrassônico aos pinos trig e echo do Arduino.
- Conecte os LEDs aos pinos digitais.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- **Definições de pinos:** Define os pinos para o sensor ultrassônico e LEDs.
- **Setup:** Inicializa a comunicação serial e configura os pinos como entrada ou saída.
- **Loop:** Envia um pulso pelo pino trig e lê o tempo que leva para receber o eco no pino echo. Calcule a distância e acenda o LED correspondente à distância bloqueada.

Resultados Esperados:

- LED vermelho acende se a distância for menor que 20 cm.
- O LED amarelo acende se a distância estiver entre 20 e 50 cm.
- O LED verde acende se a distância for maior que 50 cm.

EXERCÍCIO 26

Objetivo: criar um circuito que exiba uma contagem automática em um display de 7 segmentos.

Materiais Necessários:

- Arduino Uno
- Exibição de 7 segmentos
- Resistores de 220Ω
- Fios de conexão

Esquema de Ligação:

- Conecte os pinos do display de 7 segmentos aos pinos digitais do Arduino.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- **Definições de pinos:** Define os pinos do display de 7 segmentos.
- **Setup:** Configure todos os pinos como saída.
- **Loop:** Conta de 0 a 9, chama a função `displayDigit` para acender os segmentos correspondentes de cada número, com um atraso de 1 segundo.
- **displayDigit:** Acende os segmentos necessários para mostrar o número correspondente.

Resultados Esperados: o display de 7 segmentos deve mostrar uma contagem de 0 a 9, mudando a cada segundo.

EXERCÍCIO 27

Objetivo: criar um sistema que monitore a temperatura e acione um ventilador se a temperatura exceder um limite.

Materiais Necessários:

- Arduino Uno
- Sensor de temperatura (LM35)
- Ventilador (ou motor)
- Transistor
- Resistores
- Fios de conexão

Esquema de Ligação:

- Conecte o sensor de temperatura ao pino analógico do Arduino.
- Conecte o transistor ao pino digital para controlar o ventilador.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- **Definições de pinos:** Define os pinos para o sensor de temperatura e ventilador.
- **Setup:** Inicializa a comunicação serial e configura o pino do ventilador como saída.
- **Loop:** Lê o valor do sensor, converte para Celsius e exibe no monitor serial. Se a temperatura ultrapassar 30°C, o ventilador é acionado.

Resultados Esperados:

- O monitor serial exibe a temperatura em tempo real.
- O ventilador é acionado se a temperatura ultrapassar 30°C e desliga quando a temperatura estiver abaixo desse valor.

EXERCÍCIO 28

Objetivo: controlar a posição de um servo motor utilizando um potenciômetro.

Materiais Necessários:

- Arduino Uno
- Potenciômetro
- Servomotor
- Fios de conexão

Esquema de Ligação:

- Conecte o potenciômetro a um pino analógico.
- Conecte o servo motor ao pino digital.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- **Biblioteca Servo:** Importe uma biblioteca para controlar servos.
- **Setup:** Anexa o servo a um pino digital.
- **Loop:** Lê o valor do potenciômetro e a mapeia para a faixa de movimento do servo (0 a 180 graus). Em seguida, defina a posição do servo.

Resultados Esperados: o servo motor se moverá para a posição correspondente ao valor do potenciômetro, permitindo um controle contínuo da posição.

EXERCÍCIO 29

Objetivo: controlar o tom de um buzzer utilizando um potenciômetro.

Materiais Necessários:

- Arduino Uno
- Potenciômetro
- Buzzer
- Fios de conexão

Esquema de Ligação:

- Conecte o potenciômetro ao pino analógico do Arduino.
- Conecte a campainha a um pino digital.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- **Definições de pinos:** Define os pinos do potenciômetro e buzzer.
- **Setup:** Configure o pino do buzzer como saída.
- **Loop:** Lê o valor do potenciômetro, o mapa para uma faixa de frequência (100 a 1000 Hz) e toca essa frequência no buzzer.

Resultados Esperados: o tom do buzzer muda conforme o valor do potenciômetro é ajustado.

EXERCÍCIO 30

Objetivo: Criar um sistema de medição de distância que permita ajustar a precisão da medição.

Materiais Necessários:

- Arduino Uno
- Sensor ultrassônico
- Potenciômetro
- Fios de conexão

Esquema de Ligação:

- Conecte o sensor ultrassônico aos pinos trig e echo do Arduino.
- Conecte o potenciômetro ao pino analógico.

Código para o Arduino: disponível no site The World of Arduino.

Explicação do Código:

- **Definições de pinos:** Defina os pinos para o sensor ultrassônico e potenciômetro.
- **Setup:** Inicializa a comunicação serial e configura os pinos.

ANEXO

Imagem 1: Um dos autores aplicando a aula de número 2 do guia.



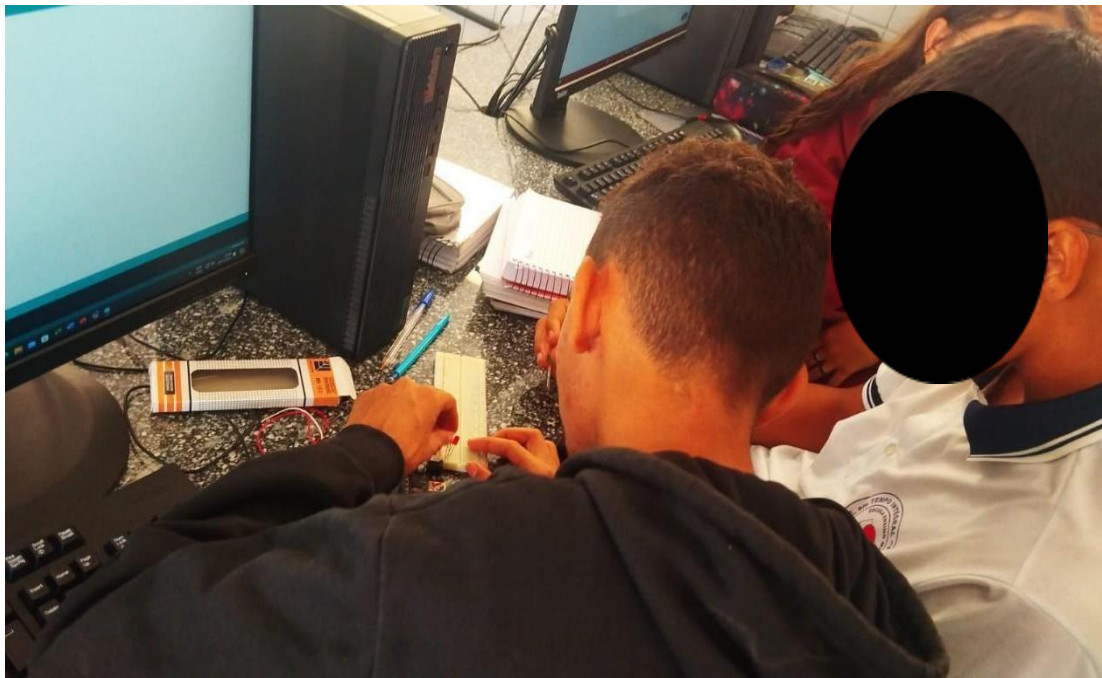
Fonte: Arquivo pessoal.

Imagem 2: Aluno participando da atividade prática proposta pelo guia.



Fonte: Arquivo pessoal.

Imagem 3: Alunos participando da atividade prática proposta pelo guia.



Fonte: Arquivo pessoal.

Imagem 4: Alunos participando da atividade prática proposta pelo guia, com um dos autores supervisionando.



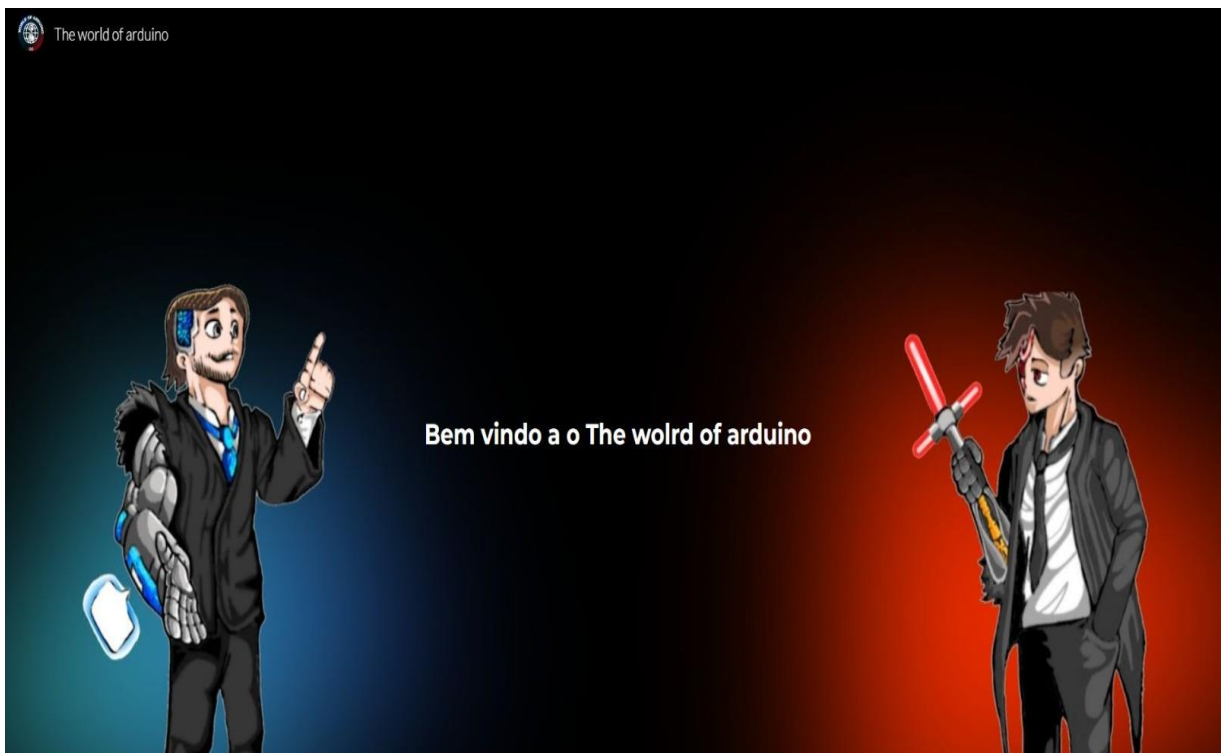
Fonte: Arquivo pessoal.

Imagem 5: Alunos realizando a prova diagnóstica.



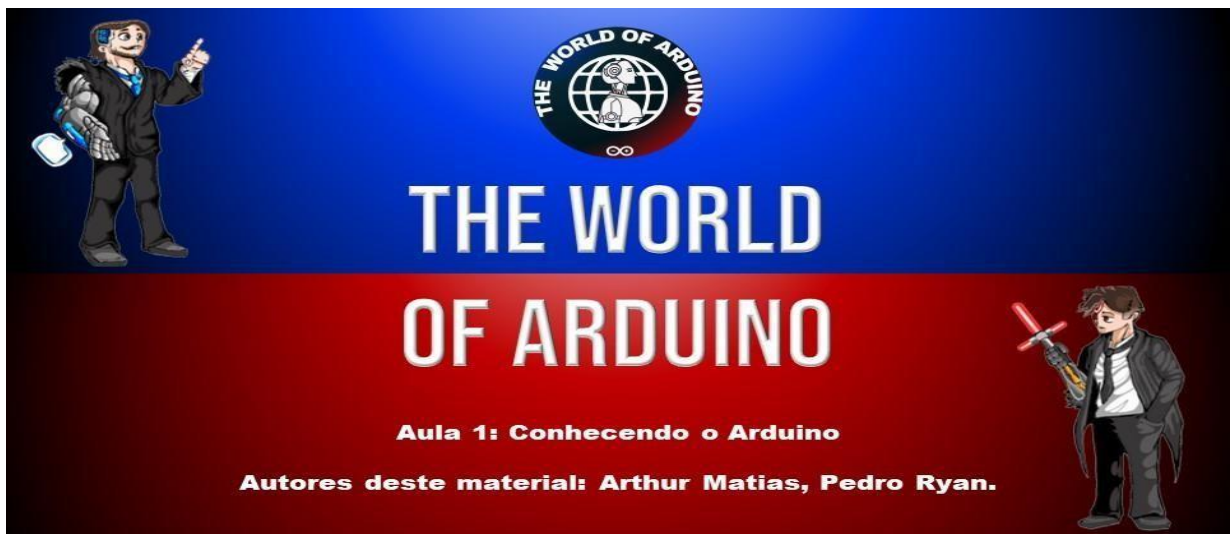
Fonte: Arquivo pessoal.

Imagem 6: Interface inicial do nosso site



Fonte: Arquivo pessoal

Imagem 7: Slide inicial de uma das aulas.



Fonte: Arquivo pessoal

