

Conectando Teoria e Prática: Uma Abordagem Experimental para o Ensino de Autômatos Finitos Determinísticos com Arduino

Arthur Roberto da Silva¹, Cecília Mesquita¹, Ana Livia Lopes¹,
William M Brito¹, Pedro N Damacena¹, Marcus Nogueira¹,
Maria Eduarda Marques¹, Francisca Ariane da Silva¹,
Maria Eduarda Lima¹, Marcelo Martins¹, Viviane Menezes¹

¹Universidade Federal do Ceará – Campus Quixadá
Av. José de Freitas Queiroz, 5003 – Cedro – Quixadá – Ceará – Brasil

{arthurroberto, cecimesquita, slopeslivia, william04, pedronobrega, marcusnogueira, eduardamarques, francisca.silva, eduarda05}@alu.ufc.br, {mmartins, vivianemenezes}@ufc.br

Abstract. Context: Deterministic Finite Automata (DFAs) concepts are often perceived as abstract by Computer Science students. **Objective:** To report the experience of a short course that integrated Arduino into the teaching of DFAs. **Method:** Conducted in 2024 and 2025, the course combined theoretical lessons with hands-on activities involving sensors, actuators, and microcontrollers. Participants analyzed and implemented automata models, such as traffic lights and automatic doors, using real hardware. Questionnaires were administered to assess students' perceptions. **Results:** The concrete visualization of states and transitions facilitated the understanding of theoretical concepts and increased participant engagement. **Conclusion:** The integration of Arduino and DFAs proved to be an effective strategy for bridging theory and practice, making learning more meaningful and motivating.

Resumo. Contexto: Conceitos de Autômatos Finitos Determinísticos (AFDs) são frequentemente considerados abstratos por estudantes de Computação. **Objetivo:** Relatar a experiência de um minicurso que integrou Arduino ao ensino de AFDs. **Método:** Realizado em 2024 e 2025, o minicurso combinou aulas teóricas e atividades práticas com sensores, atuadores e microcontroladores. Os participantes analisaram e implementaram modelos de autômatos, como semáforos e portas automáticas, em hardware real. Questionários foram aplicados para avaliar a percepção dos estudantes. **Resultados:** A visualização concreta de estados e transições facilitou a compreensão dos conceitos teóricos, além de aumentar o engajamento dos participantes. **Conclusão:** A integração entre Arduino e AFDs mostrou-se uma estratégia eficaz para aproximar teoria e prática, tornando o aprendizado mais significativo e motivador.

1. Introdução

Conteúdos teóricos como os Autômatos Finitos Determinísticos (AFD) desempenham papel fundamental na formação em Computação, pois fornecem a base conceitual para áreas como linguagens formais, compiladores e teoria da computação. Entretanto, seu ensino frequentemente envolve formalismos matemáticos e conceitos abstratos que podem dificultar a compreensão e reduzir o engajamento dos estudantes [Honda et al. 2022, Binsfeld et al. 2009]. Diante desse desafio, diferentes iniciativas têm buscado aproximar a teoria da prática por meio

de abordagens mais interativas e contextualizadas. Exemplos incluem o projeto Automigos [Honda et al. 2023], que utiliza elementos de ludificação espacial para o ensino de autômatos, e a proposta de [Vasconcelos and Guerra 2023], baseada em Jupyter Notebooks, que transforma conceitos teóricos em conteúdos interativos e executáveis.

Nesse contexto, a robótica educacional e plataformas de prototipagem como o Arduino apresentam-se como alternativas promissoras para reduzir a distância entre conceitos abstratos e aplicações concretas. Estudos como os de [Melo and Barbosa 2025, Sangali et al. 2024, Santos and Oliveira 2025] evidenciam o potencial do uso de circuitos eletrônicos e dispositivos programáveis no desenvolvimento do Pensamento Computacional e na aprendizagem de conceitos de computação. Ao possibilitar que estados, transições e entradas sejam representados por componentes físicos, o uso de *hardware* favorece a visualização e a experimentação de conceitos que, tradicionalmente, permanecem restritos ao campo teórico. Sob a perspectiva construcionista de Papert [Papert 1980], tais recursos transformam a teoria em artefatos manipuláveis, permitindo que os estudantes construam conhecimento por meio da exploração e da reflexão sobre suas próprias criações.

Diante desse cenário, este trabalho apresenta um relato de experiência sobre a realização de um minicurso de Arduino voltado à integração entre Teoria de Autômatos e computação física. A proposta buscou proporcionar aos participantes uma compreensão mais concreta dos conceitos de AFD, por meio da implementação e observação de comportamentos de autômatos utilizando componentes eletrônicos e microcontroladores.

O artigo está estruturado de forma a apresentar: na Seção 2 o Referencial teórico; na Seção 3, os Trabalhos Relacionados; na Seção 4, a Metodologia do projeto; na Seção 5, os Resultados alcançados e, na Seção 6, as Conclusões.

2. Referencial Teórico

Esta seção apresenta os fundamentos teóricos que embasam o trabalho, abordando conceitos de Arduino e sistemas embarcados, os princípios dos Autômatos Finitos Determinísticos (AFD) e a importância da integração entre teoria e prática no ensino de Computação.

2.1. Arduino

A plataforma Arduino consolidou-se como uma das principais ferramentas para o ensino de computação física e robótica educacional devido ao seu baixo custo, facilidade de uso e ampla disponibilidade de materiais didáticos. Sua proposta permite que estudantes desenvolvam projetos interativos por meio da integração entre programação e componentes eletrônicos, favorecendo abordagens de aprendizagem prática e ativa [da Silva et al. 2023]. Nesse contexto, o Arduino tem sido amplamente utilizado em atividades educacionais para aproximar conceitos abstratos de situações concretas, estimulando a experimentação, a criatividade e a resolução de problemas.

No presente trabalho foi utilizada a placa *WeMos D1 R1*, compatível com o ecossistema Arduino e baseada no microcontrolador ESP8266EX [Espressif Systems 2015]. A escolha dessa placa deve-se à sua facilidade de programação, disponibilidade de recursos de entrada e saída e conectividade Wi-Fi integrada, características que ampliam as possibilidades de desenvolvimento de projetos educacionais [Idris et al. 2021]. Embora possua recursos voltados à Internet das Coisas (IoT), neste minicurso seu papel principal foi servir como plataforma de implementação e visualização prática dos modelos de Autômatos Finitos Determinísticos abordados no curso.

2.2. Autômatos Finitos Determinísticos

A Teoria da Computação investiga os limites e os recursos necessários para a resolução computacional de problemas através de modelos abstratos [Sipser 2007]. O modelo mais elementar é o Autômato Finito Determinístico (AFD), definido formalmente como uma quintupla $\langle Q, \Sigma, \delta, q_0, F \rangle$, onde Q é o conjunto finito de estados, Σ o alfabeto, $\delta : Q \times \Sigma \rightarrow Q$ a função de transição, $q_0 \in Q$ é o estado inicial e $F \subseteq Q$ é o conjunto de estados de aceitação [Sipser 2007].

O processamento de uma cadeia ocorre sequencialmente a partir de q_0 . A cada símbolo, o AFD altera sua configuração conforme δ . Se a computação terminar em um estado pertencente a F , a cadeia é aceita; caso contrário, é rejeitada. A Figura 1 ilustra um AFD com $Q = \{q_0, q_1, q_2\}$ e $F = \{q_2\}$, onde o processamento da cadeia 1101 segue o fluxo: $q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1$ (rejeitada, pois $q_1 \notin F$) [Sipser 2007].

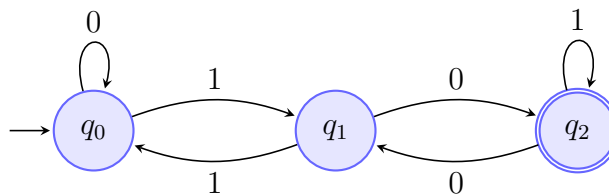


Figura 1. Autômato finito com 3 estados.

Como aplicação prática do modelo de Autômatos Finitos Determinísticos, [Sipser 2007] apresenta o clássico exemplo do controlador de uma porta automática. O sistema físico, ilustrado na Figura 2a, consiste em uma abertura equipada com dois tapetes sensores: um posicionado na parte frontal (para detectar a aproximação de alguém) e outro na parte posterior (para garantir a travessia segura e evitar que a porta feche precocemente).

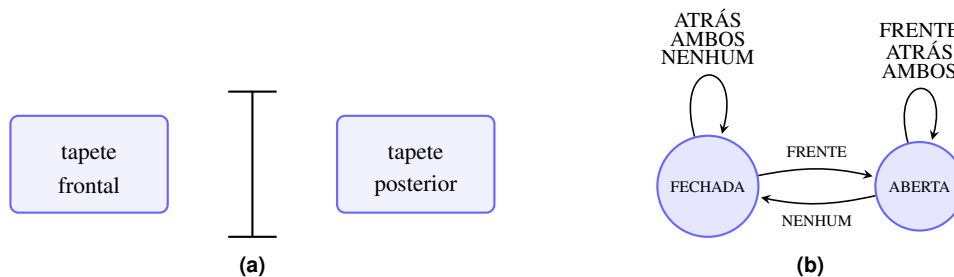


Figura 2. Controlador de porta automática e seu AFD correspondente.

Esse mecanismo é modelado formalmente por meio de estados e transições, conforme o diagrama apresentado na Figura 2b:

- **Estados (Q):** O sistema possui duas condições operacionais possíveis, representadas pelos estados ABERTA e FECHADA.
- **Entradas (Σ):** A combinação de leitura dos tapetes gera quatro estímulos possíveis: FRENTE (apenas sensor frontal ativo), ATRÁS (apenas sensor posterior ativo), AMBOS (ambos ativos) ou NENHUM (nenhum sensor acionado).

A transição entre os estados ocorre de forma puramente determinística: estando no estado FECHADA, a porta só altera sua condição para ABERTA caso receba o sinal FRENTE. Uma vez no estado ABERTA, ela permanece nele sob qualquer estímulo que indique a presença de pessoas

(FRENTE, ATRÁS ou AMBOS), retornando para FECHADA apenas quando o sinal detectado for NENHUM [Sipser 2007].

Outro exemplo que evidencia a aplicação prática dos AFDs é o semáforo de três LEDs, modelado como um sistema sequencial temporizado. Para estabelecer o rigor exigido na Computação e espelhar o modelo aplicado à porta automática, o comportamento desse circuito é expresso formalmente por meio da quintupla $M = \langle Q, \Sigma, \delta, q_0, F \rangle$, onde:

- $Q = \{\text{VERMELHO}, \text{AMARELO}, \text{VERDE}\}$: representa as três condições possíveis de iluminação dos LEDs;
- $\Sigma = \{t\}$: é o alfabeto do sistema, composto pelo estímulo de *timeout* gerado pelo cronômetro interno do hardware;
- $\delta : Q \times \Sigma \rightarrow Q$: é a função de transição que determina o avanço ordenado do ciclo tri-estável:

$$\delta(\text{VERMELHO}, t) = \text{AMARELO} \quad (\text{após } 5s)$$

$$\delta(\text{AMARELO}, t) = \text{VERDE} \quad (\text{após } 2s)$$

$$\delta(\text{VERDE}, t) = \text{VERMELHO} \quad (\text{após } 2s)$$

- $q_0 = \text{VERMELHO}$: indica o estado inicial de segurança em que o semáforo se encontra imediatamente ao ser energizado;
- $F = Q$: define que, neste contexto de controle contínuo e infinito, todas as fases de operação são válidas e representam condições estáveis de execução.

Os exemplos apresentados evidenciam uma característica fundamental dos sistemas embarcados: seu comportamento é frequentemente baseado em estados e eventos. Sensores, temporizadores e atuadores produzem entradas que alteram o estado interno do sistema, enquanto o microcontrolador executa ações associadas a cada estado. Dessa forma, modelos formais como os Autômatos Finitos Determinísticos fornecem uma representação matemática natural para o desenvolvimento de aplicações embarcadas [Dash et al. 2011].

2.3. A Relação entre Teoria e Prática na Computação

A articulação entre teoria e prática é um pilar da Computação, essencial para formar profissionais capazes de resolver problemas complexos em cenários dinâmicos. Segundo os referenciais da área, o profissional de Computação deve unir o domínio conceitual à capacidade de projetar e manter sistemas que integrem hardware e software [Sociedade Brasileira de Computação 2017]. O objetivo é equilibrar a compreensão teórica e a experimentação técnica para atender às demandas da sociedade.

[Papert 1980] defende que a tecnologia atua como uma ferramenta valiosa para a educação, impulsionando o desenvolvimento cognitivo e o aprendizado colaborativo. Alinhado à perspectiva do construcionismo, o autor aponta que o aprendizado torna-se mais efetivo quando o indivíduo constrói ativamente o próprio conhecimento, utilizando o computador não como um tutor passivo, mas como um instrumento de expressão e resolução de problemas.

As metodologias ativas colocam o estudante como protagonista do processo de aprendizagem por meio de projetos, experimentação e resolução de problemas reais. Nesse cenário, o Pensamento Computacional é desenvolvido de forma aplicada, permitindo que os estudantes utilizem conceitos computacionais na análise e solução de desafios concretos [Wing 2006,

Wing 2021]. Essas estratégias aumentam o engajamento e consolidam o conhecimento técnico por meio da experimentação [de Oliveira Colares et al. 2025], reduzindo a distância entre conceitos abstratos e a prática de desenvolvimento [Santiago et al. 2023]. Tal modelo reforça a orientação por competências, definida pela Sociedade Brasileira de Computação como a capacidade de mobilizar conhecimentos, habilidades e atitudes em situações reais [Sociedade Brasileira de Computação 2017].

3. Trabalhos Relacionados

A integração entre modelos teóricos e aplicações práticas no ensino superior tem sido explorada sob diversas perspectivas. No contexto do design de sistemas e *hardware*, [Ioannou et al. 2024] propõem o ambiente *Worksheet Integrated Development Environment* (WIDE), para o desenvolvimento de sistemas embarcados com Arduino. A ferramenta utiliza uma interface baseada em tabelas para que o estudante defina a descrição e o fluxo de estados, gerando automaticamente o código necessário para o microcontrolador. Esta abordagem estabelece uma relação direta entre a modelagem formal e a implementação física, permitindo que o foco do aluno de graduação esteja na lógica do sistema e na resolução de problemas, em vez de se limitar à sintaxe da linguagem de programação.

Buscando unir o engajamento interativo à teoria na graduação, o grupo Manna Team desenvolveu ferramentas como o *Labirinto Gramático* e o *Simplification Game* [Campano Junior et al. 2024]. Essas soluções traduzem conceitos complexos de gramáticas e autômatos para ambientes de experimentação lúdica, visando reduzir a carga cognitiva no aprendizado de Linguagens Formais. Da mesma forma, o jogo *Automigos* [Honda et al. 2023] utiliza mecânicas de jogos para representar elementos formais de AFDs. Contudo, apesar de promoverem a relação entre teoria e prática de modelagem, tais ferramentas mantêm o aprendizado limitado à simulação em tela.

O trabalho de [Vasconcelos and Guerra 2023], propõe o uso de Jupyter Notebooks no Google Colab como apoio ao ensino-aprendizagem de Teoria dos Autômatos. Nessa abordagem, o conteúdo teórico, definições e videoaulas são combinados com células de código em Python, que podem ser executadas e modificadas pelos alunos no navegador. Isso facilita o ensino de conceitos abstratos, como Autômatos Finitos e Máquinas de Turing, por meio de simulações e da execução passo a passo dos algoritmos. Assim, o aluno passa a interagir diretamente com os modelos estudados, o que favorece a compreensão prática.

Este trabalho diferencia-se das abordagens citadas ao propor a materialização do modelo formal por meio do hardware Arduino. Enquanto as soluções vigentes, como as propostas por pauloT e pelo grupo Manna Team [Campano Junior et al. 2024], priorizam a interação em tela e a simulação digital, nossa proposta traduz a lógica do autômato para um sistema de controle tangível.

Essa mudança de paradigma permite que o estudante conecte a teoria abstrata à prática de sistemas embarcados, visualizando o determinismo e as transições de estados como eventos físicos reais condicionados pelo hardware. Diferente do ambiente WIDE [Ioannou et al. 2024], que foca na automação do código para engenharia, este projeto utiliza o rigor da teoria de AFDs para gerenciar a interação física direta. Assim, oferece-se uma experiência de aprendizagem que expande o cenário de ferramentas puramente digitais hoje disponíveis para o ensino de graduação.

4. Metodologia

A metodologia adotada neste trabalho foi estruturada com o objetivo de integrar conceitos teóricos de AFDs com atividades práticas utilizando a plataforma Arduino. O minicurso foi planejado de forma incremental, partindo dos fundamentos teóricos necessários para, em seguida, permitir a implementação prática dos modelos computacionais em sistemas embarcados. A Figura 3 apresenta as principais etapas da metodologia adotada.

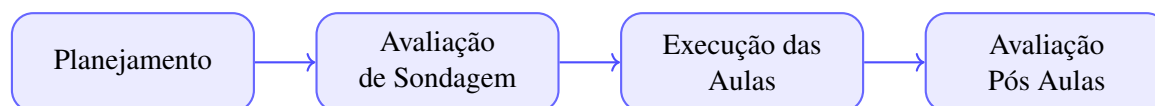


Figura 3. Etapas da metodologia adotada no minicurso.

4.1. Planejamento

A etapa de planejamento envolveu a definição dos conteúdos programáticos, a elaboração do material didático e a seleção das atividades práticas a serem desenvolvidas ao longo do minicurso. Foram definidos os seguintes eixos temáticos: (i) fundamentos de autômatos finitos determinísticos; (ii) introdução à plataforma Arduino e à eletrônica básica; e (iii) implementação de AFDs em sistemas embarcados.

Paralelamente, foi disponibilizado um formulário de inscrição com o objetivo de coletar informações necessárias à comunicação com os participantes e à posterior emissão dos certificados. Além disso, foram preparados os recursos de laboratório, incluindo kits de Arduino, componentes eletrônicos (LEDs, resistores, botões e protoboards) e tutorial configuração do ambiente de desenvolvimento (Arduino IDE).

4.2. Avaliação de Sondagem

No início do minicurso, foi aplicado um formulário de sondagem para identificar o perfil dos participantes, mapear seus conhecimentos prévios e ajustar o equilíbrio entre teoria e prática. As questões abordaram aspectos como familiaridade com Arduino e eletrônica, experiência com programação para sistemas embarcados, expectativas em relação ao minicurso e interesse em projetos de hardware.

O formulário de sondagem aplicado estruturou-se em oito questões fundamentais, englobando: o consentimento para o uso de dados anônimos na pesquisa (Q1); o nível de familiaridade prévia dos participantes com a plataforma Arduino (Q2) e com conceitos de eletrônica básica (Q3); a experiência anterior em programação para sistemas embarcados (Q4) e no manuseio de equipamentos de laboratório, tais como fontes, multímetros e osciloscópios (Q5); além do mapeamento de suas expectativas gerais em relação ao minicurso (Q6), interesse no desenvolvimento de projetos de hardware (Q7) e a existência de ideias prévias para a concepção de projetos próprios (Q8).

4.3. Execução das Aulas

O minicurso foi realizado em duas edições. A primeira ocorreu no semestre 2024.2 e contou com a participação de 24 estudantes. Já a segunda edição, realizada em 2025.2, teve a participação de 34 estudantes. O planejamento do minicurso foi organizado para ocorrer ao longo de quatro semanas, com a realização de duas aulas presenciais semanais, totalizando oito encontros. Cada

aula possuía duração de duas horas e era realizada no laboratório de eletrônica da instituição. As aulas foram ministradas por bolsistas do projeto, sob supervisão da tutora responsável. Durante as atividades práticas, os participantes foram organizados em duplas, permitindo o desenvolvimento colaborativo das implementações em Arduino.

4.4. Avaliação Pós Aulas

A pesquisa foi conduzida mediante a obtenção do Termo de Consentimento Livre e Esclarecido (TCLE). Todos os participantes foram previamente informados sobre os objetivos e a natureza acadêmica do estudo, participando de forma livre, voluntária e sem qualquer tipo de remuneração.

Para atender aos requisitos éticos da pesquisa, o formulário de avaliação aplicado no final do minicurso foi totalmente anônimo e de participação voluntária. Os estudantes foram informados de que o preenchimento não era obrigatório e não influenciaria a emissão de certificados. O questionário teve como objetivo avaliar o impacto das atividades em quatro dimensões: atendimento das expectativas, percepção de aprendizagem, trabalho em equipe e compreensão da relação entre a teoria dos autômatos e o controle de *hardware*.

Não foram coletados dados pessoais sensíveis nem informações que possibilitassem a identificação dos participantes. Os dados obtidos foram utilizados exclusivamente para fins científicos e acadêmicos, em conformidade com a Lei Geral de Proteção de Dados Pessoais (LGPD – Lei n.º 13.709/2018), sendo armazenados de forma segura e acessados apenas pelos pesquisadores envolvidos.

O formulário pós-minicurso foi composto por onze questões voltadas à avaliação do impacto da intervenção pedagógica, abrangendo: a renovação do consentimento para uso de dados (Q9); a verificação do atendimento das expectativas iniciais (Q10); a percepção de progresso individual em automação (Q11); o reforço do entendimento sobre autômatos finitos (Q12) e a familiaridade adquirida com sistemas embarcados (Q13). Adicionalmente, investigou-se o interesse despertado pela teoria de autômatos após a experiência prática (Q14); o impacto percebido dos conteúdos na formação acadêmica dentro da graduação (Q15); a identificação de possíveis lacunas de conteúdo (Q16); a efetividade da dinâmica de trabalho em equipe (Q17); a indicação dos tópicos considerados mais relevantes pelos estudantes (Q18); e, por fim, a compreensão da relação entre a teoria de autômatos e o controle de hardware real (Q19).

5. Resultados

O planejamento resultou na estrutura de oito aulas. As três primeiras aulas introduzem a plataforma Arduino e conceitos de eletrônica e programação; as aulas 4 e 5 apresentam os fundamentos teóricos de Autômatos Finitos Determinísticos (AFDs) e a modelagem formal dos sistemas a serem construídos; as aulas 6 e 7 dedicam-se à implementação prática da porta automática e do semáforo, respectivamente; e a aula 8 encerra o curso com a apresentação e discussão dos projetos.

A estrutura curricular do minicurso foi organizada em oito módulos sequenciais:

- **Aula 1 – Introdução ao Arduino:** Plataforma, instalação da IDE, computação física, uso de *protoboard*, componentes eletrônicos fundamentais e práticas iniciais com circuitos e LEDs.
- **Aula 2 – Programação Básica:** Variáveis, estruturas condicionais, entradas/saídas digitais, sinais analógicos e práticas com botões, LEDs e potenciômetros.

- **Aula 3 – Sensores e Atuadores:** Integração de sensores ultrassônicos e servomotores, explorando a relação entre *hardware* e *software* em sistemas embarcados.
- **Aula 4 – Fundamentos de AFDs:** Introdução conceitual aos Autômatos Finitos Determinísticos, abrangendo estados, transições, alfabetos, estado inicial e de aceitação.
- **Aula 5 – Modelagem de Sistemas Reativos:** Construção formal de modelos para uma porta automática e um semáforo via diagramas de estados, discutindo sistemas físicos e computacionais.
- **Aula 6 – Prática da Porta Automática:** Implementação do controlador em Arduino usando LEDs, botões e servomotor, codificando as transições de estados definidas pelo AFD.
- **Aula 7 – Prática do Semáforo:** Montagem do circuito eletrônico com LEDs e desenvolvimento da lógica de controle temporal baseada no modelo formal do autômato.
- **Aula 8 – Apresentação de Projetos:** Demonstração dos sistemas implementados pelos participantes e debate sobre a integração entre Teoria da Computação e sistemas embarcados.

5.1. Prática I - Controlador de Porta Automática

A prática da porta automática foi conduzida em duas etapas. Na primeira, os participantes montaram uma porta simples controlada por botão e servomotor, sem formalização explícita do autômato (Figura 4). Na segunda, foi simulado o modelo formal de [Sipser 2007]: dois botões representaram os tapetes frontal e posterior e um servomotor representou a porta. A cada ciclo do `loop()`, o estado atual era verificado e as entradas determinavam a transição seguinte, tornando observável no hardware o determinismo previsto pelo AFD.

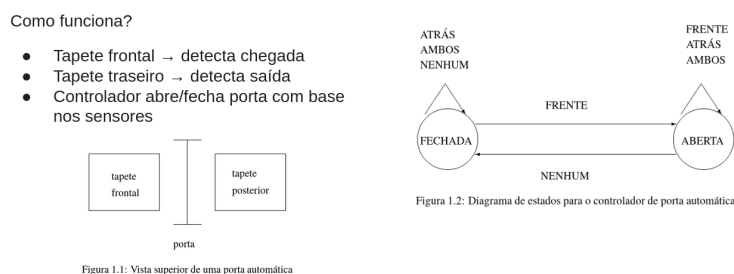


Figura 4. Montagem do controlador de porta automática.

5.2. Prática II - Controlador de Semáforo

O semáforo foi apresentado como autômato sequencial antes da montagem em hardware (Figura 5): verde, amarelo e vermelho correspondem a estados do sistema, com transições temporizadas e sem entradas externas. A partir desse modelo, os alunos construíram o circuito com três LEDs na placa Wemos D1 R1, traduzindo cada estado em uma combinação de sinais digitais e cada transição em um intervalo controlado por `delay()`, tornando diretamente observável o funcionamento do autômato no hardware.

5.3. Resultados da Avaliação Pré-Curso

A Tabela 1 apresenta o perfil de conhecimento prévio dos alunos que responderam ao questionário inicial. Na primeira edição obtivemos 12 respostas; na segunda, 25. Três alunos participaram das

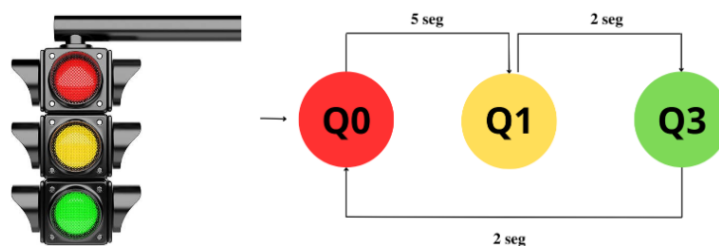


Figura 5. Montagem do controlador de semáforo.

Tabela 1. Perfil de conhecimento prévio dos participantes (% nos níveis 1 e 2)

Variável	1ª edição (n = 12)	2ª edição (n = 25)
Familiaridade com Arduino (níveis 1–2)	50,0%	84,0%
Familiaridade com eletrônica (níveis 1–2)	58,3%	88,0%
Familiaridade c/ prog. embarcada/IoT (níveis 1–2)	72,8%	88,0%
Interesse em projetos de eletrônica (níveis 4–5)	91,7%	72,0%

duas edições. Optamos por agregar os níveis 1 e 2 da escala Likert para evidenciar o caráter introdutório do público.

Em ambas as edições a maioria declarou-se iniciante, o que valida a proposta do minicurso como porta de entrada para o tema. Destaca-se o aumento da inexperiência na segunda edição: 84% dos alunos estavam nos níveis mais baixos de familiaridade com Arduino, frente a 50% na primeira edição. O dado mais expressivo refere-se à programação embarcada: na segunda edição 88% situavam-se no nível 1, ante 45,5% na primeira. Ainda assim, o interesse por projetos de eletrônica permaneceu elevado nas duas turmas (91,7% e 72%, respectivamente, nos níveis máximos de motivação), demonstrando que a falta de experiência prévia não inibiu o engajamento dos alunos.

5.4. Resultados da Avaliação Pós-Curso

Os resultados indicam que a proposta foi eficaz em aproximar os conceitos de Autômatos Finitos Determinísticos da prática em sistemas embarcados. Quando questionados sobre a compreensão da relação entre autômatos e Arduino, 88,9% dos participantes responderam positivamente em 2024, percentual que alcançou 100% em 2025. De forma semelhante, o interesse em aprofundar os conhecimentos na área permaneceu elevado nas duas edições (88,9% e 91,7%, respectivamente), sugerindo que a abordagem prática não apenas favoreceu a aprendizagem, mas também despertou a curiosidade dos estudantes para explorar novos conteúdos.

Embora nenhum participante tenha relatado desconforto durante as atividades, observou-se uma redução na percepção de conforto pleno entre 2024 (55,6%) e 2025 (25%), acompanhada pelo aumento do conforto moderado (44,4% para 75%). Esse resultado pode ser explicado pelo perfil mais iniciante da segunda turma, que demandou maior adaptação aos conceitos e ferramentas apresentados. Ainda assim, a ausência de avaliações negativas sugere que a metodologia foi capaz de acomodar diferentes níveis de experiência.

As respostas abertas reforçam essa interpretação. O termo “autômatos” foi mencionado espontaneamente por 22% dos participantes, evidenciando que os estudantes reconheceram a teoria como elemento central das atividades desenvolvidas. Além disso, o progresso autodeclarado em Arduino e automação aumentou de 77,7% em 2024 para 91,6% em 2025. Esse crescimento

pode estar associado ao amadurecimento da proposta didática e à inclusão de projetos envolvendo IoT, frequentemente citados pelos participantes como uma forma de familiarização com o tema e inspiração para o desenvolvimento de novos projetos.

Outro aspecto relevante foi a colaboração entre os participantes. A avaliação positiva do trabalho em equipe permaneceu elevada em ambas as edições, alcançando 100% em 2024 e 91,7% em 2025. A dinâmica colaborativa favoreceu a troca de conhecimentos e a resolução conjunta de problemas, contribuindo para a construção coletiva do aprendizado.

De maneira geral, os resultados sugerem que a utilização do Arduino como ferramenta de ensino permitiu transformar conceitos abstratos de modelagem por estados em experiências concretas e observáveis. A implementação de circuitos, sensores, atuadores e protocolos de comunicação, possibilitou aos estudantes visualizar como estados e transições são empregados no desenvolvimento de sistemas reais. Essa conexão entre teoria e prática mostrou-se particularmente importante para o desenvolvimento do Pensamento Computacional, ao incentivar a decomposição de problemas, a modelagem de comportamentos e a construção de soluções de forma estruturada. Como lição aprendida, destaca-se que metodologias ativas apoiadas por computação física podem tornar conteúdos formais mais acessíveis e significativos, desde que acompanhadas de um planejamento cuidadoso capaz de equilibrar os aspectos teóricos e práticos da aprendizagem.

6. Considerações Finais

Este trabalho apresentou um relato de experiência sobre a realização de um minicurso de Arduino voltado à aproximação entre os conceitos da Teoria dos Autômatos e a prática em sistemas embarcados. Os resultados obtidos indicam que a utilização de atividades práticas contribuiu para tornar conteúdos tradicionalmente abstratos mais acessíveis e compreensíveis aos participantes. Ao relacionar estados, transições e eventos com sensores, atuadores e temporizadores, os estudantes puderam observar, de forma concreta, como os princípios dos Autômatos Finitos Determinísticos estão presentes no funcionamento de sistemas reais.

Entre as principais lições aprendidas, destaca-se que a materialização de conceitos teóricos por meio do hardware favorece a compreensão de modelos computacionais abstratos, reduzindo a distância entre teoria e prática. Além disso, a utilização do Arduino mostrou-se uma estratégia eficaz para aumentar o engajamento dos participantes, estimular a aprendizagem ativa e promover discussões sobre a aplicação dos conceitos estudados em problemas do mundo real. Observou-se ainda que mesmo estudantes sem experiência prévia em sistemas embarcados foram capazes de compreender a relação entre autômatos e o comportamento de dispositivos programáveis, evidenciando o potencial dessa abordagem como ferramenta de apoio ao ensino de disciplinas teóricas da Computação.

As avaliações realizadas também indicaram uma percepção positiva quanto ao aprendizado e à organização das atividades, sugerindo que experiências práticas e contextualizadas podem contribuir para aumentar o interesse dos estudantes por conteúdos formais frequentemente considerados complexos ou excessivamente abstratos.

Como trabalhos futuros, pretende-se expandir a proposta para contemplar conceitos mais avançados de Linguagens Formais e Teoria da Computação, incluindo atividades relacionadas a Autômatos Finitos Não Determinísticos (AFNDs) e Máquinas de Estados mais complexas. Também se pretende investigar o impacto da abordagem em contextos educacionais distintos, comparando seu efeito sobre a aprendizagem, o engajamento e a retenção de conhecimento em disciplinas da área de Computação.

Declaração sobre uso de Inteligência Artificial

Durante o desenvolvimento deste trabalho, utilizamos LLMs para apoiar à revisão gramatical, à geração de tabelas e alinhamento de imagens.

Referências

- Binsfeld, R. L., Watanabe, R., Silva, R. C., and Carelli, I. M. (2009). Alunos como designers: relato de experiência para aprendizagem de linguagens formais e autômatos. In *VIII Brazilian Symposium on Games and Digital Entertainment*, pages 21–24. sn.
- Campano Junior, M. M., Assumpção, M., Gama, G. V. d. M., Track, M. B., and Aylon, L. B. R. (2024). Manna team games: jogos educativos para ensino de computação. In *Anais do XXXIII Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames 2024)*, Manaus, AM, Brasil. SBC. WGP-Jogos.
- da Silva, F. J. M., Silva, E. C., de Sousa Lopes, K. C., Brito, W. M., Bezerra, C., Silva, M. M., and de Menezes, M. V. (2023). Arduino como ferramenta para ensino de programação em sistemas embarcados: Um relato de experiência. In *Workshop de Informática na Escola (WIE)*, pages 670–681. SBC.
- Dash, N. P., Dasgupta, R., Chepada, J., and Halder, A. (2011). Event driven programming for embedded systems-a finite state machine based approach. In *The Sixth International Conference on Systems*, pages 23–28.
- de Oliveira Colares, A. F., Oliveira, S. R. B., and Furtado, J. C. C. (2025). A qualitative evaluation of an experiment on the application of active methodologies in teaching software process improvement. In *Simpósio Brasileiro de Qualidade de Software (SBQS)*, pages 463–473. SBC.
- Espressif Systems (2015). *ESP8266EX Datasheet*. Versão 6.7.
- Honda, F., Pires, F., Pessoa, M., and Melo, R. (2022). Aplicando learning design na ludificação de percurso em grafos: uma jornada de aprendizagem. In *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*, pages 609–620, Porto Alegre, RS, Brasil. SBC.
- Honda, F., Pires, F., Pessoa, M., and Oliveira, E. (2023). Automigos: learning design para ludificação de autômatos finitos determinísticos. In *Anais do XXXI Workshop sobre Educação em Computação*, pages 545–556, Porto Alegre, RS, Brasil. SBC.
- Idris, M. R., Mat Nashir, I., Nordin, A., and Ramlee, M. (2021). *The Internet of Things (IoT) Practical Book using Arduino WeMos D1 R1 Microcontroller*.
- Ioannou, N., Tatas, K., Constantinides, A., and Kyriacou, C. (2024). Wide: Worksheet integrated development environment for arduino-based embedded system design. In *2024 13th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, pages 1–5.
- Melo, E. and Barbosa, J. A. (2025). Uma introdução a robótica educacional com arduino para o pensamento computacional. In *Anais do IX Congresso sobre Tecnologias na Educação*, pages 520–526, Porto Alegre, RS, Brasil. SBC.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.
- Sangali, R., Catabriga, L., and Boeres, M. C. (2024). Robótica educativa para desenvolvimento de habilidades do pensamento computacional por meio de eletiva complementar. In *Anais do XXX Workshop de Informática na Escola*, pages 437–448, Porto Alegre, RS, Brasil. SBC.

- Santiago, C. P., Menezes, J. W. M., and de Aquino, F. J. A. (2023). Proposta e avaliação de uma metodologia de aprendizagem baseada em projetos em disciplinas de engenharia de software através de uma sequência didática. *Revista Brasileira de Informática na Educação*, 31:31–59.
- Santos, G. and Oliveira, R. (2025). Robótica educacional e pensamento computacional: Uma abordagem prática no ensino fundamental. In *Anais do II Simpósio Brasileiro de Computação na Educação Básica*, pages 118–122, Porto Alegre, RS, Brasil. SBC.
- Sipser, M. (2007). *Introdução à teoria da computação*. Cengage Learning, São Paulo, 2 edition.
- Sociedade Brasileira de Computação (2017). Referenciais de formação para cursos de graduação em computação. Disponível em: <https://www.sbc.org.br>.
- Vasconcelos, D. and Guerra, P. (2023). Ensinando teoria da computação com jupyter notebook. In *Anais do XXXI Workshop sobre Educação em Computação*, pages 9–19, Porto Alegre, RS, Brasil. SBC.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.
- Wing, J. M. (2021). Pensamento computacional. *Educação e Matemática*, (162):2–4.