LUZES, CÓDIGO E AÇÃO: PRIMEIROS PASSOS COM ARDUINO

Paulo Ricardo Mello Ataide de Oliveira[1], Jeohvana Vaz[2], Maria Eduarda Jardim Tavares[3], Sofia Magalhães Pires[4], Marlipe Garcia Fagundes Neto [5]

[1]Estudante de Engenharia Elétrica da Universidade Federal de Goiás, Aparecida de Goiânia, Brasil(paulo.ricardo2@discente.ufg.br);
[2]Estudante de Engenharia Elétrica da Universidade Federal de Goiás, Porungatu, Brasil(jeohvana@discente.ufg.br);
[3]Estudante de Engenharia Elétrica da Universidade Federal de Goiás, Aparecida de Goiânia, Brasil(maria_eduarda23456@discente.ufg.br);
[4]Estudante de Engenharia Elétrica da Universidade Federal de Goiás, Aparecida de Goiânia, Brasil(sofiamagalhaes@discente.ufg.br);
[5]Professor Doutor em Engenharia Mecânica da Universidade Federal de Goiás e tutor do PET Engenharias, Goiânia, Brasil(marlipe.ufg.br).

Resumo: Este trabalho relata a oficina de introdução ao Arduino realizada pelo PET Engenharias da UFG no ENAPET 2025. A atividade de caráter prático, abordou conceitos de microcontroladores e a execução de três projetos progressivos com hardware físico. O objetivo foi desmistificar a eletrônica e a programação para estudantes de diversas áreas, promovendo o protagonismo petiano e o desenvolvimento de competências tecnológicas transversais.

Palavras-chave: Arduino; PET Engenharias; ENAPET; Microcontroladores; Oficina.

INTRODUÇÃO

Em um contexto acadêmico e profissional cada vez mais digital, a habilidade de entender e interagir com tecnologias de programação e eletrônica passou a ser uma competência relevante para estudantes de todas as áreas. A formação de profissionais capazes de desenvolver soluções inovadoras depende do aprimoramento de habilidades como raciocínio lógico, resolução de problemas e capacidade de projetar sistemas, atributos nos quais são desenvolvidos através do pensamento computacional (Wing, 2006).

Nesse sentido, a plataforma Arduino se destaca como uma ferramenta educacional eficaz, sendo amplamente reconhecida por seu custo reduzido e apresenta uma tecnologia simples de ser utilizada para criar projetos baseados em microcontroladores (Monk, 2013).

O Programa de Educação Tutorial (PET), vinculado à Secretaria de Educação Superior do MEC, tem como premissa o desenvolvimento de atividades de ensino, pesquisa e extensão de forma integrada. Formado por grupos tutoriais de aprendizagem, o PET proporciona aos alunos participantes a realização de atividades extracurriculares com o intuito de complementar a formação acadêmica dos estudantes (UFG, 2026). O PET Engenharias da Universidade Federal de Goiás, que possui a peculiaridade de ser formado pelos estudantes dos cursos de Engenharia Elétrica, Engenharia Mecânica e Engenharia de Computação (UFG, 2026), atua nessa perspectiva, promovendo ações que articulam o conhecimento técnico à formação cidadã e interdisciplinar dos seus membros e da comunidade ao redor.

O Encontro Nacional dos Grupos PET (ENAPET) é o principal evento que reúne petianos (membros do PET) de todo o país, constituindo-se como um espaço de troca de experiências, apresentação de atividades e fortalecimento da identidade petiana. A edição de 2025 foi sediada pela

Universidade de Brasília (UnB), nos dias 21, 22 e 23 de novembro de 2025, e reuniu grupos PET de diversas instituições superiores (Encontro Nacional dos Grupos, 2026) .

Em consonância com esse contexto, o PET Engenharias da UFG propôs e executou uma oficina de introdução da plataforma Arduino durante o ENAPET 2025, com o objetivo de proporcionar aos petianos de diferentes áreas do conhecimento uma experiência prática com microcontroladores. Este trabalho tem como objetivo desmistificar a tecnologia, mostrando a execução e os resultados obtidos na oficina de Arduino no ENAPET 2025, no qual foi capaz de oferecer uma base em eletrônica básica, programação embarcada e reforçar o caráter interdisciplinar e inovador do Programa de Educação Tutorial para alunos de graduação de diferentes áreas do conhecimento.

MATERIAL E MÉTODOS

A oficina de Arduino "**LUZES, CÓDIGO E AÇÃO: PRIMEIROS PASSOS COM ARDUINO**" foi planejada e conduzida por quatro bolsistas do PET Engenharias da UFG e contou com a participação de 9 petianos (alunos da oficina) de diferentes grupos e instituições, nos quais foram divididos em 3 grupos de 3 pessoas, além dos dois petianos da comissão de organização do XXX ENAPET. O evento ocorreu de forma presencial no espaço reservado para atividades durante o ENAPET 2025.

Para a execução das atividades práticas, foram utilizados os seguintes materiais: 5 placas Arduino Uno, 4 *protoboards*, LEDs, resistores de 1 $k\Omega$, botões (*push buttons*), transistores MOSFET, *jumpers*, motores 130 de 6 V e pilhas AA de 1,5 V. A programação dos arduinos foi realizada por meio do *software* Arduino IDE, e foram utilizados os computadores dos ministrantes e dos participantes para gravar os códigos nos Arduinos.

A metodologia adotou uma abordagem progressiva, estruturada em três projetos de complexidade crescente, precedidos por uma apresentação teórica introdutória. O conteúdo teórico abordou o conceito de microcontrolador, o funcionamento da *protoboard* e do transistor tipo MOSFET e uma introdução básica ao Arduino IDE.

O primeiro projeto consistiu na montagem do circuito base com um LED externo, introduzindo os participantes ao funcionamento da *protoboard*, ao uso de resistores para proteção do LED e à estrutura básica de um código Arduino com as funções *setup()* e *loop()*. Nessa etapa, os participantes realizaram o projeto *Blink*, acionando o LED para piscar em intervalos regulares de um segundo.

Como mostrado na Figura 1, o objetivo deste circuito é ligar e desligar o LED em *loop*, e para isso o terminal negativo do LED foi conectado ao pino GND do Arduino Uno através de *jumpers* e a placa *protoboard*, e o terminal positivo do LED deve ser ligado ao pino 13 do Arduino Uno, pino no qual será habilitado para liberar uma tensão de 5v para acender o LED. Além disso é necessário adicionar um resistor de aproximadamente 1 $k\Omega$ no terminal positivo do LED, entre o LED e o *jumper* conectado ao pino do Arduino, como mostrado na Figura 1.

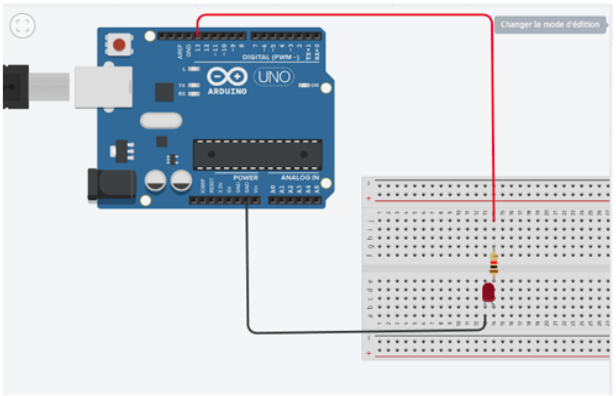


Figura 1. Esquema eletrônico do Projeto 1.

A Figura 2 apresenta o código utilizado no Projeto 1. Este código foi escrito na linguagem do Arduino baseada em C/C + + utilizando o *software* Arduino IDE. Em códigos Arduino, a configuração dos pinos é realizada na função *setup()*, que é utilizada para configurar os pinos de entrada e de saída do Arduino, conforme destacado por McRoberts (2011). Os pinos de entrada, são aqueles nos quais o Arduino recebe um sinal externo, e a partir deste sinal ele executa uma ação, e estes pinos são configurados como:

pinMode(pin, INPUT);

Enquanto que os pinos de saída do Arduino são configurados para enviar um sinal do

microcontrolador para algum componente externo, como no acionamento de um LED (McRoberts 2011). Os pinos de saída do Arduino são configurados como:

pinMode(pino, OUTPUT);

No código apresentado na Figura 2, o pino 13 foi nomeado como *LED_BUILTIN* (que é a forma padrão do código *Blink* que é um exemplo pronto do Arduino IDE), e foi configurado como saída na função *setup()*. Os códigos escritos dentro das chaves da função *setup()* são executados apenas uma vez (Monk, 2013). Após a função *setup()* encontra-se a função *loop()*, onde realmente é escrito as ações que devem ser executadas pelo Arduino, e os códigos dentro de *loop()* são executados repetidamente (McRoberts, 2011).

Para o código da figura 2, utilizou-se o comando *digitalWrite(LED_BUILTIN, HIGH)*, este comando serve para o Arduino acionar algum pino no nível lógico alto (Monk, 2013), que nesse caso é o *LED_BUILTIN* (pino 13) ligando o LED. Em seguida, foi implementado o *delay (1000)*, que servirá para que o Arduino não execute nenhuma ação durante 1 segundo (1000 ms), fazendo com que o LED permaneça aceso durante este tempo.

O código apresenta o comando *digitalWrite(LED_BUILTIN, LOW),* que vai habilitar o pino selecionado no nível lógico baixo (Monk, 2017), que neste exemplo é o *LED_BUILTIN,* desligando assim o LED.

Na próxima linha de código ocorre mais um *delay* de 1 segundo, para o LED permanecer desligado, e como esta é a última linha de código da função *loop()*, logo este bloco de código irá se repetir de novo, fazendo com que o LED ligue e apague a cada 1 segundo.

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

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

Figura 2. Código do projeto 1 (*Blink*).

O segundo projeto substituiu o LED por um motor controlado por um transistor MOSFET, demonstrando o uso do transistor como chave eletrônica e a possibilidade de controlar atuadores de maior potência por meio de uma saída digital do Arduino. O código utilizado foi o mesmo do projeto 1 (com a exceção de que o pino 13 foi substituído pelo pino 9 do Arduino Uno), no qual o motor foi acionado de forma autônoma, alternando entre ligado e desligado em intervalos programados de 1 segundo. Entretanto, ao longo da oficina foram feitos testes de intervalos maiores e menores para o acionamento do motor.

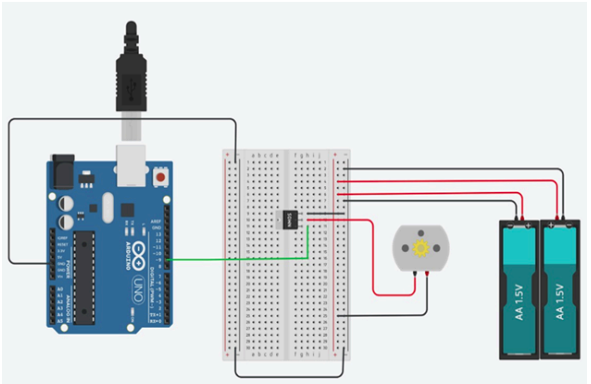


Figura 3. Esquema eletrônico do Projeto 2.

Conforme apresentado na figura 3, o circuito é semelhante com o circuito do LED, porém o LED foi substituído por um motor dc, e ocorreu a adição de um *Metal-Oxide-Semiconductor Field Effect Transistor,* (MOSFET) em português, Transistor de Efeito de Campo de Óxido de Metal. O MOSFET é um componente eletrônico que funciona como amplificador para pequenos sinais, ou uma chave para circuitos de corrente contínua (Sedra e Smith, 2004).

O MOSFET possui 3 terminais chamados de dreno, porta e fonte, em que ao aplicar uma tensão no terminal da porta, o MOSFET funciona como chave fechada, acionando o componente conectado ao transistor, como um motor, mas quando é interrompida a alimentação do terminal porta do

MOSFET, o transistor funciona como uma chave aberta, interrompendo a alimentação do motor.

Como mostrado no circuito da Figura 3, o terminal porta do MOSFET está conectado ao pino 9 e a cada 1 segundo este pino será ativado, e enviará 5V para o MOSFET, acionando o motor , e após passado 1 segundo, o pino será desativado, fazendo com que o eixo do motor pare de girar, e este *loop* se repete indefinidamente.

O Projeto 3 incorporou ao circuito anterior um botão como elemento de entrada digital. Os participantes aprenderam a utilizar portas digitais como entrada, configurar o resistor de pull-up interno do Arduino e implementar uma lógica condicional para acionar o motor somente quando o botão fosse pressionado.

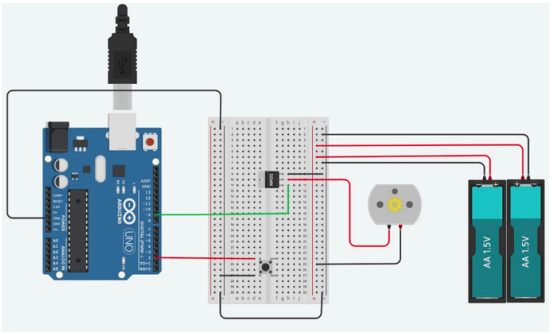


Figura 4. Esquema eletrônico do Projeto 3.

```
// Declaração dos pinos
const int MOTEpin = 9;   // LED / motor / dispositivo de saída
const int butpin  = 2;   // Botão com resistor pull-up interno
int state;

void setup() {
  pinMode(MOTEpin, OUTPUT);
  pinMode(butpin, INPUT_PULLUP); // Mantém o pino em HIGH quando
}

void loop() {
  // Leitura do botão
  state = digitalRead(butpin);

  // Lógica de acionamento:
  // INPUT_PULLUP inverte a lógica:
  // HIGH = não pressionado | LOW = pressionado
  if (state == LOW) {
    digitalWrite(MOTEpin, HIGH);  // Botão pressionado: liga
  } else {
    digitalWrite(MOTEpin, LOW);   // Botão solto: desliga
  }
}
```

Figura 5. Código do projeto 3.

Conforme apresentado na Figura 4, o circuito mantém a estrutura do projeto anterior com o MOSFET e o motor, mas inclui a adição de um botão (*push button*) conectado ao pino 2 do Arduino e ao barramento de GND da *protoboard*. O código utilizado no Projeto 3 (Figura 5) introduz a leitura de sinais externos para a tomada de decisão do microcontrolador. Primeiramente, na função *setup()* da Figura 5, o pino do motor *MOTEpin* (pino 9) é configurado como OUTPUT, enquanto o pino do botão *butpin* (pino 2) é configurado como *INPUT_PULLUP*. Essa configuração de *pull-up* interno é fundamental, pois garante que o pino receba um sinal de 5V (nível lógico alto) quando o botão estiver em repouso, evitando flutuações elétricas que poderiam causar acionamentos indevidos (McRoberts 2011).

Na função *loop()*, o Arduino realiza a leitura constante do estado do botão através do comando *digitalRead(butpin)*. A lógica condicional *if (state == LOW)* é implementada para verificar se o botão foi pressionado. Como o botão está conectado ao GND e configurado com *pull-up*, o pressionamento fecha o circuito e envia um sinal de nível baixo *(LOW)* ao pino 2. Quando essa condição é verdadeira, o comando *digitalWrite(MOTEpin, HIGH)* aciona o terminal porta do MOSFET, que atua como uma chave fechada e permite o acionamento do motor. Caso o botão não esteja pressionado *(else)*, o pino do motor permanece em *LOW*, mantendo o transistor como uma chave aberta e o motor desligado.

Todo o material de apoio utilizado na oficina foi organizado em uma apostila desenvolvida pelos próprios membros do PET Engenharias, disponibilizada aos participantes durante a atividade.

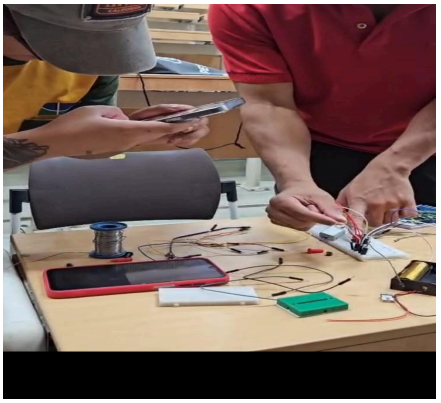


Figura 6. Execução do Projeto 3



Figura 7. Ministrantes da oficina de Arduino do PET Engenharias .

RESULTADOS E DISCUSSÃO

A oficina do PET Engenharias sobre introdução ao Arduino contou com participantes de diversos cursos, e instituições diferentes, como mostrado no formulário de *feedback* abaixo.

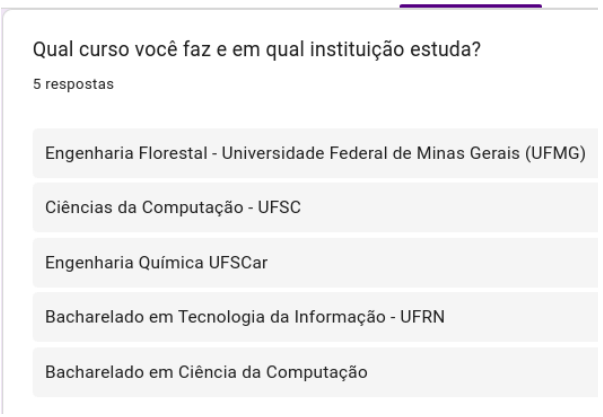


Figura 8. Cursos dos participantes.

De acordo com a Figura 8, os cursos dos participantes na oficina foram: Engenharia Florestal, Ciências da Computação, Engenharia Química e Tecnologia da Informação. Mostrando desta forma que conteúdos ligados à eletrônica e microcontroladores atraem estudantes de outras áreas do conhecimento além da Engenharia Elétrica e Engenharia da Computação.



Figura 9. Opinião dos participantes sobre a dificuldade da oficina.

Como mostrado na Figura 9, 100% dos participantes consideraram o nível da oficina como “adequado”. Desta forma pode-se inferir que os conteúdos apresentados na oficina foram ideais para iniciantes em Arduino, de forma que o conteúdo não foi nem extremamente básico, porém não foi avançado.

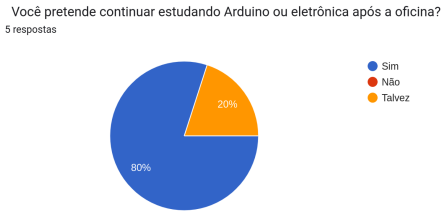


Figura 10. Prosseguimento no estudo de arduino após a oficina.

De acordo com a Figura 10, 80% dos participantes que responderam o formulário de *feedback* afirmaram que irão continuar os estudos de Arduino após a oficina, e 20% dos participantes afirmaram que talvez continuariam. Estes dados mostraram como conteúdos relacionados a microcontroladores e eletrônica básica são conhecimentos que podem ser aplicados de forma útil em diversas áreas do conhecimento diferentes.

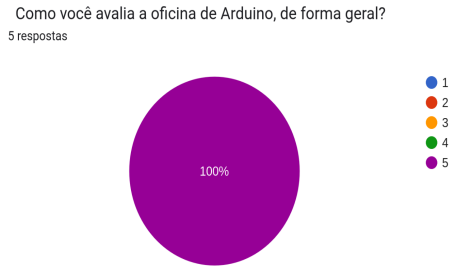


Figura 11. Prosseguimento no estudo de arduino após a oficina

Por fim, como mostrado na Figura 11, em uma escala de 1 a 5, 100% dos participantes que preencheram o formulário de *feedback* da oficina de Arduino, avaliaram a oficina com a nota 5, mostrando uma percepção extremamente positiva do trabalho realizado pelos membros do PET Engenharias.

CONCLUSÃO

Para concluir, como mostrado pelas respostas do formulário de *feedback* dos participantes, o nível de dificuldade da oficina foi adequado, a maior parte dos participantes pretendem continuar os estudos em Arduino e 100% dos que responderam o formulário de *feedback* atribuíram a nota máxima à oficina. Desta forma pode-se concluir que a oficina "**LUZES, CÓDIGO E AÇÃO: PRIMEIROS PASSOS COM ARDUINO**" atendeu os seus objetivos, disseminando assim conhecimentos de eletrônica e microcontroladores para estudantes de diversas áreas do conhecimento diferentes.

AGRADECIMENTOS

Agradecimentos ao MEC e ao Fundo Nacional de Desenvolvimento da Educação (FNDE), que graças ao custeio anual dos grupos PET, o PET Engenharias foi capaz de arcar com os custos da viagem e dos componentes necessários para a realização da oficina de Arduino para o XXX ENAPET. E agradecimentos ao PET Engenharias, por permitir a realização da oficina.

REFERÊNCIAS

ENCONTRO NACIONAL DOS GRUPOS PET, 30., 2025, Brasília. **Anais do XXX ENAPET...** Even3, Brasília, 2025. Disponível em: https://www.even3.com.br/anais/xxx-encontro-nacional-grupos-do-pet/. Acesso em: 29 maio 2026.

MCROBERTS, Michael. **Arduino básico**. (Rubens Prates). Novatec Editora, São Paulo, 2011.

MONK, Simon. **Programação com Arduino: começando com sketches**. (Verônica de Abreu Amaral). Bookman, Porto Alegre, 2013.

SEDRA, Adel S.; SMITH, Kenneth C. **Microelectronic circuits**. Oxford University Press, New York, 2004.

UNIVERSIDADE FEDERAL DE GOIÁS. **O que é o PET**. PROGRAD/UFG, Goiânia, 2026. Disponível em: https://prograd.ufg.br/p/7197-o-que-e-o-pet. Acesso em: 29 maio 2026.

WING, Jeannette M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, mar. 2006. Disponível em: https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf. Acesso em: 29 maio 2026.