ISSN: 2358-8829

EXPRESSÕES REGULARES COMO RECURSO DIDÁTICO PARA O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL E LÓGICO

João Pedro Gonçalves Dias da Silva[1]
Harlem Alves do Nascimento [2]

1 INTRODUÇÃO

O pensamento computacional tem recebido atenção no campo educacional por abranger maneiras de formular problemas, representar informações e construir soluções passíveis de execução ou verificação. Na formulação de Wing (2006), não se trata de um conjunto de conhecimentos restrito aos profissionais de computação, mas de uma perspectiva que mobiliza conceitos fundamentais dessa área em múltiplos domínios. Essa ampliação coloca um desafio didático: identificar atividades que tornem visíveis processos de raciocínio frequentemente ocultos por trás de linguagens de programação, ferramentas digitais e educação tecnológica.

Uma possibilidade é explorar tarefas que exigem dos estudantes observar regularidades, estabelecer categorias, formular regras e avaliar se determinadas situações satisfazem critérios previamente definidos. Tais ações aparecem em problemas cotidianos, como reconhecer formatos de códigos de identificação em matrículas acadêmicas, separar palavras com características comuns e verificar padrões de preenchimento. Quando formalizadas, podem ser representadas por expressões regulares, também conhecidas pela abreviação Regex, recurso empregado em busca, identificação e validação de sequências de caracteres.

Entretanto, a introdução de Regex costuma ocorrer diretamente por meio de símbolos como colchetes, parênteses, operadores de repetição e alternativas. Para iniciantes, o contato isolado com esses sinais pode deslocar a atenção da lógica subjacente para a memorização

[1] Graduando do Curso de Ciência da Computação da Universidade Estadual da Paraíba - PB, contato.pedrojson@gmail.com;

[2] Professor orientador: Bacharel em Ciência da Computação pela Universidade Estadual da Paraíba - PB, harllem.nascimento@servidor.uepb.edu.br;

ISSN: 2358-8829

sintática. É justamente nessa dificuldade que se localiza o problema educacional deste estudo: Quais são as possibilidades e limitações pedagógicas da utilização de conceitos associados às Expressões Regulares, por meio de atividades concretas e programação introdutória, para o ensino do pensamento computacional e lógico?

O objetivo geral é analisar o potencial pedagógico das expressões regulares como referência conceitual para o ensino introdutório do pensamento computacional, a partir de uma oficina efetivamente realizada com atividades concretas e programação básica. Especificamente, busca-se: (i) relacionar o reconhecimento de padrões e as regras de aceitação a habilidades do pensamento computacional; (ii) descrever as atividades desenvolvidas nos dois encontros; (iii) discutir a transição entre a classificação de cartas e as estruturas de programação em Python; e (iv) analisar as observações e os limites da experiência.

A contribuição do estudo consiste em documentar uma experiência pedagógica com dois encontros, incluindo uma oficina prática com aproximadamente 10 participantes, e discutir suas relações com a teoria das linguagens formais. A intervenção não é apresentada como experimento controlado nem como demonstração de eficácia. Como não houve avaliação formal da aprendizagem, as observações relatadas são examinadas de maneira qualitativa, sem pretensão de generalização.

2 REFERENCIAL TEÓRICO

2.1 Pensamento computacional e raciocínio lógico

Wing (2006) descreve o pensamento computacional em conexão com a resolução de problemas, o desenho de sistemas e a compreensão de processos. Essa perspectiva permite abordar a computação não apenas como operação de máquinas, mas como construção de representações e procedimentos. No âmbito pedagógico, é frequente organizar atividades em torno de quatro processos: decomposição, reconhecimento de padrões, abstração e pensamento algorítmico. A organização em quatro elementos é uma síntese didática útil, porém não deve ser interpretada como uma definição única ou exaustiva do campo.

Na decomposição, um problema maior é separado em partes observáveis; no reconhecimento de padrões, identificam-se semelhanças e diferenças entre exemplos; na abstração, características relevantes são preservadas enquanto informações acessórias são

ISSN: 2358-8829

desconsideradas; e, no pensamento algorítmico, procedimentos claros e ordenados são elaborados para realizar determinada tarefa. Esses movimentos não ocorrem necessariamente em uma ordem rígida. Ao identificar que um código possui uma letra inicial e dois algarismos, por exemplo, o estudante simultaneamente distingue componentes, percebe uma estrutura comum e explicita uma regra.

Brennan e Resnick (2012) chamam atenção para a necessidade de examinar o pensamento computacional em diferentes dimensões, que incluem conceitos, práticas e perspectivas. Para os autores, observar somente a resposta final empobrece a análise da aprendizagem: ações como testar, revisar, identificar erros e explicar escolhas também fornecem evidências relevantes. Essa compreensão fundamenta a decisão metodológica de incluir, na oficina, justificativas, discussão de contraexemplos e correção de padrões mal formulados.

A aproximação entre raciocínio lógico e Regex não exige que a pessoa conheça previamente uma linguagem de programação. Ao considerar um conjunto de sequências, o participante precisa decidir quais pertencem a uma categoria e quais devem ser excluídas, explicar os critérios de pertencimento e revisar a regra caso apareça um caso que a contradiga. Trata-se de um exercício de classificação baseado em condições explícitas, que pode estimular argumentação e precisão na comunicação.

2.2 Expressões regulares e teoria das linguagens formais

A Teoria das Linguagens Formais estuda conjuntos de cadeias formadas a partir de alfabetos e regras de composição. Nesse contexto, uma linguagem regular pode ser descrita por uma expressão regular e reconhecida por um autômato finito. Um autômato finito é um modelo matemático de computação composto por um conjunto limitado de estados e regras de transição entre eles. Seu funcionamento consiste em processar sequencialmente os símbolos de uma entrada, alterando seu estado conforme regras previamente estabelecidas e determinando, ao final, se a sequência é aceita ou rejeitada. A correspondência entre expressões regulares e autômatos é apresentada no *Computer Science Field Guide*, desenvolvido pela University of Canterbury, e integra o estudo das linguagens regulares no curso *Theory of Computation*, disponibilizado pelo Massachusetts Institute of Technology (MIT, 2020).

ISSN: 2358-8829

Uma expressão regular reúne símbolos literais e operadores que especificam padrões. Entre as construções introdutórias utilizadas em ferramentas comuns estão a concatenação, a alternativa, a repetição e as classes de caracteres. Por exemplo, a expressão "^C[0-9]{2}$" descreve, sob uma sintaxe habitual de mecanismos de Regex, cadeias inteiras compostas pela letra "C" seguida exatamente por dois algarismos. Nesse caso, "C07" e "C42" satisfazem a regra, enquanto "C7", "C123" e "D07" não a satisfazem. Os símbolos "^" e "$" indicam, nesse uso, respectivamente, o início e o fim da cadeia.

É importante distinguir a teoria formal dos mecanismos usados em linguagens de programação. Recursos como classes de caracteres e âncoras são conveniências presentes em numerosas implementações, mas não constituem necessariamente os operadores primitivos empregados na definição matemática de expressão regular. Além disso, algumas implementações oferecem recursos adicionais que ultrapassam o poder expressivo das linguagens regulares. Para evitar confusões, este artigo limita os exemplos da oficina a construções simples, compatíveis com a compreensão introdutória de padrões regulares.

A utilidade didática não reside somente em obter uma notação correta. Mais relevante é a possibilidade de comparar a regra escrita com um conjunto de exemplos: a expressão aceita casos que deveriam ser rejeitados? Exclui algum exemplo desejado? Permite uma sequência vazia quando a tarefa exige ao menos um elemento? Questões desse tipo tornam explícitos aspectos da modelagem, da verificação e da depuração, essenciais à aprendizagem de conceitos computacionais.

2.3 Educação desplugada, ludicidade e mediação

O ensino desplugado de computação propõe atividades que tornam conceitos computacionais acessíveis sem depender da utilização direta de computadores. Henderson (2008), ao apresentar atividades do projeto Computer Science Unplugged, ressalta o uso de materiais práticos e envolventes para explorar conteúdos da área com públicos variados. A proposta discutida neste artigo aproxima-se dessa perspectiva porque utiliza cartas de baralho como suportes concretos para a compreensão progressiva de regras formais.

O Digital Technologies Hub divulga o capítulo Formal Languages, do Computer Science Field Guide, como recurso para o ensino de linguagens formais, expressões regulares e autômatos finitos. Nesse material, o conteúdo é abordado por exemplos de correspondência e desafios interativos. A consulta reforça a pertinência de uma sequência que primeiro

ISSN: 2358-8829

estimule os estudantes a experimentar padrões e depois introduza a notação técnica, sem apresentar os símbolos como um conjunto arbitrário de comandos.

A ludicidade pode aparecer em desafios graduais e na manipulação de objetos, mas não se confunde automaticamente com gamificação. Kapp (2012) discute a incorporação de elementos de jogos a experiências instrucionais, observando a relação entre estratégias adotadas e objetivos de aprendizagem. Neste estudo, a escolha por pequenos puzzles lógicos privilegia exploração, feedback e possibilidade de revisão, evitando reduzir a atividade à atribuição de pontos. O engajamento, embora desejável, deve ser observado em aplicação e não presumido a partir do formato do desafio.

Em trabalho de natureza empírica, Rico Lugo (2016) desenvolveu e implementou o material educativo digital *Evolución*, composto por desafios matemáticos e algorítmicos, com estudantes de lógica de programação. O autor relata dificuldades iniciais dos participantes e descreve a relevância do exercício reiterado dos desafios para o processo de aprendizagem no contexto investigado. A experiência não demonstra efeitos específicos de Regex, mas oferece uma referência para pensar a progressão de tarefas, o uso pedagógico dos erros e a importância da mediação ao trabalhar raciocínio algorítmico.

A passagem do concreto para o abstrato, portanto, é compreendida aqui como decisão de desenho didático, e não como uma garantia de que qualquer material físico produzirá aprendizagem. Seu valor depende do tipo de pergunta proposta pelo professor: pedir que separem peças por cor é diferente de solicitar que construam uma regra geral, examinem os limites dessa regra e encontrem contraexemplos que a tornem insuficiente.

3 METODOLOGIA

O presente trabalho caracteriza-se como um relato de experiência de caráter exploratório e abordagem qualitativa, fundamentado na descrição e análise de uma experiência pedagógica realizada em dois encontros, envolvendo a apresentação de conceitos associados às Expressões Regulares, atividades concretas de reconhecimento de padrões e programação introdutória em Python.

A experiência pedagógica foi desenvolvida em dois encontros síncronos, realizados remotamente por meio da plataforma *Google Meet*, com participantes de 17 a 19 anos vinculados ao primeiro período de um curso superior da área de Computação. O primeiro

ISSN: 2358-8829

encontro contou com aproximadamente 20 participantes e apresentou caráter predominantemente expositivo, enquanto o segundo reuniu aproximadamente 10 participantes e foi estruturado como uma oficina didática.

A análise da experiência fundamentou-se em observações realizadas pelo responsável pelas atividades, registradas manualmente em um caderno. As anotações serviram de base para a descrição das situações observadas, especialmente em relação à compreensão das regras de classificação e às dificuldades apresentadas durante a introdução das estruturas de programação em Python. Não foram utilizados instrumentos padronizados de avaliação da aprendizagem, sendo os registros interpretados de maneira qualitativa e descritiva.

O primeiro encontro contou com aproximadamente 20 participantes e apresentou caráter predominantemente expositivo. A atividade foi dedicada à apresentação e discussão de conceitos introdutórios de Expressões Regulares, reconhecimento de padrões, regras de aceitação e rejeição de símbolos e fundamentos do funcionamento de autômatos finitos. Durante a exposição, buscou-se relacionar esses conteúdos a situações cotidianas que envolvem a identificação de regularidades e a aplicação de critérios de classificação, com o objetivo de tornar mais acessíveis conceitos frequentemente considerados abstratos. O encontro priorizou a introdução conceitual e a discussão dos fundamentos teóricos, não sendo possível desenvolver atividades práticas devido às limitações de tempo.

O segundo encontro contou com aproximadamente 10 participantes e foi estruturado como uma oficina didática, envolvendo a utilização de materiais concretos e atividades introdutórias de programação. A proposta buscou estabelecer uma transição gradual entre a compreensão de regras lógicas por meio de recursos físicos e sua representação computacional.

Inicialmente, foram retomados os conceitos de reconhecimento de padrões e aceitação de símbolos. Na etapa prática, utilizou-se um baralho convencional, cujas cartas eram apresentadas sequencialmente pela câmera durante o encontro realizado no *Google Meet*. A cada carta, os participantes deveriam indicar se ela atendia à regra estabelecida, que consistia em aceitar exclusivamente cartas do naipe de copas e rejeitar as demais. Ao longo da dinâmica, também foi realizada a contagem das cartas aceitas, permitindo explorar operações de classificação, verificação de condições e contabilização de elementos.

A cada carta apresentada, os participantes eram convidados a identificar suas características e responder, em tempo real, se ela deveria ser aceita ou rejeitada conforme a regra estabelecida. As cartas pertencentes ao naipe de copas eram consideradas aceitas, enquanto aquelas dos demais naipes eram rejeitadas.

ISSN: 2358-8829

A dinâmica buscou tornar concretos os processos de reconhecimento de padrões, verificação de condições e tomada de decisão, estabelecendo uma analogia introdutória com conceitos relacionados às expressões regulares e aos autômatos finitos.

Posteriormente, foi realizada uma etapa de programação em Python, na qual foram apresentados conceitos básicos de sintaxe, estruturas condicionais (if/else) e estruturas de repetição (for).

Nessa etapa, foram explorados procedimentos de identificação, aceitação e rejeição de elementos com base em características específicas, incluindo exemplos envolvendo números, letras e cartas de diferentes naipes.

A utilização das Expressões Regulares ocorreu no âmbito conceitual e pedagógico, como referência para a compreensão das regras de reconhecimento de padrões. Não houve implementação direta de expressões regulares no código Python desenvolvido, uma vez que o objetivo central da oficina consistia na introdução aos fundamentos do pensamento computacional e da lógica de programação.

A análise da experiência foi realizada qualitativamente, considerando as observações relatadas pelo responsável pela atividade acerca da compreensão dos conceitos, das dificuldades identificadas e das estratégias utilizadas pelos participantes durante a resolução dos desafios.

Por não terem sido utilizados instrumentos padronizados de avaliação de aprendizagem, a análise concentra-se na descrição e interpretação da experiência pedagógica, sem estabelecer relações causais entre a intervenção e o desenvolvimento de habilidades cognitivas.

4 RESULTADOS E DISCUSSÃO

A realização dos encontros possibilitou observar aspectos relacionados à utilização de conceitos das Expressões Regulares e das linguagens formais como recursos introdutórios ao ensino do pensamento computacional.

No primeiro encontro, de caráter expositivo, a apresentação dos fundamentos de reconhecimento de padrões, aceitação de símbolos e funcionamento de autômatos proporcionou uma aproximação inicial dos participantes com os conteúdos abordados.

ISSN: 2358-8829

Segundo a percepção do responsável pela atividade, as explicações apresentadas foram compreendidas de maneira satisfatória. Entretanto, a ausência de atividades práticas e de instrumentos de avaliação impossibilitou uma análise mais aprofundada da aprendizagem nesse momento.

No segundo encontro, a utilização de cartas de baralho permitiu explorar de maneira concreta conceitos relacionados à identificação de características, classificação de elementos e aplicação de regras lógicas.

Durante a apresentação sequencial das cartas pela câmera, os participantes responderam às situações propostas, identificando se cada elemento atendia ou não ao critério estabelecido. Segundo as observações registradas pelo responsável pela oficina, a identificação do naipe e a aplicação da regra de aceitação e rejeição apresentaram menor dificuldade em comparação com a compreensão posterior das estruturas de repetição em Python.

A dinâmica também permitiu discutir a abstração, uma vez que, para resolver o desafio, era necessário considerar apenas o naipe da carta, desconsiderando características irrelevantes para a classificação, como seus valores numéricos ou figuras.

Outro aspecto relevante foi a relação entre a atividade concreta e os fundamentos da lógica de programação. Ao transportar os procedimentos de classificação para exemplos desenvolvidos em Python, tornou-se possível explorar diferentes estruturas de controle e suas respectivas funções.

As observações relatadas indicaram maior facilidade na compreensão das estruturas condicionais (if/else), utilizadas para representar decisões baseadas em critérios previamente estabelecidos.

Por outro lado, a estrutura de repetição (for) apresentou maior dificuldade de compreensão, especialmente em relação à necessidade de executar sucessivamente uma mesma operação sobre diferentes elementos de uma sequência.

Essa diferença pode estar associada à natureza das operações envolvidas. Enquanto as estruturas condicionais representam decisões pontuais, os laços de repetição exigem compreender o processamento sequencial de múltiplos elementos e a atualização das informações ao longo da execução.

Apesar das dificuldades inicialmente observadas com a repetição, os participantes envolvidos nos desafios de classificação demonstraram capacidade de identificar e contabilizar as cartas pertencentes ao naipe solicitado sem necessidade de auxílio direto, conforme o relato da experiência.

ISSN: 2358-8829

Essa observação sugere que o estabelecimento de relações entre ações concretas e operações computacionais pode favorecer a compreensão inicial de procedimentos algorítmicos.

A experiência também permitiu identificar relações conceituais entre as atividades desenvolvidas e os quatro pilares frequentemente associados ao pensamento computacional: decomposição, reconhecimento de padrões, abstração e pensamento algorítmico.

A decomposição esteve relacionada à divisão do problema em etapas, como selecionar uma carta, identificar seu naipe, verificar a condição estabelecida e determinar sua classificação.

O reconhecimento de padrões foi explorado por meio da identificação de características comuns entre diferentes elementos. A abstração esteve presente na seleção das propriedades relevantes para a resolução do problema, enquanto o pensamento algorítmico foi abordado pela organização de procedimentos sequenciais e sua posterior representação em estruturas de programação.

Nesse contexto, as Expressões Regulares constituíram uma referência conceitual para a compreensão de padrões e regras de reconhecimento, embora não tenham sido utilizadas diretamente como instruções executáveis nos programas desenvolvidos.

Essa distinção é relevante, pois evidencia que os fundamentos associados às linguagens formais podem ser explorados pedagogicamente antes da introdução de sua sintaxe específica.

Os resultados observacionais apresentam aproximações com abordagens de computação desplugada, nas quais materiais concretos são utilizados para representar conceitos computacionais e apoiar atividades introdutórias de ensino.

Entretanto, é necessário reconhecer as limitações da experiência. O número reduzido de participantes na etapa prática, a ausência de instrumentos formais de avaliação e a inexistência de medidas anteriores e posteriores à intervenção impedem a comprovação de ganhos efetivos de aprendizagem.

Assim, os resultados apresentados não demonstram a eficácia da metodologia em termos experimentais, mas oferecem elementos para discutir suas possibilidades de utilização em contextos introdutórios de ensino de Ciência da Computação.

A experiência indica que a associação entre materiais manipuláveis, reconhecimento de padrões e programação básica constitui uma possibilidade pedagógica a ser investigada em estudos posteriores com procedimentos sistemáticos de avaliação.

ISSN: 2358-8829

5 CONSIDERAÇÕES FINAIS

Este artigo discutiu o uso de fundamentos conceituais das expressões regulares como recurso didático para o pensamento computacional e lógico, articulando linguagens formais, atividades com cartas de baralho e programação introdutória em Python. A experiência pedagógica relatada ocorreu em dois encontros: uma apresentação expositiva com aproximadamente vinte participantes e uma oficina prática com aproximadamente 10 participantes.

Na oficina, a classificação sequencial de cartas por regras de aceitação e rejeição estabeleceu uma ligação entre ações concretas e decisões computacionais. As observações relatadas apontaram compreensão relativamente simples dos naipes e das estruturas condicionais if/else, em contraste com maior dificuldade diante do laço for. Os participantes conseguiram identificar as cartas que atendiam à regra estabelecida e realizar sua contagem durante a atividade conduzida pelo *Google Meet*, conforme as observações registradas.

A oficina foi realizada, mas não incluiu a implementação direta de expressões regulares em código: Regex funcionou como fundamento teórico-pedagógico para reconhecer e descrever padrões, enquanto os programas desenvolvidos utilizaram estruturas básicas de Python. A ausência de instrumentos padronizados, registros individuais e avaliação antes e depois da intervenção limita as conclusões a uma discussão exploratória das possibilidades didáticas observadas.

Como continuidade, recomenda-se avaliar a abordagem em investigações com procedimentos de registro e avaliação sistemáticos, verificando especialmente a compreensão de estruturas de repetição, o reconhecimento de padrões e a capacidade de justificar regras de classificação. O uso das observações de participantes em publicação científica também deve ter o enquadramento ético confirmado com o orientador e a instituição.

REFERÊNCIAS

BRENNAN, Karen; RESNICK, Mitchel. *New frameworks for studying and assessing the development of computational thinking*. In: ANNUAL MEETING OF THE AMERICAN EDUCATIONAL RESEARCH ASSOCIATION, 2012. Vancouver. Anais [...]. 2012. Disponível em:

https://www.media.mit.edu/publications/new-frameworks-for-studying-and-assessing-the-development-of-computational-thinking/. Acesso em: 28 out. 2026.

DIGITAL TECHNOLOGIES HUB. *CS Unplugged: Field guide: Formal Languages*. [S. l.], [s. d.]. Disponível em: https://www.digitaltechnologieshub.edu.au/search/cs-unplugged-field-guide-formal-languages/. Acesso em: 28 out. 2026.

HENDERSON, Peter B. Computer science unplugged. *Journal of Computing Sciences in Colleges*, v. 23, n. 3, p. 168, jan. 2008. Disponível em: https://www.researchgate.net/publication/262220173_Computer_Science_unplugged. Acesso em: 29 out. 2026.

KAPP, Karl M. *The gamification of learning and instruction: game-based methods and strategies for training and education.* San Francisco: Pfeiffer, 2012.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY (MIT). 18.404J *Theory of Computation: Fall 2020*. Professor Michael Sipser. Cambridge, 2020. Disponível em: https://ocw.mit.edu/courses/18-404j-theory-of-computation-fall-2020/. Acesso em: 28 out. 2026.

RICO LUGO, Mauricio Javier. *MED Evolución: un camino al pensamiento computacional.* 2016. Dissertação (Mestrado em Informática Educativa) – Universidad de La Sabana, Chía, 2016.

UNIVERSITY OF CANTERBURY. *Computer Science Field Guide: Formal Languages – Regular expressions.* Christchurch, [s. d.]. Disponível em: https://www.csfieldguide.org.nz/en/chapters/formal-languages/regular-expressions/. Acesso em: 28 out. 2026.

WING, Jeannette M. Computational thinking. *Communications of the ACM*, New York, v. 49, n. 3, p. 33–35, 2006. DOI: https://doi.org/10.1145/1118178.1118215.