



UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE FÍSICA

LUCIANE GUARNIERI BRODBECK

**O USO DO MICRO:BIT PARA O ENSINO E A PRÁTICA DO PENSAMENTO
COMPUTACIONAL E DA ROBÓTICA**

MARINGÁ
2024

LUCIANE GUARNIERI BRODBECK

**O USO DO MICRO:BIT PARA O ENSINO E A PRÁTICA DO PENSAMENTO
COMPUTACIONAL E DA ROBÓTICA**

Trabalho de Conclusão de Curso apresentado ao
Departamento de Física como requisito parcial
para obtenção do grau de Licenciado em Física
pela Universidade Estadual de Maringá.

Orientador: Prof. Dr. Breno Ferraz de Oliveira

MARINGÁ

2024

LUCIANE GUARNIERI BRODBECK

**O USO DO MICRO:BIT PARA O ENSINO E A PRÁTICA DO PENSAMENTO
COMPUTACIONAL E DA ROBÓTICA**

Trabalho de Conclusão de Curso apresentado ao
Departamento de Física como requisito parcial
para obtenção do grau de Licenciado em Física
pela Universidade Estadual de Maringá.

BANCA EXAMINADORA

Prof. Dr. André Marino Gonçalves
Universidade Estadual de Maringá

Prof. Dr. Anuar Jose Mincache
Universidade Estadual de Maringá

Prof. Dr. Breno Ferraz de Oliveira (orientador)
Universidade Estadual de Maringá

MARINGÁ
2024

Agradecimentos

Agradeço imensamente ao Prof. Dr. Breno Ferraz de Oliveira por ter aceitado me orientar no desenvolvimento deste trabalho, por sua paciência e prontidão durante todo o processo. Nunca terei palavras suficientes para agradecer-lo por também ter me orientado em um projeto de iniciação científica e por me permitir fazer parte do projeto de extensão trUEM, onde pude me capacitar e desenvolver habilidades muito importantes para o sucesso acadêmico, profissional e pessoal.

Aos meus colegas do trUEM pela grande troca de experiências que pudemos compartilhar.

Aos meus colegas de graduação que me ajudaram nos momentos de dificuldade com os conteúdos do curso.

A todos os professores que me guiaram nessa incrível jornada de aprendizado.

Ao meu marido que sempre me incentivou e apoiou minhas decisões relacionadas a estudo e carreira profissional.

E aos meus pais que me ensinaram que o conhecimento é o maior bem que alguém pode possuir.

Resumo

O uso da tecnologia está cada vez mais presente na vida cotidiana e a educação é uma área que pode muito se beneficiar de recursos tecnológicos como ferramentas para o ensino e a prática de conteúdos escolares. Abordagens de ensino ativo e multidisciplinares, além de promoverem a integração entre as disciplinas e colocarem o aluno como participante ativo do aprendizado, capacitam o aprendiz para resolver problemas da vida real. O pensamento computacional e a robótica podem ser importantes aliados como recursos pedagógicos para a aprendizagem significativa. Assim, o objetivo deste trabalho é construir uma sequência didática para o ensino e a prática do pensamento computacional e da robótica fazendo uso do microcontrolador micro:bit. Para isso, será programado um jogo na plataforma MakeCode e pretende-se que, enquanto o aluno aprende conteúdos de Física, pensamento computacional e robótica em um ambiente lúdico, desenvolva habilidades e competências preconizadas pela Base Nacional Comum Curricular (BNCC). O jogo desenvolvido permite também uma avaliação da capacidade de memória operacional do aluno.

Palavras-chave: Física, ensino STEAM, memória operacional, gamificação e sequência didática.

SUMÁRIO

| | |
|---|-----------|
| Agradecimentos | 4 |
| Resumo | 5 |
| Introdução | 7 |
| Capítulo 1 – O ensino STEAM | 9 |
| Capítulo 2 – Pensamento Computacional e Robótica | 13 |
| 2.1 – Pensamento Computacional | 13 |
| 2.2 – Robótica | 17 |
| Capítulo 3 – Memória Operacional | 18 |
| Capítulo 4 – Gamificação | 21 |
| Capítulo 5 – Conhecendo o micro:bit | 24 |
| Capítulo 6 – Sequência didática | 29 |
| 6.1 - Proposta da sequência didática | 31 |
| 6.1.1 - Primeira aula | 32 |
| 6.1.2 - Segunda aula | 34 |
| Considerações finais | 40 |
| Referências bibliográficas | 42 |
| Apêndice | 45 |
| Anexo A | 48 |
| Anexo B | 49 |
| Anexo C | 50 |
| Anexo D | 51 |
| Anexo E | 52 |

Introdução

O uso de tecnologia tem se tornado cada vez mais presente e os conhecimentos em Computação são tão importantes quanto os conhecimentos básicos de Física, Matemática e Filosofia, dentre outros, para a vida na sociedade contemporânea. Saber contar, abstrair, pensar, relacionar ou medir não são conhecimentos específicos da computação. Eles são fundamentais para que os indivíduos possam resolver os problemas da vida cotidiana.

Ler um livro utilizando um leitor digital, assinar documentos digitalmente, adquirir músicas, filmes, jogos, falar ao telefone por videochamadas, trabalhar colaborativamente em diversos formatos de arquivos, fazer uso de veículos sem motoristas e realizar compras pela Internet são apenas alguns dos exemplos de como a Computação está presente em nossas vidas.

A Computação também impacta nas mais diversas áreas do conhecimento resolvendo problemas complexos de diferentes áreas da ciência que passaram a ser abordados com uma perspectiva computacional com avanços nas estratégias e artefatos desenvolvidos para lidar com a complexidade das mais variadas situações.

Na Biologia, por exemplo, foi possível mapear o genoma humano. Na saúde, pode-se simular a previsão de tempo de contaminação de doenças. Na Química, simulações de reações químicas foram desenvolvidas para que pesquisadores não coloquem suas vidas em risco. Na Matemática foi possível gerar desenhos geométricos complexos. A resistência de estruturas pode ser testada através de *softwares* auxiliando a Arquitetura e a Engenharia. Simuladores de voo e de direção de veículos terrestres servem como recurso preparatório para pilotos e motoristas. Na Física, aceleradores de partículas podem ser simulados permitindo que se realizem diversos testes nas mais variadas configurações.

Podemos observar que praticamente todas as áreas do conhecimento estão sendo impactadas e transformadas pela abordagem computacional. Inclusive, na Lei de Diretrizes e Bases (LDB), em seu artigo 32, afirma que é objetivo da formação básica do cidadão “a compreensão do ambiente natural e social, do sistema político, da tecnologia, das artes e dos valores em que se fundamenta a sociedade” (SOUZA; SILVA, 1997, p. 53).

O impacto dos computadores em quase todos os aspectos de nossas vidas é inegável. Nos dias de hoje, praticamente qualquer serviço ou produto que utilizamos diariamente depende, de forma direta ou indireta, de algum processo ou decisão determinado por um chip de computador. No que diz respeito ao uso de computadores nas salas de aula, equipar as mesmas com dispositivos eletrônicos não garante que haverá melhoria no desempenho dos estudantes, pois uma tecnologia de alta qualidade não pode substituir um ensino fraco. Inserir

o pensamento computacional e a robótica, de maneira obrigatória, no Novo Ensino Médio não resultará no direto desenvolvimento intelectual dos alunos, pois as crianças e os jovens interagem facilmente com as novas tecnologias, mas têm pouca experiência para criar coisas ou se expressar com novas tecnologias. É como dizer que eles conseguem ler, mas não conseguem escrever com as novas tecnologias. É nesse sentido que podemos dizer que ao aprenderem a programar, eles também estão programando para aprender.

O ensino da programação desenvolve a capacidade de pensar de forma crítica, que é pensar com clareza e racionalidade, de maneira reflexiva e independente, e compreender a conexão lógica entre ideias, com pensamento estruturado e trabalho em colaboração, que são elementos essenciais para as mais diversas profissões. Além de ser um fascinante recurso didático pedagógico de grande impacto, o uso da tecnologia como ferramenta de aprendizagem auxilia na forma de estruturar problemas e encontrar soluções para os mesmos, utilizando os fundamentos do Pensamento Computacional.

Este trabalho pretende contribuir para o entendimento da interação existente entre Ciência, Tecnologia, Engenharia, Artes e Matemática, que é a abordagem do ensino ativo chamado STEAM, sua relação com o Pensamento Computacional e a Robótica, que proporcionam o aumento da capacidade de memória operacional dos indivíduos, e sobre os benefícios diversos que a inclusão desse ensino promove gerando uma melhor compreensão do mundo, por meio da integração entre as áreas do conhecimento, e uma maior empregabilidade.

No primeiro capítulo será explanado o que é o ensino STEAM e como ele pode contribuir na participação ativa do aprendizado multidisciplinar. O segundo capítulo discorrerá a respeito do Pensamento Computacional e da Robótica e de como auxiliam no processo de resolução de problemas, explicando os Quatro Pilares do Pensamento Computacional e suas aplicações.

O terceiro capítulo tratará sobre memória operacional e a relação existente entre a mesma e a Robótica, ressaltando os benefícios para a sociedade, educação e indústria. O tema do quarto capítulo é a gamificação, que utiliza elementos de jogos em situações da vida real para criar engajamento e motivação.

No capítulo cinco, o microcontrolador micro:bit será apresentado. E, no capítulo seis, será elaborada uma sequência didática, que pode ser utilizada em aulas de Ciências do Ensino Fundamental, de Física do Novo Ensino Médio e até mesmo no Ensino Superior, utilizando o microcontrolador micro:bit como uma ferramenta de ensino para o desenvolvimento do pensamento computacional e treinamento da memória operacional.

Capítulo 1 – O ensino STEAM

O termo STEAM é um acrônimo em inglês para *Science, Technology, Engineering, Arts and Mathematics* e é uma abordagem de ensino ativo e multidisciplinar que integra as disciplinas de Ciências, Tecnologia, Engenharia, Artes, Matemática e Humanidades tendo a interação entre as disciplinas como ponto central onde o aluno participa ativamente do aprendizado ao ser instigado a encontrar soluções para problemas reais.

Inicialmente, o termo surgiu nos Estados Unidos nos anos 90 com o nome de SMET (Science, Mathematics, Engineering and Technology) e um dos argumentos para sua criação foi que o modelo educacional vigente não atendia à crescente demanda da transformação tecnocientífica que estava ocorrendo nos Estados Unidos e em outros países desenvolvidos. Foi evidenciado, por meio de relatórios realizados por órgãos estadunidenses, que os estudantes não tinham interesse por carreiras relacionadas à área SMET. Para que não houvesse falta de mão de obra e um atraso do país em relação à tecnologia, o governo iniciou grandes investimentos em programas educacionais ligados a essas áreas do conhecimento.

Em 2001, o termo SMET foi reorganizado para STEM e sua abordagem ganhou visibilidade para o restante do mundo. Foi também nesse ano que ele foi oficialmente reconhecido pela National Science Foundation (NSF) que é um órgão norte americano que tem por finalidade promover o avanço científico, visando impulsionar projetos para desenvolver habilidades de ciências nos estudantes e estimular o ingresso em carreiras associadas à tecnologia e à engenharia (DUGGER JUNIOR, 2010).

Em 2008, houve a inclusão de *Arts* (Artes) ao acrônimo que passou a ser denominado STEAM. Em sua tese *STEAM Education: an overview of creating a model of integrative education*, a pesquisadora Georgette Yakman, do Virginia Polytechnic Institute and State University, explica a importância da inclusão das Artes para incorporar a contribuição das áreas das humanidades e do design criativo para processos de criação de ideias inovadoras.

Por sugerir uma articulação das áreas que compõem seu acrônimo, a abordagem STEAM propõe a resolução de problemas reais utilizando os conhecimentos aplicados no processo. As situações características dessa abordagem são o levantamento de hipóteses, definição de estratégias, debate e criação de sequências para a resolução do problema. Nessa perspectiva, o Pensamento Computacional pode ser um importante aliado dessa abordagem, pois na etapa de resolução de problemas há o desenvolvimento do pensamento lógico, ou seja, de uma sequência lógica de procedimentos que culminam em um resultado ou na resolução de

um problema.

A abordagem STEAM não se restringe a conhecimentos do campo cognitivo, ou seja, não se limita ao pensar, aprender, raciocinar, lembrar e prestar atenção. O movimento de integração entre áreas distintas permite uma compreensão mais completa do objeto de estudo, uma percepção mais clara do mundo e o entendimento da realidade a partir de uma base científica sólida. É nesse cruzamento que se aglutinam habilidades, atitudes, valores e os saberes do desenvolvimento socioemocional: significados e sentidos que o próprio aluno constrói, colaborativamente (SENAC, 2021).

O pesquisador Gustavo Pugliesi, que estuda STEAM no Brasil, na Inglaterra e nos EUA, aponta o ensino STEAM como “algo maior”, no sentido de ser política econômica, política educacional, demanda da indústria e, também, parte de uma tendência maior, global, de incentivo a Ciências, Tecnologia, Engenharia, Artes e Matemática.

É importante notar que, enquanto política pública, acaba englobando as outras duas representações: atua-se modificando o currículo e a metodologia para atingir determinados fins. (PUGLIESE, 2020, p. 221)

STEAM pode ser denominado como tendência educacional, organização curricular, técnica ou metodologia (SENAC, 2021). Ao estabelecer como diferencial uma experiência de aprendizagem cada vez mais interdisciplinar, com foco em levar o aluno a exercitar habilidades diversas, o STEAM vai ao encontro das propostas indicadas na Base Nacional Comum Curricular (BNCC, 2018). O documento estabelece as competências gerais, que traduzem uma perspectiva na qual as escolas devem, além de promover o desenvolvimento intelectual do aluno, desenvolver também o social, o físico, o emocional e o cultural. Uma das competências gerais, por exemplo, é a argumentação, em que a construção está conectada com todos os componentes curriculares.

Na educação brasileira, o movimento STEAM pode ser percebido por meio de programas educacionais nas escolas públicas, que são incentivadas por organizações não-governamentais, empresas de educação com produtos voltados para atividades STEM e as escolas privadas que utilizam atividades STEM/STEAM (PUGLIESE, 2020). As ideias STEAM também se apresentam no Brasil por meio do Programa Nacional do Livro Didático e do Material Didático 2021, no Novo Ensino Médio.

Enquanto proposta metodológica, alguns trabalhos sobre STEAM publicados nos

últimos cinco anos, utilizam também a teoria da Aprendizagem Significativa. Um exemplo é o trabalho com o tema de Tomografia Computadorizada:

Dourado (2019) apresenta uma proposta com o tema “Tomografia computadorizada”, utilizando a teoria de Aprendizagem Significativa de Ausubel. A autora dividiu as atividades em quatro momentos, utilizando a interdisciplinaridade STEAM nas aulas de Física e Biologia do ensino médio. (CAMPOS *et al*, 2022)

Como metodologia de ensino e aprendizagem, alguns trabalhos analisados abordaram propostas metodológicas aplicadas em níveis de ensino diferentes. No Ensino Médio,

Albuquerque (2021) trabalhou com robótica educacional utilizando a aprendizagem baseada em projetos e a metodologia STEAM, com o objetivo de avaliar o ensino-aprendizagem dos alunos do primeiro ano do curso médio técnico em informática. A autora utilizou o uso do micro:bit para elaboração dos projetos e realizou um minicurso com os alunos, onde obteve resultados satisfatórios com a junção dessa ferramenta com as metodologias adotadas. Concluindo que os alunos estavam estimulados e motivados a criatividade, contribuindo para a inserção do pensamento e do letramento computacional. (CAMPOS *et al*, 2022)

Em trabalhos realizados com alunos do Ensino Fundamental II, o uso de jogos como ferramenta didática, no âmbito das metodologias ativas, resultou em um aumento da aprendizagem ativa dos alunos de maneira lúdica favorecendo a aprendizagem formal.

No Ensino Superior, para ensinar o conteúdo de Termodinâmica para alunos do curso de Licenciatura em Física de uma Universidade Federal, foi utilizada a plataforma de prototipagem eletrônica Arduino juntamente com a educação STEAM.

De acordo com Silva (2021), a educação STEAM proporcionou a participação ativa dos alunos, “atingindo e fomentando um alto nível de protagonismo dos participantes” (p. 100), desenvolvendo a interação e comunicação entre o professor/aluno e entre aluno/aluno. (CAMPOS *et al*, 2022)

No cenário brasileiro, grande parte dos trabalhos publicados nos principais meios de divulgação científica utilizam a abordagem STEAM como metodologia de ensino, baseada em

metodologias ativas, colocando o aluno como protagonista do processo de aprendizagem. Mas não se trata de uma receita pronta para melhorar as aulas. Há uma carência de pesquisas que buscam compreender as contribuições e o impacto no sistema educacional como um todo. Para Pugliese (2020) “as interpretações de tal tendência estão longe de serem esgotadas, principalmente do ponto de vista da pesquisa educacional em ensino de ciências” (p. 219).

Capítulo 2 – Pensamento Computacional e Robótica

2.1 – Pensamento Computacional

O pensamento computacional não se relaciona apenas à tecnologia ou à programação. Essa habilidade propõe que os indivíduos sejam capazes de identificar problemas, dividi-los em partes menores, focar nos processos importantes em vez de priorizar os detalhes, reconhecer os padrões e criar passos para encontrar soluções com criatividade utilizando outros tipos de conhecimento (BRACKMANN, 2017). O ensino e a prática do pensamento computacional e da robótica promovem o desenvolvimento sociocognitivo, a autonomia, fomenta o conhecimento interdisciplinar e proporciona o aumento da capacidade da memória operacional (abordaremos este assunto no Capítulo 3) que é composta por um conjunto de processos cognitivos que envolvem atividades como a compreensão da linguagem, leitura, aritmética e resolução de problemas (AQUINO; BORGES, 2019).

O pensamento computacional ganhou maior relevância com a publicação das Normas sobre Computação na Educação Básica, na forma de complemento à BNCC (Brasil, 2022). A partir desse complemento, essa habilidade passa a ocupar um espaço próprio no currículo, com a determinação de competências, objetivos de aprendizagem e objetos de conhecimento a serem desenvolvidos ao longo de toda a Educação Básica. Na etapa da educação infantil,

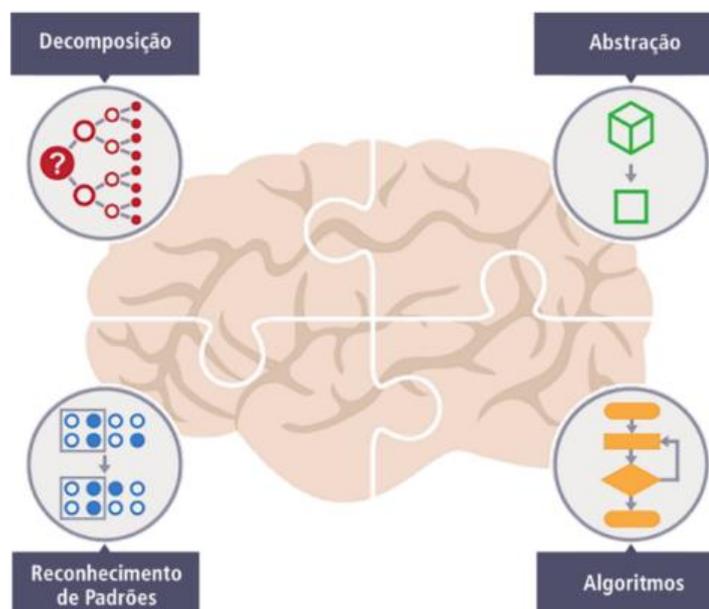
A Computação permite explorar e vivenciar experiências, sempre movidas pela ludicidade por meio da interação com seus pares. Estas experiências se relacionam com diversos dos campos de experiência da Educação Infantil e devem considerar as seguintes premissas.

1. Desenvolver o reconhecimento e a identificação de padrões, construindo conjuntos de objetos com base em diferentes critérios como: quantidade, forma, tamanho, cor e comportamento.
2. Vivenciar e identificar diferentes formas de interação mediada por artefatos computacionais.
3. Criar e testar algoritmos brincando com objetos do ambiente e com movimentos do corpo de maneira individual ou em grupo.
4. Solucionar problemas decompondo-os em partes menores identificando passos, etapas ou ciclos que se repetem e que podem ser generalizadas ou reutilizadas para outros problemas. (BNCC,

Essas premissas vão ao encontro das quatro dimensões do Pensamento Computacional, também conhecidas como os Quatro Pilares do Pensamento Computacional, que são a decomposição, o reconhecimento de padrões, a abstração e os algoritmos.

Decompor um problema significa quebrá-lo em partes, ou seja, dividir um problema ou sistema complexo em partes menores para que se torne mais fácil de entender. Essa prática aumenta a atenção aos detalhes permitindo que haja uma identificação mais nítida de cada uma das partes e suas funcionalidades. Os programadores utilizam essa técnica para dividir um código-fonte em pedaços menores para facilitar sua compreensão e manutenção. Isso possibilita resolver problemas complexos de forma mais simples, facilita a compreensão de novas situações e possibilita projetar sistemas de grande porte (BRACKMANN, 2017).

Figura 2.1: Os quatro pilares do Pensamento Computacional.



Fonte: BRACKMANN, 2017.

Padrões são similaridades ou características entre objetos, situações ou problemas. Quando se realiza a decomposição de um problema é possível encontrar padrões entre as partes menores. O Reconhecimento de Padrões é uma forma de resolver problemas rapidamente fazendo uso de soluções previamente definidas em outros problemas com base em experiências anteriores (BRACKMANN, 2017).

A Abstração envolve a filtragem dos dados e sua classificação. Neste pilar, os elementos

que não são necessários devem ser ignorados para que se possa concentrar nos que são relevantes e criar uma representação do que precisa ser resolvido. Histórias infantis que envolvem atividades matemáticas são um exemplo de abstração, pois é necessário que a abstração das informações importantes da história ocorra para que a resolução do problema matemático aconteça.

Os Algoritmos podem ser comparados a receitas de bolo, isto é, um conjunto de passos específicos usado para solucionar um problema. São soluções prontas, pois já passaram pelo processo de decomposição, abstração e reconhecimento de padrões para sua formulação. Na linguagem de programação, antes de serem escritos códigos, as instruções são descritas e ordenadas em formato de diagramas para se atingir um objetivo. Podemos chamar um algoritmo de núcleo principal porque possui uma grande abrangência em diversos momentos das atividades propostas pelo Pensamento Computacional. Um exemplo de algoritmo é a atividade de realizar uma soma ensinada nas aulas de Matemática. Há uma sequência de passos necessários para atingir o resultado, como somar primeiro as unidades, depois as dezenas, as centenas e assim por diante. Seguindo as regras, os alunos, assim como os computadores, podem resolver contas com quaisquer números. Portanto, a principal característica do algoritmo é a possibilidade de automação das soluções (BRACKMANN, 2017).

A Computação na Educação Básica envolve três eixos: Pensamento Computacional, Mundo Digital e Cultura Digital. Na etapa da Educação Infantil, um dos objetivos de aprendizagem do eixo Pensamento Computacional é experienciar a execução de algoritmos. Exemplos de atividades de computação desplugada (que não necessitam de computadores ou outros dispositivos eletrônicos) são jogos de labirinto, amarelinha, sequência de números e sequência de cores.

As competências da etapa do Ensino Fundamental envolvem compreender a Computação como uma área de conhecimento para explicar o mundo atual, reconhecer o impacto dos artefatos computacionais e os desafios da sociedade, utilizar diferentes linguagens e tecnologias, utilizar técnicas da Computação para identificar e resolver problemas, avaliar soluções, desenvolver projetos, automatizar processos, agir com respeito, autonomia, responsabilidade, flexibilidade, resiliência e determinação (BNCC, Computação, 2022).

Na Computação para o terceiro ano do ensino fundamental, a decomposição como objeto de conhecimento do eixo Pensamento Computacional está relacionada a uma atividade de criar uma receita (algoritmo) que descreve a tarefa (problema) de preparar o café da manhã. Há ainda um subproblema na divisão da atividade em duas etapas: preparar o café e fazer um sanduíche.

No quinto ano, a lógica computacional é o objeto de conhecimento da habilidade de realizar operações de negação (não), conjunção (e) e disjunção (ou) sobre sentenças lógicas e valores “verdadeiro” e “falso”.

O professor pode apresentar diferentes lógicas e solicitar que os alunos determinem seus valores verdade, como por exemplo:

Cinco é maior que seis. (Falso)

Cinco NÃO é maior que seis. (Verdadeiro).

Cinco é maior que seis E maior que dois. (Falso)

Cinco é maior que seis OU maior que dez. (Falso)

Cinco é maior que seis OU maior que dois. (Verdadeiro)

(BNCC, Computação, 2022, p. 29)

A Programação é objeto de conhecimento do Pensamento Computacional do nono ano. Neste eixo deve ser trabalhada a habilidade de construir soluções computacionais de problemas de diferentes áreas do conhecimento. Exemplos dessa habilidade são a criação de um programa que manipula imagens, para manipular os pixels dessa imagem a partir de um vetor ou uma matriz, um jogo para armazenar a pontuação dos usuários numa lista e salvar esses dados na nuvem, dentre outros.

Na etapa do Ensino Médio, não há mais a divisão por eixos e as competências estão relacionadas com a compreensão das possibilidades e dos limites da Computação para resolver problemas, análise crítica de artefatos computacionais, análise de situações do mundo contemporâneo, construção do conhecimento usando técnicas e tecnologias computacionais, desenvolvimento de projetos, construção de soluções, tomada de decisões éticas, democráticas e socialmente responsáveis, compartilhamento de informações e de ideias utilizando diferentes plataformas, agindo coletivamente com respeito, autonomia, responsabilidade, flexibilidade, resiliência e determinação.

Na competência específica de construir conhecimento usando técnicas e tecnologias computacionais, há a habilidade de desenvolvimento de projetos com robótica, utilizando artefatos físicos ou simuladores. Exemplos dessa habilidade é o uso do Arduino IDE (Integrated Development Environment)¹ ou da plataforma gratuita de *software* MakeCode, que é a utilizada pelo micro:bit.

¹ Ambiente de desenvolvimento integrado que funciona como uma ferramenta de desenvolvimento para editar o código, acessar um terminal, executar um *script*, depurar e compilar usando um único ambiente.

2.2 – Robótica

A Robótica é um ramo da ciência baseado na construção de estruturas mecânicas autômatas multifuncionais reprogramáveis, operadas por meio de circuitos integrados. É uma área em constante expansão e considerada multidisciplinar, pois nela são aplicados conhecimentos de microeletrônica, engenharia mecânica, física cinemática, matemática, inteligência artificial e outras ciências. O uso da Robótica na educação une as teorias Construcionista de Seymour Papert e Construtivista de Jean Piaget ao uso do computador na educação. Papert (1993), complementando Piaget (1974), adiciona que a manipulação de objetos é chave para as crianças construírem seu próprio conhecimento e que essa construção se dá de forma mais efetiva quando o aprendiz se engaja de maneira consciente na construção de algo tangível.

Juntamente com o Pensamento Computacional, a Robótica oportuniza situações de aprendizagem pela resolução de problemas, que podem ser simples ou complexos, permitindo que os alunos trabalhem uma grande diversidade de competências e habilidades por englobar diferentes áreas do conhecimento. No processo de ensino e aprendizagem, assim como na robótica, há o estímulo da colaboração, a habilidade de trabalhar em grupo, a elaboração de estratégias para a solução de problemas e o desenvolvimento do pensamento formal (realização de ações no plano das ideias sem necessitar de manipulação ou referências concretas), que constitui um sistema complexo, mas coerente, relativamente diferente da lógica da criança e a essência da lógica dos adultos que é a base das formas elementares do pensamento científico (PIAGET, 1972).

Na habilidade de Programação do eixo Pensamento Computacional do nono ano que prevê o uso de autômatos para descrever comportamentos de forma abstrata automatizando-os através de uma linguagem de programação baseada em eventos, o exemplo dado é a modelagem do comportamento de um robô que descreve eventos acionados a partir da leitura de seus sensores (BNCC, Computação, 2022).

Robôs costumam instigar a curiosidade de crianças e adultos e, por esse motivo, a Robótica torna-se uma ferramenta de aprendizagem capaz de oferecer atividades práticas e divertidas que alimentam o interesse e a curiosidade dos alunos.

Capítulo 3 – Memória Operacional

A memória operacional é uma função do nosso cérebro que consiste na capacidade de lembrar e processar informações ao mesmo tempo. Seria como a memória RAM² de um computador ou como um caderno de anotações no qual armazenamos informações e as acessamos quando precisamos delas.

Um bom exemplo de uso da memória operacional é quando precisamos realizar uma operação de multiplicação sem usar lápis, papel ou calculadora.

Tente multiplicar 56 por 17 sem usar lápis e papel, ou calculadora. Primeiro você tem que manter os dois números 56 e 17 na sua memória operacional, depois então você precisa aplicar regras matemáticas previamente aprendidas (por exemplo, 56 vezes 10 e 56 vezes 7) que lhe trarão novos números, os quais novamente você precisará manter na sua memória operacional até que enfim seja apto a encontrar a resposta correta. (ABREU; TOURINHO, 2011, p. 9)

Sem a memória operacional não seríamos capazes de resolver problemas matemáticos. Mas essa função não está associada somente a esse tipo de problema. Armazenar letras, significado de palavras e saber usá-las em uma frase também requer o uso da memória operacional. O ditado em sala de aula é um exemplo em que as crianças devem se lembrar das palavras ditadas (armazenamento) e descobrir como cada palavra deve ser escrita (processamento). Em frases longas, pessoas com baixa capacidade de memória operacional, conseguem escrever as três primeiras palavras e apresentam dificuldade para terminar a frase porque se esquecem do restante, mesmo sabendo escrever todas as palavras ditas. Os autores Abreu e Tourinho (2011) relatam em seus trabalhos que estudos mostram que a memória operacional parece influenciar mais a habilidade de leitura, escrita e matemática.

Segundo especialistas, citados por Abreu e Tourinho, é na infância que a capacidade da memória operacional se desenvolve. Ela é gradualmente aumentada ao longo dos anos e atinge seu ponto máximo na adolescência. A capacidade de memória operacional tende a diminuir com o envelhecimento. Sendo assim, a evolução temporal da memória operacional pode ser bem representada por uma curva em forma de U invertida.

Conforme citam Abreu e Tourinho, pesquisas mostram que crianças com alta

² *Random Access Memory*, ou Memória de Acesso Randômico, é um tipo de memória que armazena de maneira temporária informações que computadores precisam durante o uso de programas.

capacidade de memória operacional têm melhor desempenho escolar.

Não é raro que problemas de memória operacional sejam confundidos com problemas de atenção por muitos professores. Trabalhos conduzidos no Reino Unido em 2008 por Gathercole e Alloway, identificaram alguns sinais que crianças com dificuldades de memória operacional apresentam:

- Crianças reservadas em atividades em sala de aula, mas que geralmente são bem integradas socialmente;
- Comumente se comportam como se não estivessem prestando atenção no que o professor está ensinando;
- Crianças que esquecem instruções dadas ou palavras em uma frase;
- Dificuldades em exercícios de memorização e processamento de informações simultâneos, como ditados;
- Confusão ao se depararem com exercícios mais difíceis, o que leva ao abandono da atividade;
- Crianças com dificuldades específicas em matemática, leitura e escrita;
- Crianças que em atividades de escrita frequentemente repetem palavras ou letras, ou as deixam de fora completamente.

Para ajudar crianças com baixa memória operacional, existem várias estratégias desenvolvidas por especialistas em psicologia e neuropsicologia que podem ser aplicadas em sala de aula. A proposta deste trabalho é desenvolver uma sequência didática, aplicando as pesquisas realizadas sobre memória operacional, pensamento computacional e robótica, utilizando o micro:bit.

Mas antes da construção da sequência didática, é importante apontar que alguns dos exemplos indicados como recursos de auxílio se assemelham aos pilares do pensamento computacional. Quando a criança encontra dificuldade em exercícios considerados difíceis ou complexos, o professor pode aplicar o conceito da decomposição para dividir o problema em partes menores. O mesmo pode ser feito quando a criança apresenta dificuldade em se lembrar das instruções. A sequência pode ser pensada como um algoritmo em que a atividade é realizada por meio de etapas.

Encorajar a criança a pedir informações adicionais pode ser visto como uma forma de reconhecer padrões, pois as perguntas podem levá-la a identificar semelhanças com conhecimentos adquiridos anteriormente e que podem ser aplicados na realização da atividade que está sendo proposta. O uso de palavras fáceis e material familiar às crianças também auxilia

na resolução do problema.

Quando uma criança está sentada quieta sem resolver os exercícios que ela deveria estar resolvendo, isso pode ser um sinal de sobrecarga da memória operacional. Uma maneira de ajudar a criança neste momento é perguntar o que ela estava fazendo e direcionar a próxima coisa que deve ser feita utilizando a abstração. O professor pode indicar os elementos mais importantes do exercício, fazendo uma filtragem dos dados com a criança, e direcioná-la para continuar com a resolução.

Sabe-se que o cérebro humano possui uma grande flexibilidade e plasticidade, ou seja, é capaz de se modificar de acordo com a necessidade, estímulo e ambiente. Sendo assim, o cérebro precisa ser treinado para funcionar de maneira ótima.

Algumas pesquisas bastante recentes mostraram que alguns programas de computador, cujo maior objetivo é o treinamento cognitivo, podem melhorar a memória operacional (veja nos artigos de Klingberg, Fernell, Olesen et al. 2005). Essa é uma fantástica descoberta que deve gerar importantes implicações quando se trata de ajudar crianças com dificuldades de memória operacional. (ABREU; TOURINHO, 2011, p. 27)

Um estudo realizado com 30 crianças do ensino fundamental do município de Apucarana, no estado do Paraná, em 2017, publicado nos anais do III Congresso Multidisciplinar / XI Fórum Científico da Faculdade de Apucarana, avaliou os resultados de um programa de treinamento da memória operacional. As crianças, entre 9 e 13 anos, que participaram do treinamento apresentavam dificuldades de aprendizagem.

O resultado do treinamento foi muito positivo e as crianças apresentaram melhoras significativas dos índices avaliados pelo programa em diferentes níveis.

Na análise de dados podemos observar que os elementos analisados podem ser reproduzidos com a utilização do micro:bit. Para o desafio de matemática, por exemplo, é possível programar o micro:bit para realizar testes de aritmética simples. Também podemos programá-lo para o exercício de seguir instruções, em que o aluno vê uma imagem e deve reproduzi-la nos LEDs do micro:bit.

A sequência didática deste trabalho será a atividade de seguir instruções com atividades de programação e da brincadeira com o micro:bit programado. Mas antes disso, precisamos entender como elementos de jogos podem auxiliar na aprendizagem, conhecer o micro:bit e como iremos usá-lo para esta finalidade.

Capítulo 4 – Gamificação

A gamificação, traduzida da palavra *gamification* da língua inglesa, pode ser definida como elementos de jogos - narrativa, *feedback*, cooperação, pontuação, entre outros - utilizados em contextos da vida real. A função dos elementos dos jogos é aumentar a motivação dos indivíduos em relação às atividades que são realizadas na vida cotidiana (MURR; FERRARI, 2020).

É importante observar que gamificação e aprendizagem baseada em jogos são conceitos diferentes. Enquanto a gamificação é uma tentativa de transformar todo um processo em uma experiência com a introdução de elementos dos jogos sem, necessariamente, utilizar jogos para este fim, a aprendizagem baseada em jogos se utiliza de um jogo para melhorar uma experiência de aprendizado específica.

Quando a gamificação é utilizada, há a sensação de que se está jogando, mas, na verdade, um conceito está sendo estudado, um trabalho está sendo realizado, produtos estão sendo comprados, uma marca está se tornando conhecida e lembrada, etc.

Um exemplo prático da vida real, no qual podemos utilizar a gamificação, é a divisão das tarefas domésticas. Pode ser proposto um jogo em que todos cooperem, cujo desafio é completar determinadas atividades em um certo tempo. Para as tarefas realizadas, deve haver um sistema de pontuação em um quadro, com estabelecimento de classificações, mudança de colocação e premiações.

No processo de gamificação os elementos devem estar interconectados, como ilustrado na Figura 4.1, para que a experiência seja a de um jogo completo. Citado por Silva, Sales e Castro (2020), McGonigal (2011), destaca quatro elementos fundamentais em qualquer jogo: voluntariedade, regras, objetivos e *feedbacks*. A voluntariedade implica na aceitação das regras, objetivos e *feedbacks*. O objetivo é o que direciona o participante a se concentrar para atingir o propósito. As regras limitam as ações e condicionam a realização do jogo. Os *feedbacks* informam aos jogadores sua relação com os diferentes aspectos que regulam sua interação para fomentar a motivação.

Figura 4.1 - Representação esquemática dos elementos de jogos interconectados.



Fonte: SILVA, SALES e CASTRO (2020).

No contexto da educação, a gamificação tem um grande potencial. Quando o indivíduo joga, ele se deixa levar pela atitude lúdica e, estando em um processo de aprendizagem, a educação pode se beneficiar muito desse quadro. Para Alves, Minho e Diniz (2014):

A gamificação pode promover a aprendizagem porque muitos de seus elementos são baseados em técnicas que os designers instrucionais e professores vêm usando há muito tempo. Características como distribuir pontuações para atividades, apresentar *feedback* e encorajar a colaboração em projetos são as metas de muitos planos pedagógicos. A diferença é que a gamificação provê uma camada mais explícita de interesse e um método para costurar esses elementos de forma a alcançar a similaridade com os games, o que resulta em uma linguagem a qual os indivíduos inseridos na cultura digital estão mais acostumados e, como resultado, conseguem alcançar essas metas de forma aparentemente mais eficiente e agradável.

Para se transpor a gamificação para a educação, o processo de engajamento no jogo deve ser voluntário e colaborativo para se gerar uma competição saudável entre os participantes. Dar ênfase muito grande na recompensa e competitividade pode gerar um efeito nocivo para a aprendizagem, e isso é apontado por muitos autores como uma desvantagem da gamificação na educação.

Motivar os alunos a jogarem e a continuarem jogando exige que os *feedbacks* sejam

constantes e que haja oportunidade de se realizar as mesmas atividades várias vezes para a superação de obstáculos para alcançar os objetivos. O processo de tentar novamente, refazer, superar uma pontuação envolve aprender a se relacionar com o fracasso de forma positiva, sendo extremamente importante para o próprio processo de aprendizado.

O aspecto social e a cooperação são elementos muito importantes em vários jogos e podem ser bem aproveitados na educação. A exploração, pesquisa, curiosidade, autossuperação, construção (e não a destruição) são outras características que podem ser usadas de maneira positiva, de modo a potencializar o aprendizado e motivar os estudantes (MURR; FERRARI, 2020).

Para complementar os elementos da gamificação, o autor Kevin Werbach (2012) aborda três conceitos importantes para a construção de um projeto gamificado: a dinâmica, a mecânica e os componentes. Na dinâmica está a estrutura do processo, onde existe uma narrativa, a progressão, as emoções e o relacionamento no jogo. As ações para a movimentação do jogo se constituem na mecânica, onde se encontram os desafios, a sorte, a cooperação e competição, o *feedback*, a aquisição de recursos, as recompensas, as transações, os turnos e os estados de vitória. Os componentes contribuem para que dinâmica e mecânica ocorram com a utilização de avatares, coleções, desbloqueio de conteúdos, placar, níveis, pontos e investigação ou exploração.

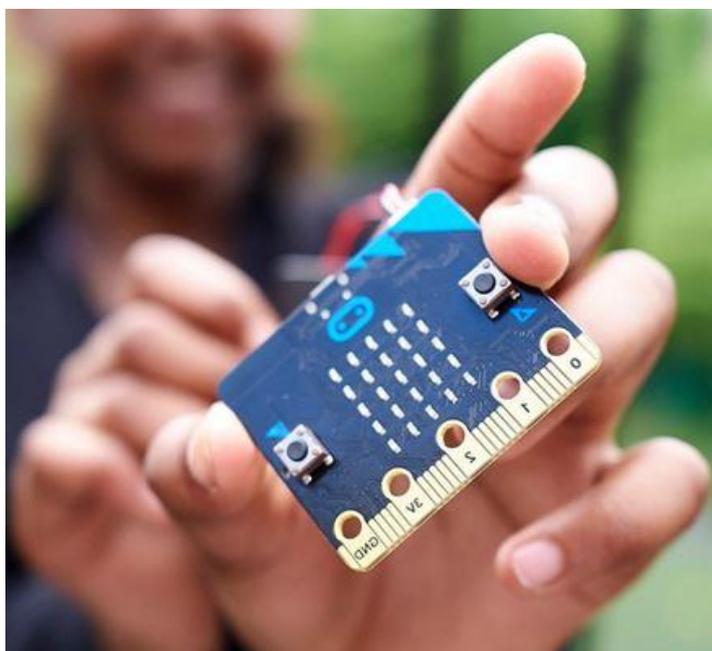
Como estratégia de ensino, a gamificação pode ser considerada como um exemplo de ensino de metodologia ativa, pois o professor, por meio de regras claras, desafia e estimula os alunos a realizarem suas missões. Para cumprir cada missão, os alunos precisam explorar o problema, levantar hipóteses, utilizar seus conhecimentos para tentar solucionar o problema, identificar o que é preciso conhecer a mais, determinar como estudar, compartilhar o novo conhecimento, aplicá-lo e avaliar a solução.

A proposta de sequência didática deste trabalho utiliza os elementos principais da gamificação como uma estratégia de aprendizagem ativa. Conheceremos a seguir o recurso tecnológico utilizado nesta proposta.

Capítulo 5 – Conhecendo o micro:bit

O micro:bit é um *hardware* livre, que foi criado em 2015 pela empresa BBC (*British Broadcasting Corporation*) em parceria com diversas organizações e tem revolucionado a área de ensino de robótica (VOŠTINÁR; KVAŠŠAYOVÁ, 2019). A facilidade no manuseio e na programação deste microcontrolador são atrativos para se fazer o uso no ensino e aplicação da robótica. Criado para ser um auxiliar no ensino de programação e tecnologia para crianças e jovens, o micro:bit, Figura 5.1, ajuda no desenvolvimento do raciocínio lógico, criatividade e resolução de problemas.

Figura 5.1: Microprocessador micro:bit.



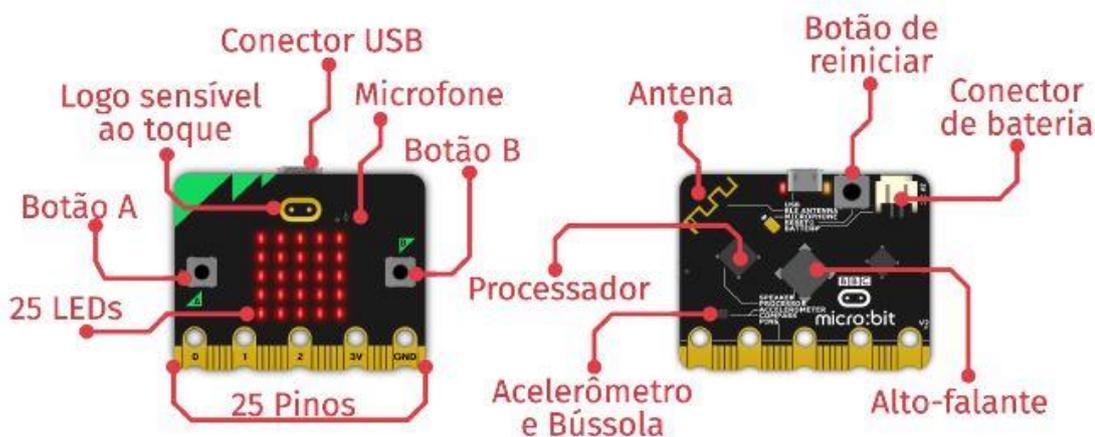
Fonte: <https://microbit.org/>.

O microprocessador micro:bit possui muitos recursos para o ensino e a aprendizagem do pensamento computacional e da robótica. Na parte da frente da placa de circuitos, há 25 LEDs (diodos emissores de luz), dispostos em uma matriz 5 x 5, que podem ser programados para acender e formar letras, números e figuras. Eles também apresentam a função de sensores de luminosidade. Ainda na parte da frente do micro:bit, existe um microfone, dois botões físicos e um botão sensível ao toque.

Na parte inferior da placa há 25 pinos que permitem a conexão com componentes elétricos como motores, sensores, atuadores, e outros dispositivos externos.

No verso da placa é possível ver vários dos componentes que compõem o micro:bit, veja a Figura 5.2. Há uma antena de rádio, uma antena BLE (*Bluetooth Low Energy*), acelerômetro, bússola, alto-falante (*speaker*), giroscópio, sensor de temperatura, magnetômetros e diversos componentes eletrônicos responsáveis pelo funcionamento destes sensores e atuadores.

Figura 5.2 - Componentes presentes no micro:bit.



Fonte: Aprendendo com micro:bit BBC, 2023, p. 10.

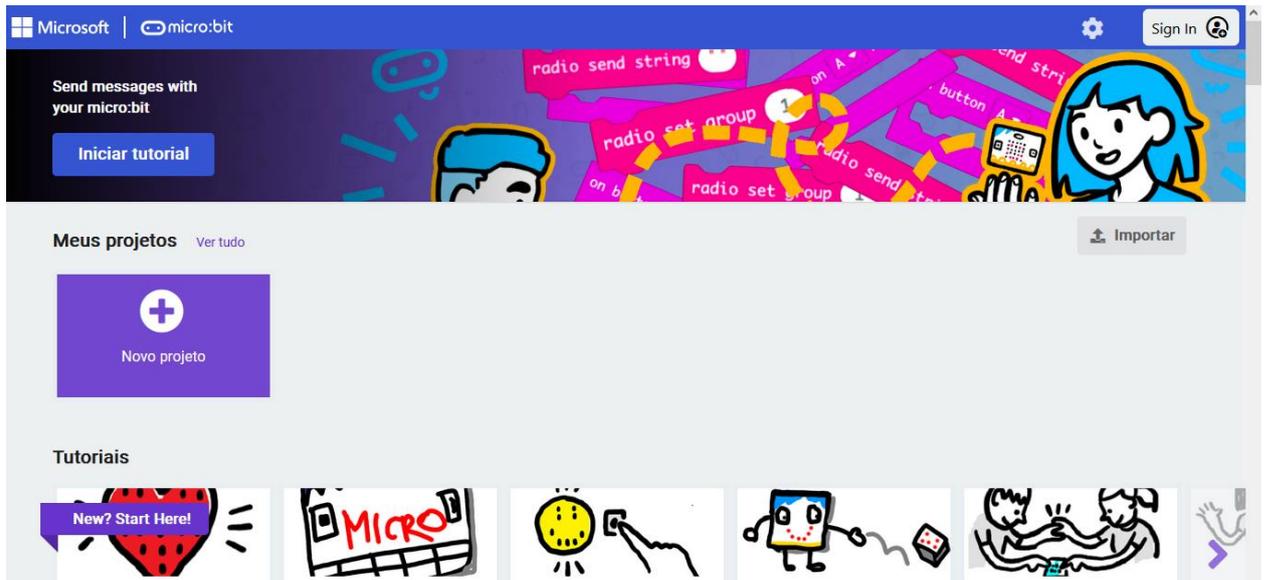
O que torna o micro:bit um grande aliado no ensino do pensamento computacional e da robótica é a versatilidade de atividades que podem ser criadas com ele, desde as mais simples, como acender e apagar um LED, até as mais complexas, como programar vários tipos de jogos que marcam o tempo e a pontuação do jogador.

Outro ponto importante é a possibilidade de utilizar três tipos de linguagem de programação (Blocos, Python e JavaScript) o que torna o micro:bit um dispositivo que pode ser utilizado por pessoas de várias idades e níveis de conhecimento em programação. Para a sequência didática deste trabalho usaremos a programação em Blocos que será feita na plataforma Microsoft MakeCode que é uma plataforma gratuita de *software* livre.

A plataforma MakeCode, Figura 5.3, está disponível para acesso *online* por meio do endereço eletrônico <https://makecode.microbit.org/>. Há também a versão de aplicativo que pode ser transferida para o computador e usada *offline*, disponível em <https://makecode.microbit.org/offline-app>. E existem ainda aplicativos para *smartphones* e *tablets*, para os sistemas operacionais iOS e Android. Todos os aplicativos são gratuitos.

Além de criar experiências envolventes de aprendizagem, a programação em Blocos proporciona um caminho de progressão para a programação do mundo real.

Figura 5.3 - A plataforma Microsoft MakeCode.



Fonte: <https://makecode.microbit.org/>

A página inicial do MakeCode é bastante intuitiva e conta com tutoriais para auxiliar na criação de projetos com o micro:bit. Os tutoriais estão disponíveis em vídeos e em instruções passo a passo e os projetos podem ser escritos nas três linguagens de programação mencionadas anteriormente.

No tutorial de jogos, há opções bem conhecidas das crianças e jovens (e adultos também) como o Pedra, Papel, Tesoura, o Cara ou Coroa e a famosa Batata Quente. Os projetos de jogos para dois micro:bits ou mais, que utilizam a função rádio por meio da antena da placa, são instigantes e desafiam a criatividade dos usuários. Projetos na seção chamada de *Fashion*, ensinam como criar relógios e contadores de passos, por exemplo. Existem ainda projetos de música, brinquedos, ciência e ferramentas.

Os vários projetos disponíveis funcionam como um ponto de partida para as inúmeras atividades que podem ser desenvolvidas com o micro:bit. Para iniciar um projeto do zero, o usuário deve acessar o ícone Novo Projeto para começar a programar. Basta escolher um nome para o projeto e clicar em Criar.

A Figura 5.4 mostra a interface inicial da plataforma MakeCode. Do lado esquerdo identificado com o número 1, fica o simulador da placa física. Caso o usuário não possua um micro:bit, a simulação pode ser projetada em tela cheia e os jogos podem ser jogados por meio de cliques nos botões ou na própria tela, se a tela for sensível ao toque, tornando a brincadeira mais interativa.

Figura 5.4 - A plataforma MakeCode para programação em Blocos.



Fonte: Aprendendo com micro:bit BBC, 2023, p. 14.

A coluna do meio é a biblioteca de funções, ou seja, uma coleção de subprogramas usados no desenvolvimento de *softwares* e que tem a função de facilitar a programação, garantindo mais agilidade e menos erros.

O espaço de edição, identificado pelo número 3 na Figura 5.4, é onde o programa será escrito. É neste espaço que são utilizadas as funções da biblioteca, disponíveis na forma de blocos, que são montados para criar os comandos que o micro:bit executará.

Quando o código estiver finalizado, ele pode ser transferido para a placa física. O procedimento consiste em conectar a placa a um computador utilizando um cabo micro USB, como na Figura 5.5. Na interface do MakeCode há a opção de Baixar, no canto inferior esquerdo, e é só seguir os passos que aparecerão para que o processo seja concluído.

Para ser utilizado, o micro:bit necessita de uma fonte de alimentação. Ele pode ser conectado a um computador, pelo mesmo cabo USB utilizado para baixar o programa, ou ligado a um suporte para duas pilhas AAA.

Figura 5.5: Conectando o micro:bit ao computador.



Fonte: Aprendendo com micro:bit BBC, 2023, p. 15.

O micro:bit é um recurso tecnológico multifuncional que pretende auxiliar, de modo muito significativo, o processo de ensino de programação, de tecnologia, de desenvolvimento de raciocínio lógico, de criatividade e de resolução de problemas. Para o ensino dessa nova tecnologia, como principal ferramenta pedagógica prática do pensamento computacional e da robótica, é necessário estruturar uma aula e uma forma de se fazer isso é por meio de uma sequência didática que será assunto do próximo capítulo.

Capítulo 6 – Sequência didática

Sequências didáticas são um conjunto de atividades ordenadas, estruturadas e articuladas para a realização de certos objetivos educacionais, que têm um princípio e um fim conhecidos tanto pelos professores como pelos alunos (ZABALA, 1998). Para compreendermos o valor educacional e as razões que justifiquem a elaboração de uma sequência didática são necessários identificar as fases, as atividades que a conformam e as relações estabelecidas.

As fases do modelo tradicional implicam em atividade motivadora, explicação de perguntas ou problemas, respostas intuitivas ou hipóteses, seleção de fontes de informação, coleta, seleção e classificação de dados, generalização das conclusões e expressão e comunicação.

Para se analisar uma sequência didática é imprescindível utilizar as condições da aprendizagem significativa que leva à introdução de uma nova unidade de análise, a sequência de conteúdo, para que se possa acompanhar os processos de ensino-aprendizagem conforme as características dos tipos de conteúdos.

O termo Aprendizagem Significativa deriva do reconhecimento de que nem tudo o que aprendemos está integrado da mesma forma na estrutura do conhecimento.

Uma aprendizagem será mais ou menos significativa quando, além de significar uma memorização compreensiva, for possível sua aplicação em contextos diferenciados e quando puder ajudar a melhorar a interpretação ou a intervenção em todas as situações em que se fizerem necessárias. (ZABALA; ARNAU, 2010, p. 110)

Em sequências didáticas em que o professor ou professora comunica a lição, onde há um estudo individual sobre o livro-texto, uma repetição do conteúdo aprendido, uma aplicação de prova ou exame e uma avaliação dos resultados, conclui-se que os conteúdos que se trabalham são apenas conceituais e as habilidades trabalhadas não podem ser consideradas como conteúdos de aprendizagem, somente de memorização.

Quando uma situação problemática é utilizada em uma aula e a busca de soluções para o problema se dá por meio de propostas elaboradas pelos alunos, os conteúdos passam a ser procedimentais, uma vez que exige a utilização de um novo conceito e um novo modelo de algoritmo para ordenação das soluções.

Os conteúdos de aprendizagem passam a ser atitudinais quando acontece o diálogo entre

os alunos por meio das repostas intuitivas e suposições que são feitas a respeito do problema proposto. É neste momento que o aluno expõe seus conhecimentos, propõe a busca por informações e a elaboração de conclusões em um ambiente de trabalho no qual aprendem a “ser” tolerantes e cooperativos, que “saibam” os termos tratados e “saibam fazer” investigações.

Uma sequência didática em que as aprendizagens sejam o mais significativas possível deve apresentar uma maior variedade de atividades com uma estrutura organizada de forma a evidenciar os conhecimentos prévios dos alunos, de atribuir significância e funcionalidade aos novos conteúdos, promover o desenvolvimento intelectual, a construção do conhecimento, favorecer o conflito cognitivo (que acontece quando o sujeito se depara com uma situação e percebe que os seus esquemas não são suficientes para solucionar a questão, gerando, então, um desequilíbrio cognitivo) e a atividade mental, motivar, melhorar a auto-estima e o autoconceito e fornecer ferramentas para se aprender a aprender.

Analisar os conteúdos de uma sequência didática também é muito importante para se alcançar os objetivos desejados. A sequência de conteúdo é:

[...] o conjunto ordenado de atividades estruturadas e articuladas para a consecução de um objetivo educacional em relação a um conteúdo concreto. [...] está inserida num contexto em que se deverá identificar, além dos objetos didáticos e conteúdo objeto da sequência, as outras variáveis metodológicas: relações interativas, organização social, materiais curriculares, etc. (ZABALA, 1998, p. 78)

Nesse sentido, deve-se relacionar a aprendizagem dos diferentes conteúdos segundo sua tipologia. Ensinar conteúdos factuais requer algum tipo de exercício de repetição utilizando estratégias que reforcem as organizações significativas e associações. O ensino de conceitos e princípios requer uma compreensão do significado e um processo de elaboração pessoal.

Para ensinar conteúdos procedimentais, as atividades devem partir de situações significativas e funcionais que apresentem modelos de desenvolvimento, com uma ordem gradual, ajudas de diferente grau e prática guiada e atividades de trabalho independente.

Ensinar conteúdos atitudinais implica em mobilizar recursos relacionados com o componente afetivo. Os tipos de interação que se estabelecem em aula são peças-chave na configuração dos valores e das atitudes pessoais. O caráter do conteúdo atitudinal deve ser adaptado às necessidades reais dos alunos, partindo da realidade para aproveitar os conflitos que se apresentem, para introduzir processos de reflexão crítica e favorecer a criação de modelos das atitudes que se queiram desenvolver, além de fomentar a autonomia moral.

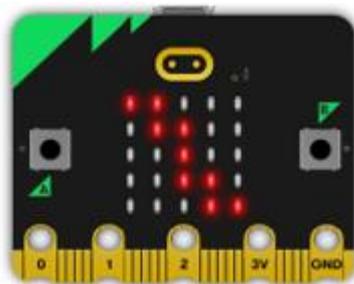
Para Zabala (1998), refletir sobre o que implica aprender o que se propõe, e o que implica aprendê-lo de maneira significativa, pode nos conduzir a estabelecer propostas mais fundamentadas, suscetíveis de ajudar mais os alunos e também os professores.

A escolha de se utilizar o micro:bit para o ensino do pensamento computacional e da robótica está alicerçada em uma sequência didática que pretende ensinar conteúdo conceitual, apresentando conceitos de física e computação, ensinar conteúdo procedimental, na prática de atividades graduais guiadas e independentes, e ensinar conteúdo atitudinal ao utilizar o componente afetivo e de integração promovido pela gamificação.

6.1 - Proposta da sequência didática

A sequência didática deste trabalho prevê a construção de um jogo, denominado de “Snake”, que tem por finalidade a repetição de uma imagem, Figura 6.1. Neste jogo, os alunos devem ver a imagem projetada e, apertando os botões “A” e “B” do micro:bit, reproduzi-las no menor tempo possível.

Figura 6.1: Exemplo de imagem que os alunos deverão reproduzir nos LEDs do micro:bit.



Fonte: Captura de tela pela autora (MakeCode).

A proposta de sequência didática utilizando o micro:bit prevê a aplicação em duas aulas de 45 a 50 minutos cada, de preferência geminadas. Os projetos criados com o micro:bit são bastante dinâmicos e, mesmo os jogos mais complexos, podem ser programados em um tempo médio de 90 minutos, dependendo do conhecimento prévio em programação e da experiência de uso do micro:bit.

A sequência didática deste trabalho pode ser utilizada desde as etapas do ensino fundamental e médio até o ensino superior. No ensino fundamental, por exemplo, ela se encaixa nas habilidades de lógica computacional do terceiro ano (EF03CO01), de matrizes e registros do quarto ano (EF04CO01), programação no sétimo ano (EF07CO03), entre outras várias

habilidades. Para o ensino médio, podemos associá-la à habilidade de desenvolvimento de projetos com robótica (EM13CO16). Os conceitos de algoritmo, lógica, pensamento computacional e programação são ensinados no curso de graduação em Física da Universidade Estadual de Maringá na disciplina de Fundamentos da Computação em seu programa de número 1640 de 2006.

A experiência adquirida como monitora e ministrante dos cursos de micro:bit, por meio do projeto de extensão trUEM³, mostraram que à medida que os conhecimentos em pensamento computacional e robótica aumentam em decorrência das atividades realizadas com o micro:bit, projetos mais complexos podem ser criados em menor tempo.

O objetivo geral desta sequência didática é ensinar pensamento computacional e robótica. Dentre os objetivos específicos estão a aprendizagem dos conteúdos conceituais, procedimentais e atitudinais com o uso do microcontrolador micro:bit, como um recurso metodológico ativo e tecnológico, e o desenvolvimento de um jogo que será utilizado para reconhecer a capacidade de memória operacional dos alunos.

Para todas as aulas, serão necessários computadores, microcontroladores micro:bit, um projetor e uma lousa ou quadro para anotações. O professor usará um computador e o projetor para mostrar aos alunos os recursos do micro:bit e o passo a passo do código. Seria muito interessante que todos os alunos tivessem um computador e um micro:bit cada para utilizar, mas se não houver essa possibilidade, os alunos podem ser dispostos em grupos. O ideal é que os grupos sejam de, no máximo, três pessoas.

6.1.1 - Primeira aula

Para a primeira aula, o professor precisará também de um copo plástico e de pequenos pedaços de papel que sejam do tamanho do fundo do copo. Esses materiais serão utilizados para a explicação do conceito de variáveis na programação.

Normalmente, esse projeto de jogo é criado depois de algumas aulas já ministradas com o micro:bit. Mas aqui, mostraremos que é possível aplicar o conteúdo proposto dada a versatilidade e facilidade de manuseio deste recurso tecnológico.

Para iniciar a aula, o professor mostrará o micro:bit e perguntará aos alunos se eles o conhecem. Independente das respostas serem afirmativas ou negativas, o professor irá apresentar o microcontrolador explicando suas partes e dando alguns exemplos de atividades

³ Projeto de Extensão da Universidade Estadual de Maringá (UEM) que tem por objetivo trazer o conhecimento e desafios da Robótica e Inteligência Artificial (IA) por meio de experiências de aprendizado que são tangíveis, acessíveis e inclusivas. Para mais informações acesse <https://truem.com.br/>.

que podem ser realizadas com o mesmo.

Em seguida, utilizando o projetor, o professor mostrará alguns dos projetos que podem ser criados com o micro:bit para instigar a curiosidade e motivar os alunos para a criação do projeto proposto.

Os alunos serão, então, instruídos a utilizarem seus computadores, individualmente ou em duplas ou trios, para entrar na plataforma MakeCode, que pode ser acessada *online* ou via aplicativo instalado no computador, como explicado no capítulo 04. Caso o professor opte pelo uso do aplicativo, a instalação deverá ser feita anteriormente à aplicação da aula. Os alunos serão ensinados a criar um novo projeto com o nome de “Snake”.

A partir deste momento, o professor irá explicar como é feita uma programação em blocos, mostrando como é utilizada a biblioteca de funções para a montagem de um código. Após a explicação e exemplificação, o professor dirá que o objetivo final é criar um jogo com o micro:bit e mostrará, com uma placa já programada ou no simulador da plataforma, o funcionamento do mesmo.

Antes de dar início à programação, os conceitos de variáveis, de lógica, de condicionais, de variação de tempo e de matriz serão explicados.

As variáveis serão explicadas com o uso do copo e dos pedaços de papel. O professor dirá que um copo pode armazenar bebidas, entre outras coisas, e que as variáveis na programação também armazenam coisas que são as informações.

O professor explicará o conceito de definir e alterar uma variável da seguinte forma: dirá que o nome dado para a variável é “copo” e pedirá aos alunos para dizerem números de zero a nove para anotar nos papéis. A anotação de sete números é suficiente. Depois, escolherá um papel e dirá “Estou definindo a variável copo para o número 2.” (por exemplo), e perguntará aos alunos quanto vale a variável copo. Todos deverão responder 2. Esse processo deve ser repetido com mais três ou quatro números anotados.

Para ensinar o conceito de alterar uma variável, o professor explicará que isso significa realizar uma operação matemática, ou seja, se a variável copo está definida com o número 2 e, se deseja alterá-la pelo número 4, a variável passa a assumir o valor da soma de 2 mais 4, que é 6. Sendo assim, o professor irá dizer “A variável copo está definida para 2 e quero alterá-la por 4 (mostra o papel com o número 4 escrito). Quanto vale a variável agora?”. A resposta dos alunos deverá ser 6. O processo deve ser repetido utilizando números negativos também para a fixação do conceito. Por exemplo, “A variável copo está definida para 3 e quero alterá-la por menos 2. Quanto vale a variável agora?”. A resposta dos alunos deverá ser 1.

Depois de concluída essa dinâmica, os alunos deverão acessar a biblioteca de funções,

na aba “Variáveis” e o professor mostrará como fazer uma variável para ser utilizada no código do jogo. No total, serão criadas cinco variáveis. Se houver mais de um aluno por computador, a tarefa de criação será dividida entre eles. A primeira variável criada será chamada de “início”. Aqui, vale ressaltar, que as palavras usadas para variáveis em programação não levam acento gráfico, espaço ou caracteres especiais, e também não podem ser iniciadas com números.

Antes de criar as demais variáveis, o professor explicará o conceito de variação de tempo, que será utilizado no jogo para a marcação do tempo decorrido para a realização da tarefa. A forma como este conceito físico será abordado ficará a critério do professor, pois depende do nível escolar da turma em que a atividade será aplicada. Após a explicação, as variáveis “tempo_final” e “tempo_inicial” deverão ser criadas.

Para finalizar a etapa de criação de variáveis, o professor revisará (ou explicará) o conceito de plano cartesiano para mostrar como os LEDs serão acesos sequencialmente nos eixos x e y para formar o caminho que se pretende construir no jogo. Ao final da explicação, as variáveis x e y serão criadas, finalizando a etapa de criação das variáveis.

O próximo passo será explicar os conceitos de verdadeiro e falso. Uma forma simples de explicá-los é escrever dois números com sinais de maior ($>$), menor ($<$), igual ($=$) e diferente (\neq) entre eles e perguntar aos alunos se o que está escrito é verdadeiro ou falso. Por exemplo, escrever na lousa ou quadro de anotações: $5 > 6$, $4 < 2$, $3 = 3$, $8 \neq 7$ e perguntar quais proposições são verdadeiras e quais são falsas. Repetir essa atividade com vários números até que os alunos consigam responder corretamente a todas as proposições.

A última etapa da primeira aula é entender o conceito de condicional. Nesta explicação, o professor pode dizer que irá falar frases em que o conceito de condicional está envolvido, como por exemplo, “se estiver chovendo, então eu levo um guarda-chuva”; “se não estiver chovendo, então eu não levo um guarda-chuva”. E, depois, pedir para que os alunos digam frases que envolvem situações que dependem de outras para acontecerem.

6.1.2 - Segunda aula

Se a segunda aula acontecer logo em seguida à primeira, basta informar que o objetivo agora é escrever o código do jogo que foi mostrado na aula anterior. Caso contrário, é interessante construir na lousa ou quadro de anotações um mapa conceitual com os alunos sobre o que foi aprendido na última aula. Se não houver um computador para cada aluno, o professor deve dividir a tarefa de programação de uma forma em que todos os alunos participem do processo de escrita do código.

O objetivo do jogo é formar um caminho de LEDs acesos apertando os botões A e B do

micro:bit em menor tempo. O botão A acenderá um LED à direita e o botão B um LED de baixo, sempre em sequência ao último que foi ligado. O professor pode mostrar o jogo aos alunos por meio do simulador ou diretamente no micro:bit, anteriormente programado.

Para começar a escrever o código, os alunos deverão abrir o projeto criado com o nome “Snake”. Na página inicial, aparecerão os blocos de acionamento “no iniciar” e “sempre”. O professor dirá que não será utilizado o bloco “sempre” para este jogo e que o mesmo pode ser excluído. O próximo passo é informar que serão criadas as condições iniciais do jogo. Para isso, os alunos deverão arrastar alguns comandos, que estão na biblioteca de funções, para dentro do bloco “no iniciar”. Os comandos serão obtidos das abas “Variáveis”, “Lógica” e “Led”, como na Figura 6.2:

Figura 6.2: Esquema de montagem do bloco das condições iniciais do jogo.



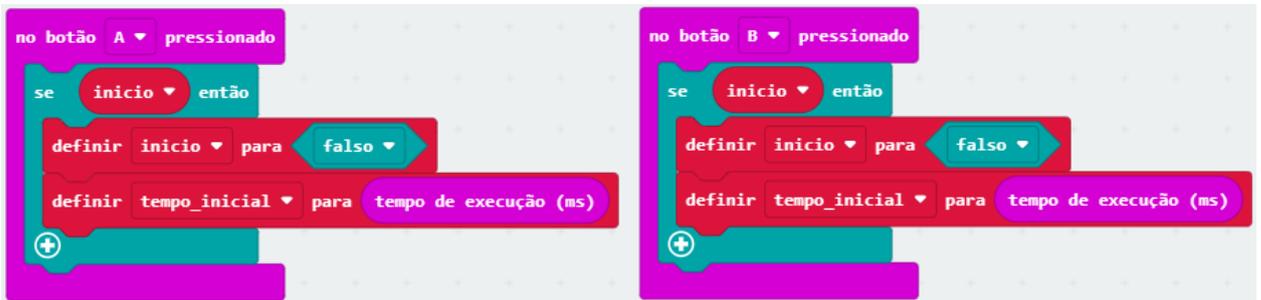
Fonte: Captura de tela pela autora (MakeCode).

Note que, na montagem deste bloco, os conceitos de variáveis e de condicional são retomados e é ensinado como as variáveis x e y são utilizadas para se acender os LEDs do micro:bit neste jogo.

O que se segue é a construção dos comandos que farão os LEDs se acenderem, e permanecerem acesos, quando os botões “A” e “B” do micro:bit forem apertados. A cada aperto do botão “A”, um LED à direita do último que foi ligado, será aceso. Para o botão “B”, a cada aperto, o LED que está abaixo do último ligado, será aceso.

Para a programação do botão “A”, as abas “Input”, “Lógica”, “Variáveis” e “Led” serão utilizadas. Primeiro, os alunos construirão a condição de ativação do tempo do jogo utilizando os blocos da aba “Lógica” e definindo as variáveis. A Figura 6.3, mostra como a sequência deve ser construída.

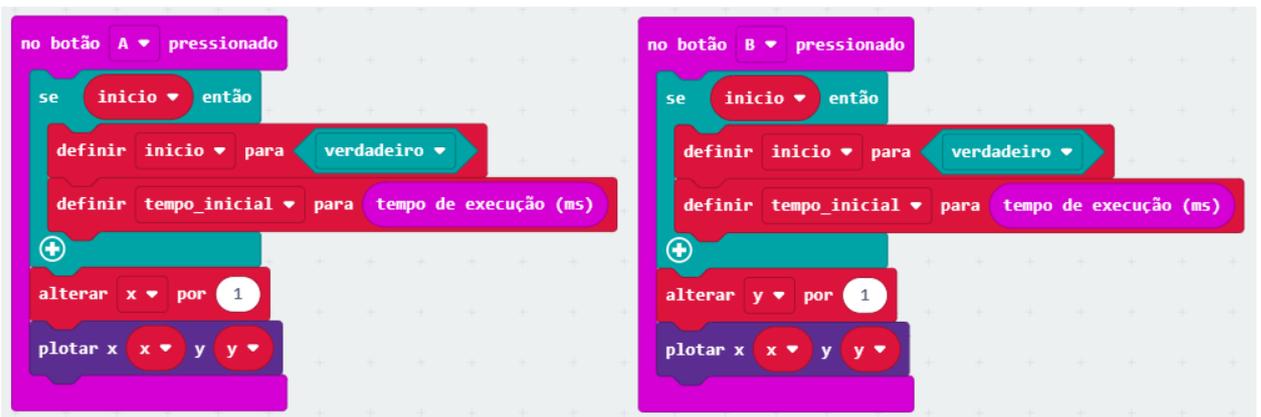
Figura 6.3: Esquema de montagem dos blocos das condições iniciais de cada botão.



Fonte: Captura de tela pela autora (MakeCode).

O conceito de lógica será trabalhado, então, no momento da construção dos comandos para os botões “A” e “B”. O professor ensinará os comandos, por meio dos blocos de alterar uma variável e plotar, mas deixará um erro para lançar um desafio. O erro é faltar uma condição para que os botões funcionem dentro do limite dos cinco LEDs nas linhas e nas colunas. O desafio será dizer como solucionar o problema de fazer o LED continuar seu caminho, para baixo ou para a direita, mesmo que o jogador aperte os botões “A” e “B” mais de cinco vezes. A Figura 6.4 mostra o código com o erro e a Figura 6.5 o código corrigido com a condição de que a variável seja menor que 4.

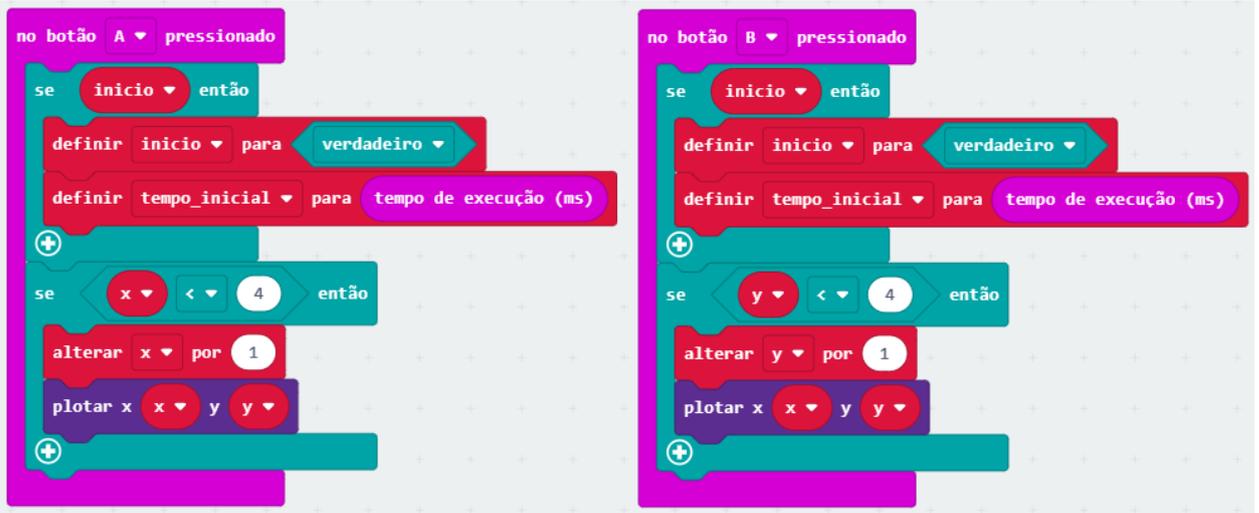
Figura 6.4: Esquema de montagem dos blocos para mover cada botão.



Fonte: Captura de tela pela autora (MakeCode).

A condição da variável ser menor que 4 significa dizer que o comando apenas será executado enquanto o número de apertos em cada botão não exceder 4 vezes, independentemente de serem pressionados em sequência ou alternadamente.

Figura 6.5: Esquema de montagem dos blocos para mover cada botão com as condições que solucionam o erro.

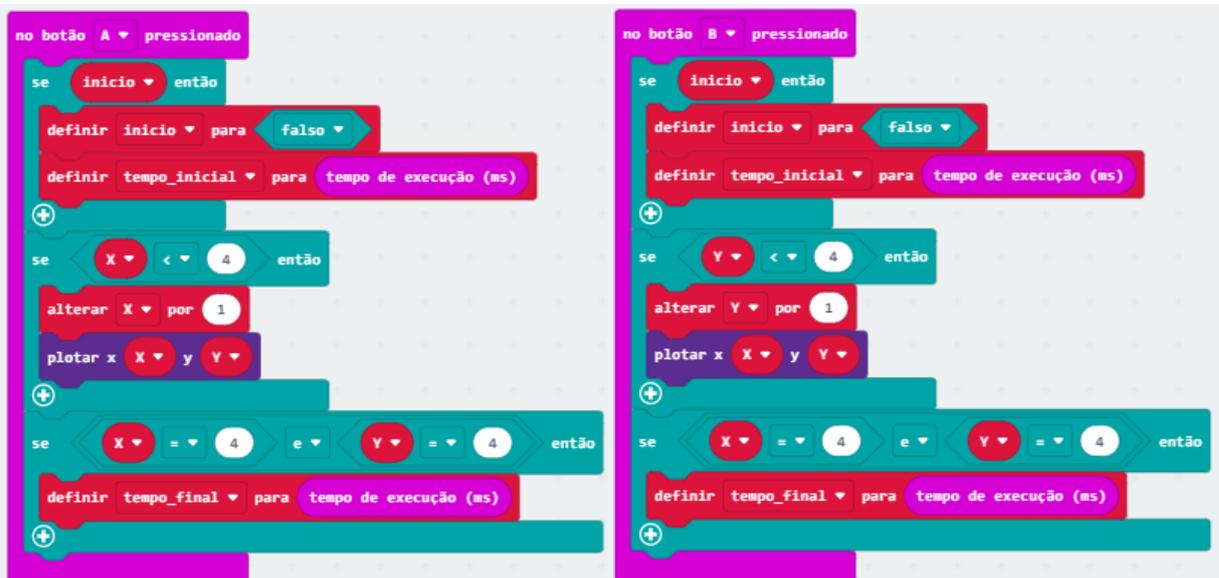


Fonte: Captura de tela pela autora (MakeCode).

As partes finais do programa são montar o código para que o tempo pare de ser contado quando o último LED da sequência for aceso e mostrar o tempo total de realização da tarefa.

Para a finalização da contagem de tempo, será necessária a inclusão de mais um bloco de lógica que envolve uma nova condição, dentro de cada um dos blocos dos botões. Essa condição trabalhada será a *booleana*, ou seja, que opera com as funções *E* e *OU*. No jogo que está sendo criado, queremos que o tempo pare de ser contado quando as variáveis *x* e *y* sejam iguais a 4. Desta forma, serão incluídas as condições de $x = 4$ e $y = 4$ para o tempo parar de ser contado. A Figura 6.6 apresenta como a condição deve ser expressa.

Figura 6.6: Esquema de montagem dos blocos com as condições que param a contagem do tempo.



Fonte: Captura de tela pela autora (MakeCode).

O jogo está quase terminado. Falta apenas programar como o tempo total para a realização da tarefa será mostrado. Para isso, foi escolhida a função de toque no logotipo do micro:bit para mostrar o tempo. Na aba “Input” será escolhido o bloco de acionamento “no logotipo pressionado”. A condição para mostrar o tempo total implica novamente na função *booleana E*. Assim, se ambos os tempos forem diferentes de zero, então, o código executará a conta de variação do tempo, fará a conversão para segundos (já que o micro:bit apresenta o tempo em milissegundos por padrão) e mostrará o tempo nos LEDs. A Figura 6.7 mostra o código completo desta etapa.

Figura 6.7: Esquema de montagem do bloco que mostra a contagem do tempo.



Fonte: Captura de tela pela autora (MakeCode).

Ao final da programação, os alunos devem testar o código no simulador e, se estiver correto, transferi-lo para o micro:bit. Os micro:bits podem ser usados conectados aos computadores ou a suportes de pilhas (o uso do suporte facilita o manuseio da placa). Neste ponto, acontece a primeira avaliação da sequência didática. Quando os alunos testam o código é possível verificar quais deles conseguiram executar a programação corretamente e, os que erraram, utilizarão os conceitos aprendidos para consertar o código reforçando e fixando o aprendizado.

O código inteiro e completo pode ser encontrado no Anexo A deste trabalho.

Concluído o objetivo de construção do jogo, o próximo passo é jogá-lo para testar a capacidade de memória operacional dos alunos, que irá verificar a habilidade de armazenamento e processamento das informações. Esta também se constitui em uma atividade avaliativa, pois os alunos que executarem a tarefa em menor tempo, demonstrarão que possuem uma memória operacional mais desenvolvida.

Para a atividade do jogo, o professor explicará que todos os alunos participarão, de cinco em cinco por vez, e que eles terão que copiar o caminho que será projetado em menor tempo.

Cada vez que errar, o aluno deve apertar o botão *reset* que fica atrás do micro:bit para recomeçar. Ganha quem fizer o caminho correto primeiro e em menor tempo. Alguns modelos de caminho estão disponíveis no Anexo B deste trabalho.

A forma de premiação do jogo é de livre escolha do professor, podendo ser de pequenos presentes, como doces, a pontos na nota final do bimestre/trimestre.

Esta sequência didática encontra-se na forma de tópicos no Apêndice deste trabalho.

Se o professor optar por uma programação menor e mais rápida, também disponibilizamos o código de um jogo sem a marcação do tempo, no Anexo C.

Nos Anexos D e E estão disponíveis os códigos nas linguagens JavaScript e Python.

A sequência didática elaborada neste trabalho buscou **ensinar pensamento computacional e robótica** utilizando um recurso metodológico ativo e tecnológico, o micro:bit, integrando as áreas de ciência, tecnologia, engenharia, artes e matemática (STEAM), com o intuito de promover uma aprendizagem significativa dos conteúdos conceituais (determinantes para a compreensão do pensamento computacional), procedimentais (ao utilizar a programação em blocos) e atitudinais (com o desenvolvimento e aplicação de um jogo).

Considerações finais

A utilização do microcontrolador micro:bit como um recurso pedagógico para o ensino do pensamento computacional e da robótica já é tema de alguns estudos científicos realizados. A pesquisa desenvolvida neste trabalho buscou estabelecer as relações entre as práticas de ensino-aprendizagem conhecidas na literatura e o uso de um recurso tecnológico criado para desenvolver habilidades e competências para a resolução de problemas da vida cotidiana.

Visto que foi possível explicar e propor o uso do micro:bit como uma ferramenta disponível para o ensino de raciocínio lógico, criatividade, conceitos físicos e resolução de problemas, pode-se inferir que os objetivos deste estudo foram alcançados. Com a compreensão dos fundamentos da abordagem de ensino STEAM, em que o estudante assimila conhecimentos de diferentes áreas de forma integrada, com a percepção da associação existente entre pensamento computacional e memória operacional e com a gamificação utilizando o micro:bit, este trabalho contribuiu para reforçar as diversas aplicações deste microcontrolador e para incentivar pesquisas futuras.

A sequência didática proposta, que foi baseada em aulas ministradas em cursos oferecidos pelo projeto de extensão trUEM para alunos de várias idades e para professores do ensino médio, mostra-se bastante eficiente, em uma análise qualitativa, em relação aos aspectos de engajamento, motivação, cooperação e criatividade. Estudos que façam uma análise quantitativa de melhoria da aprendizagem e do desenvolvimento da memória operacional com o uso do micro:bit são algumas das perspectivas futuras deste trabalho.

Outros estudos sugeridos são testes em diferentes faixas etárias para verificar se haverá uma redução no tempo de execução da tarefa. Também seria interessante verificar se a memória operacional em crianças que fazem uso excessivo de aparelhos celulares é melhor ou pior em comparação àquelas que utilizam os celulares de forma adequada.

Incluir o pensamento computacional e a robótica na formação dos acadêmicos de licenciatura em Física pode ser uma forma de promover a integração entre os conteúdos tornando-os mais acessíveis, aplicáveis e possibilitar uma aprendizagem mais significativa. Os pilares do pensamento computacional (decomposição, reconhecimento de padrões, abstração e algoritmos) podem auxiliar os alunos na resolução de listas de exercícios propostos e a robótica pode promover a integração entre teoria e prática nas disciplinas experimentais.

Assim, conclui-se que o estudo realizado neste trabalho tem o potencial de contribuir para o ensino e a prática do pensamento computacional e da robótica, fazendo uso do microcontrolador micro:bit como uma nova ferramenta metodológica ativa de ensino. A

sequência didática indica, ao promover a integração entre as diversas áreas do conhecimento, que a abordagem de ensino STEAM em conjunto com uma atividade gamificada utilizando o micro:bit como recurso tecnológico promove o aprendizado de conteúdos de Física e capacita os alunos para conseguirem solucionar problemas da vida real. Além disso, a sequência didática proposta coloca o aluno como protagonista do aprendizado, ensina conteúdos de maneira inovadora e permite o desenvolvimento e treinamento da memória operacional. Tais benefícios não costumam ser obtidos por meio de uma aula tradicional.

Referências bibliográficas

ABREU, P. E. de; TOURINHO, C. **Memória operacional e aprendizado - Guia prático.**

FNR - Fonds National de la Recherche, Universidade de Luxemburgo, 2011.

Disponível em: <https://hdl.handle.net/10993/2044>. Acesso em: 22 nov. 2023.

ALVES, L. R. G; MINHO, M. R. S.; DINIZ, M. V. C. **Gamificação: diálogos com a educação.** In: Gamificação na Educação. Pimenta Cultural: São Paulo, 2014. Disponível em:

[https://www.academia.edu/9139616/_eBook_-_](https://www.academia.edu/9139616/_eBook_-_PDF_Gamifica%C3%A7%C3%A3o_na_Educa%C3%A7%C3%A3o)

[_PDF_Gamifica%C3%A7%C3%A3o_na_Educa%C3%A7%C3%A3o](https://www.academia.edu/9139616/_eBook_-_PDF_Gamifica%C3%A7%C3%A3o_na_Educa%C3%A7%C3%A3o). Acesso em: 16 jan. 2024.

Aprendendo com micro:bit BBC. Projeto trUEM, vários autores, 2023. Disponível em:

https://truem.com.br/dwl/micro_bit-v1.pdf. Acesso em: 08 out. 2023.

AQUINO, J. L; BORGES, M de O. **Avaliação neuropsicológica da memória operacional em escolares.** Rev. Psicopedagogia 2019; 36 (109): 3-9

BRACKMANN, C. P. **Desenvolvimento do Pensamento Computacional através de Atividades Desplugadas na Educação Básica.** 2017. 226 f. Tese (Programa de Pós-Graduação em Informática na Educação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

BRASIL. Ministério da Educação. Base Nacional Comum Curricular. Brasília: MEC, 2018.

BRASIL. Normas sobre Computação na Educação Básica – Complemento à

Base Nacional Comum Curricular (BNCC). Brasília: MEC, SEB, 2022. Disponível em:

<http://portal.mec.gov.br/docman/fevereiro-2022-pdf/236791-anexo-ao-parecer-cneceb-n-2-2022-bncc-computacao/file>. Acesso em: 30 ago. 2023.

CAMPOS, D. C.; LIMA, E. J. de; CINTRA, D. D.; MORAES, D. V. de. **A abordagem STEAM e suas tendências pedagógicas e metodológicas.** Research, Society and Development, [S. l.], v. 11, n. 15, p. e190111537148, 2022. Disponível em:

<https://rsdjournal.org/index.php/rsd/article/view/37148>. Acesso em: 19 dez. 2023.

DUGGER JUNIOR, William E. **Evolution of STEM in the United States**. [S. l.]: Virginia Tech, 2010.

Guia de Consulta Rápida para aprender pensamento computacional e robótica com o BBC micro:bit. Projeto trUEM, vários autores, 2023. Disponível em: <https://truem.com.br/dwl/GCR-v1.pdf>. Acesso em: 08 out. 2023.

MURR, C. E.; FERRARI, G. **Entendendo e aplicando a gamificação**. Florianópolis: UFSC: UAB, 2020.

PAPERT, S. *Mindstorms: Children, Computers and Powerful Ideas*. 2. Ed. New York: BasicBooks, 1993. 252 p.

PAPERT, S. *The Children's Machine: Rethinking School In The Age Of The Computer*. New York: BasicBooks, 1994. 256 p.

PIAGET, J. *To understand is to invent*. New York: Grossman Publishers, 1972.

PUGLIESE, G. O. **Um panorama do STEAM education como tendência global**. In: Lilian Bacich, Leandro Holanda. (Org.). *STEAM em Sala de Aula - A Aprendizagem Baseada em Projetos - Integrando Conhecimentos na Educação Básica*. 1ed. Porto Alegre: Penso, 2020, v. , p. 35-.

PUGLIESE, G. **STEM Education – um panorama e sua relação com a educação brasileira**. Currículo sem Fronteiras. 20. 10.35786/1645-1384. v20. n1.12., 2020.

SILVA, J. B da; SALES, G. L.; CASTRO, J. B. de. **Gamificação como estratégia de aprendizagem ativa no ensino de Física**. Revista Brasileira de Ensino de Física, vol. 41, nº 4, e20180309 (2019)

SOUZA, P. N. P. de; SILVA, E. B. da. **Como entender e aplicar a nova LDB: LEI no. 9.394/96**. São Paulo, Brasil: Cengage Learning, 1997.

VOŠTINÁR, P.; KVAŠŠAYOVÁ, N. *Experience with Using Robots for Teaching*

Programming, 2019. International Conference on Information and Digital Technologies (IDT), 10.1109/DT.2019.8813310 (2019).

WERBACH, K.; HUNTER, D. *For the win*. Philadelphia: Wharton, 2012.

YAKMAN, G. *STEAM Education: an overview of creating a model of integrative education*. Tese. Virginia Polytechnic and State University, Pulaski, 2008.

ZABALA, A. *A prática educativa: como ensinar*. Porto Alegre: ArtMed, 1998.

ZABALA, A.; ARNAU, L. *Como aprender e ensinar competências*. Porto Alegre: Penso, 2010.

APÊNDICE

Aula 1 - Introdução aos conceitos de pensamento computacional e robótica

Duração: 45-50 minutos

Objetivos:

- Entender o que são e como são utilizadas as variáveis na programação (conteúdos conceituais);
- Compreender conceitos de variação de tempo (conteúdos conceituais);
- Entender o conceito de condicional (conteúdos conceituais).

Metodologia:

- Será feita uma apresentação do micro:bit e dos projetos que podem ser criados com ele;
- Os alunos acessarão a plataforma MakeCode para aprenderem como criar projetos;
- O professor mostrará o jogo pronto para despertar o interesse e motivar os alunos;
- O conceito de variável será explicado por meio de um copo e pedaços de papel;
- Os alunos aprenderão a fazer variáveis para o programa do jogo;
- Explicação do conceito de variação de tempo e criação das variáveis tempo_final e tempo_inicial; e
- Explicação do conceito de verdadeiro e falso utilizando os sinais de maior (>), menor (<), igual (=) e diferente (≠).

Materiais Utilizados:

- Computadores:
 - um para o professor; e
 - um para cada aluno (caso não haja disponibilidade, os alunos podem ser agrupados em duplas ou trios);
- Microcontroladores micro:bit, com cabos de transferência de dados e um suporte para pilhas:
 - um para o professor; e
 - um para cada aluno ou grupo.
- Um projetor;
- Uma lousa ou quadro para anotações;

- Um copo plástico; e
- Papéis cortados em pequenos quadrados que se encaixem no fundo do copo.

Aula 2 - Criação do jogo “Snake”

Duração: 45-50 minutos

Objetivos:

- Compreender o conceito de lógica (conteúdos conceituais);
- Aprender como plotar na matriz de LEDs (conteúdos procedimentais);
- Aprender a escrever um código para um jogo (conteúdos procedimentais);
- Jogar um jogo programado no micro:bit (conteúdos atitudinais).

Metodologia:

- Será feita uma revisão do conteúdo abordado na aula anterior utilizando um mapa conceitual (caso as aulas não sejam sequenciais);
- O professor mostrará novamente o jogo, explicando detalhadamente como os botões devem funcionar;
- O código das condições iniciais de jogo será escrito na plataforma MakeCode;
- O código das condições de contagem de tempo do jogo será escrito;
- Os comandos de plotar x e y será montado sem a condição que elimina o erro, primeiramente, e o código correto somente será escrito após as tentativas de conserto dos alunos;
- Explicação da condição booleana e escrita do código para a finalização da contagem de tempo;
- O código para mostrar o tempo total de jogo será escrito; e
- O jogo será jogado.

Materiais Utilizados:

- Computadores:
 - um para o professor; e
 - um para cada aluno (caso não haja disponibilidade, os alunos podem ser agrupados em duplas ou trios);
- Microcontroladores micro:bit, com cabos de transferência de dados e um suporte para

pilhas:

- um para o professor; e
- um para cada aluno ou grupo.
- Um projetor; e
- Uma lousa ou quadro para anotações.

Anexo A

Código completo da atividade chamada de “Snake”.

```
no iniciar
  definir tempo_inicial para 0
  definir tempo_final para 0
  definir inicio para verdadeiro
  definir X para 0
  definir Y para 0
  plotar x X y Y

no botão A pressionado
  se inicio então
    definir inicio para falso
    definir tempo_inicial para tempo de execução (ms)
  +
  se X < 4 então
    alterar X por 1
    plotar x X y Y
  +
  se X = 4 e Y = 4 então
    definir tempo_final para tempo de execução (ms)

no botão B pressionado
  se inicio então
    definir inicio para falso
    definir tempo_inicial para tempo de execução (ms)
  +
  se Y < 4 então
    alterar Y por 1
    plotar x X y Y
  +
  se X = 4 e Y = 4 então
    definir tempo_final para tempo de execução (ms)

no logotipo pressionado
  se tempo_inicial + 0 e tempo_final ≠ 0 então
    limpar tela
    mostrar número tempo_final - tempo_inicial / 1000
    mostrar string "SEGUNDOS"
```

Fonte: Captura de tela pela autora (MakeCode).

Figura A-1: QR Code para acesso ao link do código completo.

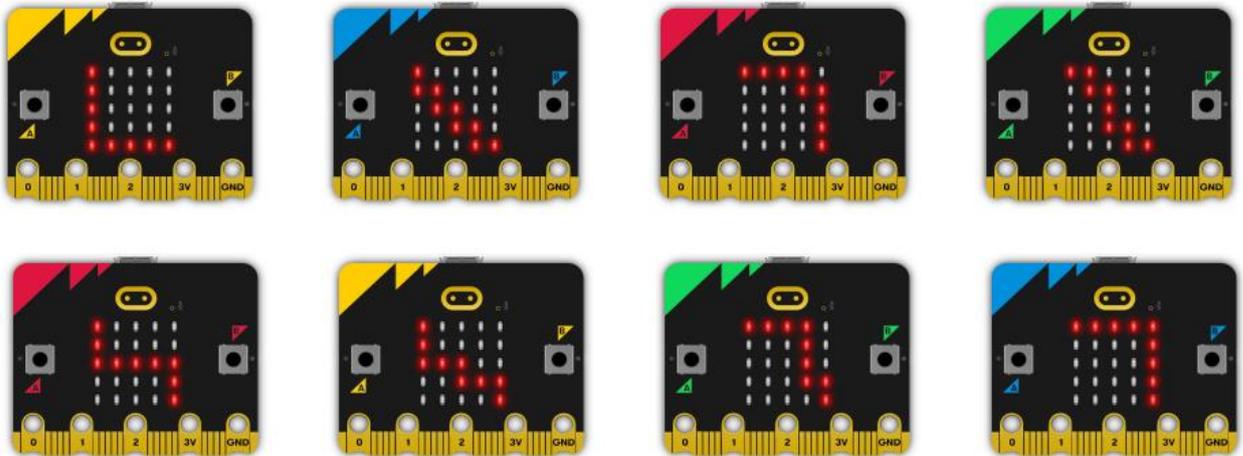


Fonte: Captura de tela pela autora (MakeCode).

Link de acesso ao código: <https://makecode.microbit.org/S48523-90837-13367-00194>

Anexo B

Exemplos de caminhos que os alunos devem construir com o micro:bit.



Fonte: Guia de Consulta Rápida para aprender pensamento computacional e robótica com o BBC micro:bit, 2023, p. 26.

Anexo C

Código simplificado da atividade chamada de “Snake”, sem a marcação de tempo.



Fonte: Captura de tela pela autora (MakeCode).

Figura C-1: QR Code para acesso ao link do código completo.



Fonte: Captura de tela pela autora (MakeCode).

Link de acesso ao código: <https://makecode.microbit.org/S48843-50573-95705-53956>

Anexo D

Código do jogo em linguagem de programação JavaScript.

```
input.onLogoEvent(TouchButtonEvent.Pressed, function () {
  if (tempo_inicial != 0 && tempo_final != 0) {
    basic.clearScreen()
    basic.showNumber((tempo_final - tempo_inicial) / 1000)
    basic.showString("SEGUNDOS")
  }
})
input.onButtonPressed(Button.A, function () {
  if (inicio) {
    inicio = false
    tempo_inicial = input.runningTime()
  }
  if (X < 4) {
    X += 1
    led.plot(X, Y)
  }
  if (X == 4 && Y == 4) {
    tempo_final = input.runningTime()
  }
})
input.onButtonPressed(Button.B, function () {
  if (inicio) {
    inicio = false
    tempo_inicial = input.runningTime()
  }
  if (Y < 4) {
    Y += 1
    led.plot(X, Y)
  }
  if (X == 4 && Y == 4) {
    tempo_final = input.runningTime()
  }
})
let Y = 0
let X = 0
let inicio = false
let tempo_final = 0
let tempo_inicial = 0
tempo_inicial = 0
tempo_final = 0
inicio = true
X = 0
Y = 0
led.plot(X, Y)
```

Anexo E

Código do jogo em linguagem de programação Python.

```
def on_logo_pressed():
    if tempo_inicial != 0 and tempo_final != 0:
        basic.clear_screen()
        basic.show_number((tempo_final - tempo_inicial) / 1000)
        basic.show_string("SEGUNDOS")
input.on_logo_event(TouchButtonEvent.PRESSED, on_logo_pressed)

def on_button_pressed_a():
    global inicio, tempo_inicial, X, tempo_final
    if inicio:
        inicio = False
        tempo_inicial = input.running_time()
    if X < 4:
        X += 1
        led.plot(X, Y)
    if X == 4 and Y == 4:
        tempo_final = input.running_time()
input.on_button_pressed(Button.A, on_button_pressed_a)

def on_button_pressed_b():
    global inicio, tempo_inicial, Y, tempo_final
    if inicio:
        inicio = False
        tempo_inicial = input.running_time()
    if Y < 4:
        Y += 1
        led.plot(X, Y)
    if X == 4 and Y == 4:
        tempo_final = input.running_time()
input.on_button_pressed(Button.B, on_button_pressed_b)

Y = 0
X = 0
inicio = False
tempo_final = 0
tempo_inicial = 0
tempo_inicial = 0
tempo_final = 0
inicio = True
X = 0
Y = 0
led.plot(X, Y)
```