

## INOVAÇÃO CURRICULAR COM PYTHON E ROBÓTICA: DESPERTANDO COMPETÊNCIAS DIGITAIS PARA O NOVO ENSINO MÉDIO

## CURRICULAR INNOVATION WITH PYTHON AND ROBOTICS: AWAKENING DIGITAL SKILLS FOR THE NEW HIGH SCHOOL

RODRIGO FELIPE DA SILVA MENDES, FLÁVIA GONÇALVES  
FERNANDES

### RESUMO

Este artigo explora a implementação de uma metodologia de ensino inovadora que integra a linguagem de programação *Python* e o kit LEGO® SPIKE™ Prime para alunos do 1º ano do Ensino Médio. Alinhado com a Base Nacional Comum Curricular (BNCC), o estudo destaca a importância crescente da programação na educação básica e seu papel no desenvolvimento de competências digitais e pensamento computacional. A experiência prática, realizada em uma oficina introdutória, demonstra como essa abordagem pode tornar o aprendizado mais interativo, motivador e relevante para os desafios tecnológicos do século XXI, preparando os estudantes para futuras oportunidades nas áreas de STEM (Ciência, Tecnologia, Engenharia e Matemática).

**Palavras chave:** Pensamento Computacional; Alfabetização Digital; Python.

### ABSTRACT

This article explores the implementation of an innovative teaching methodology that integrates the *Python* programming language and the LEGO® SPIKE™ Prime kit for 1st-year high school students. Aligned with the Brazilian National Common Core Curriculum (BNCC), the study highlights the growing importance of programming in basic education and its role in developing digital skills and computational thinking. The practical experience, carried out in an introductory workshop, demonstrates how this approach can make learning more interactive, motivating, and relevant to the technological challenges of the 21st century, preparing students for future opportunities in STEM (Science, Technology, Engineering, and Mathematics) areas.

**Keywords:** Computational Thinking; Digital Literacy; Python.

## INTRODUÇÃO

À medida que o mundo se torna cada vez mais digital, a educação em tecnologia assume um papel central na formação de futuros profissionais capacitados para enfrentar desafios modernos. No Brasil, a Base Nacional Comum Curricular (BRASIL, 2018, p. 531) destaca a importância de integrar competências tecnológicas e computacionais desde a educação básica e estabelece que “[...] utilizar os conceitos básicos de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática” (BRASIL, 2018, p. 531).

Um dos caminhos para alcançar essa integração é através do ensino de programação e robótica, disciplinas que fomentam não apenas o conhecimento técnico, mas também habilidades de resolução de problemas, lógica e criatividade.

## MATERIAIS E MÉTODOS

Este trabalho propõe a implementação de uma metodologia de ensino que utiliza a linguagem de programação *Python* em conjunto com o KIT LEGO® SPIKETM PRIME, direcionada a alunos do 1º ano do Ensino Médio. O KIT LEGO® SPIKETM PRIME, uma ferramenta educacional inovadora, oferece uma plataforma robusta para a exploração de conceitos básicos de robótica e programação de forma prática e engajadora. A escolha da *Python*, por sua sintaxe clara e sua ampla aplicabilidade, visa facilitar o primeiro contato dos estudantes com a programação, proporcionando uma curva de aprendizado suave e motivadora.

Integrar o ensino de *Python* e robótica com o KIT LEGO® SPIKETM PRIME ao currículo do primeiro ano do ensino médio, conforme orientações da Base Nacional Comum Curricular (BRASIL, 2018, p. 511), pode não apenas aumentar o interesse dos alunos por áreas STEM (Ciência, Tecnologia, Engenharia e Matemática), mas também prepará-los melhor para os desafios acadêmicos e profissionais futuros.

De maneira geral, o objetivo principal deste trabalho foi investigar e propor técnicas para a introdução de noções de programação com *Python* utilizando o KIT LEGO® SPIKETM PRIME para alunos do 1º ano do novo ensino médio, de forma a motivar mais alunos a procurarem pelos cursos relacionados a programação e para tentar melhorar a maneira como se introduz programação Marques et al. (2011).

Este trabalho relata a experiência obtida no desenvolvimento de algumas das técnicas levantadas em uma oficina de introdução a programação com *Python* para alunos do 1º ano do novo ensino médio de uma escola localizada no Estado do Rio de Janeiro como forma de incentivar a replicação dos resultados de aceitação encontrados pelos alunos e a possível aplicação em outras turmas e escolas.

## FUNDAMENTAÇÃO TEÓRICA

### O KIT LEGO EDUCATION SPIKE PRIME

A plataforma LEGO® Education SPIKE™ Prime é a sucessora da linha LEGO® Mindstorms®, oferecendo uma experiência inovadora e envolvente para estudantes que estão dando seus primeiros passos no mundo da robótica. Este kit (figura 1) é projetado para ser intuitivo e estimulante, com cerca de 528 peças que são fáceis de montar e desmontar, promovendo uma aprendizagem prática e criativa.

Figura 1 - Kit LEGO® Education SPIKE™ Prime



Fonte: LEGO® Education (2024).

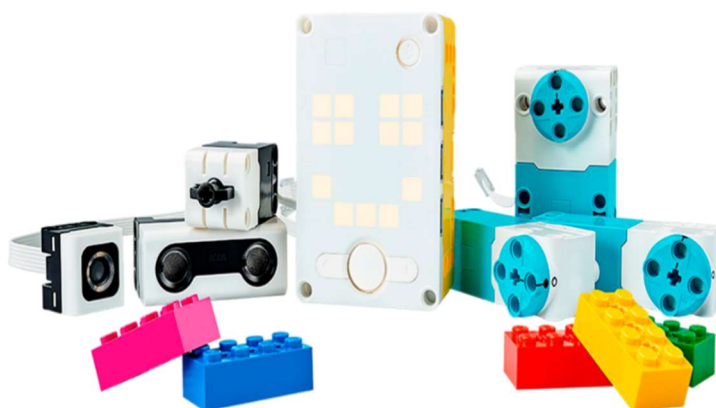
O “coração” do kit LEGO® SPIKETM PRIME é o Hub Inteligente, que é um bloco programável avançado que controla motores e sensores, além de oferecer conectividade sem fio via Bluetooth. Isso permite que os alunos usem um smartphone ou tablet com o aplicativo SPIKE™ para dar vida às suas criações robóticas. O Kit LEGO® Education SPIKE™ Prime é ideal para estudantes do ensino fundamental ao médio, proporcionando uma base sólida em STEM e codificação.

O kit inclui:

- Hub Inteligente, programável com 6 portas de entrada/saída;
- 5 Servomotores (2 grandes e 3 médios);
- Sensores: Distância, Força, Cor e Giroscópio (um de cada no kit básico e um segundo após a aquisição do kit expansão);
- Bateria recarregável e carregador USB;
- 528 peças LEGO® Technic intercambiáveis e coloridas.

Abaixo, encontra-se na figura 2 os componentes eletrônicos do kit LEGO® SPIKE™ Prime que ilustra alguns dos itens essenciais descritos acima:

Figura 2 - Componentes eletrônicos do Kit LEGO® SPIKE™ Prime



Fonte: LEGO® Education (2024).

### **O ambiente de programação e suas possibilidades**

A programação do SPIKE™ Prime é realizada dentro do ambiente de programação LEGO® Education SPIKE™ App. A linguagem de programação gráfica do SPIKE™ Prime é intuitiva e muito similar a aplicativos que utilizam a linguagem de programação em blocos como o caso do Scratch, consistindo em blocos funcionais que são organizados para a construção da programação. O programa de controle é composto por um conjunto de blocos que permitem aos alunos pensarem criticamente e resolver problemas complexos, independentemente do nível de aprendizado.

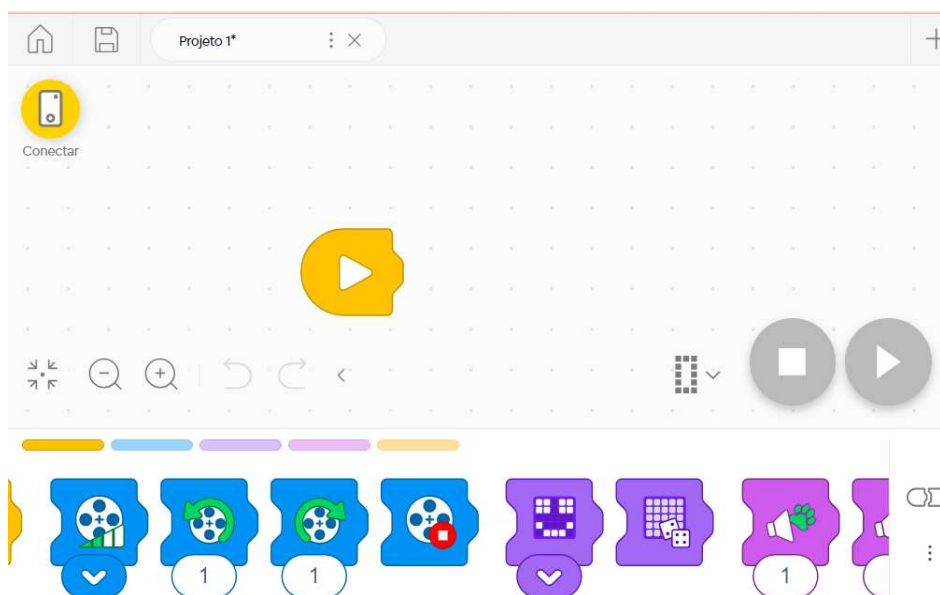
Nele, existem blocos pré-definidos para mover o robô para frente por algum tempo, para virar à esquerda ou à direita, além da opção de explorar a codificação baseada em texto com *Python*. O ambiente de programação do SPIKE™ Prime inclui áreas principais como a Tela de Programação e as Paletas de Programação, onde todos os blocos de programação estão localizados e categorizados para facilitar o acesso e a organização. Como podemos observar abaixo:

As lições do LEGO® Education SPIKE™ Prime estão alinhadas com os padrões educacionais NGSS (Next Generation Science Standards), ISTE (International Society for Technology in Education) e CSTA (Computer Science Teachers Association), além de oferecer extensões que atendem aos padrões CCSS (Common Core State Standards) para letramento e matemática. O ambiente de programação LEGO® Education SPIKE™ App pode ser acessado diretamente no site oficial da LEGO® Education.

### **Linguagem de programação em Blocos**

O LEGO® Education SPIKE™ App, apresenta um ambiente de programação de alto nível, onde eventos que ocorrerão durante a execução do programa são modelados por blocos de função, também conhecidos como blocos ícones. Na plataforma do SPIKE™ Prime, encontram-se blocos que correspondem a ações executadas pelos motores LEGO®, que estarão conectados ao hub inteligente. Sensores e motores podem receber e utilizar valores medidos, integrando-os à lógica do programa. Além dos blocos que desencadeiam eventos, existem outros que realizam operações lógicas e matemáticas, ampliando as possibilidades de programação e aprendizado. Na (figura 3) é possível observarmos o ambiente de aprendizagem para blocos de ícones.

Figura 3 - Ambiente de aprendizagem para blocos de ícones



Fonte: O autor (2024)

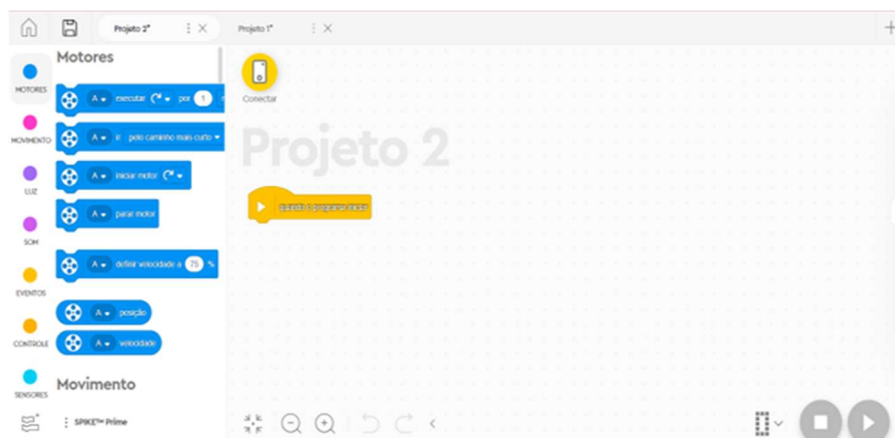
Nesse contexto, e de maneira similar ao Scratch, nos blocos de palavras o LEGO® Education SPIKE™ App possibilita que as crianças/adolescentes aprimorem suas habilidades cognitivas através da programação. Além de compartilharem suas criações, elas têm acesso a uma variedade de projetos já existentes no ambiente virtual da plataforma, o que enriquece e desenvolve o seu raciocínio lógico. Conforme Barbosa; Cambruzzi; Cardozo (2015, p. 9) destaca:

“É neste contexto de ensino através do concreto e do lúdico que a robótica educativa pode levar o estudante a explorar novas ideias, formular raciocínios complexos, contextualizando e descobrindo novos caminhos para a aplicação de conceitos adquiridos em sala de aula. Ao utilizar a robótica educativa na resolução de problemas, o aluno também desenvolve sua capacidade de elaborar hipóteses, investigar soluções, estabelecer relações sociais e de grupo e formular conclusões.”

A plataforma LEGO® Education SPIKE™ App é de fácil acesso e bastante intuitiva, conforme ilustra ambiente de aprendizagem para blocos de palavras na (figura 4). Essa imagem representa a tela inicial da interface na aba blocos de palavras, onde todos os comandos são organizados para a execução das ações e movimentos escolhidos.



Figura 4 - Ambiente de aprendizagem para blocos de palavras



Fonte: O autor (2024)

Na figura 5, é possível visualizar na tela inicial do programa LEGO® Education SPIKE™ App (localizada à esquerda) todos os botões operacionais e relacionais, organizados e diferenciados por cores e a possibilidade de acrescentarmos mais blocos. Assim como, criarmos os nossos próprios blocos para ações específicas. Isso facilita para o usuário o reconhecimento da função de cada botão, permitindo uma execução mais intuitiva das tarefas a serem desempenhadas pelos alunos ao longo de cada atividade.

Figura 5 - Comandos executáveis



Fonte: O autor (2024)

A plataforma é repleta de detalhes minuciosos. Possui um menu intuitivo para que os usuários possam se familiarizar com agilidade aos seus comandos, além de um menu de tutoriais integrado, que é alimentado pelos elementos pré-fornecidos e pré-moldados da biblioteca da plataforma. Isso facilita o aprendizado e a criação de projetos interativos e educativos.

### **Programação em Python**

É importante refletir sobre o papel do pensamento computacional, destacado na Base Nacional Comum Curricular (BNCC) como uma meta para a disciplina de matemática nas etapas finais do ensino fundamental e médio. Conforme descrito na BNCC (2018, p. 271),

[...] a aprendizagem de Álgebra, como também aquelas relacionadas a Números, Geometria e Probabilidade e estatística, podem contribuir para o desenvolvimento do pensamento computacional dos alunos, tendo em vista que eles precisam ser capazes de traduzir uma situação dada em outras linguagens, como transformar situações-problema, apresentadas em língua materna, em fórmulas, tabelas e gráficos e vice-versa.

A programação de computadores está se tornando cada vez mais prevalente na sociedade. O mercado de trabalho apresenta uma demanda crescente por profissionais com conhecimento nesta área, mesmo aqueles sem formação técnica ou superior. Por isso, acredita-se que a programação deve ser introduzida nas escolas desde as séries iniciais. A BNCC apresenta o tema como um facilitador do processo de aprendizagem.

[...] o Tema Especial culturas digitais e computação se relaciona à abordagem, nas diferentes etapas da Educação Básica e pelos diferentes componentes curriculares, do uso pedagógico das novas tecnologias da comunicação e da exploração dessas novas tecnologias para a compreensão do mundo e para a atuação nele. Numa perspectiva crítica, as tecnologias da informação e comunicação são instrumentos de mediação da aprendizagem e as escolas, especialmente os professores, devem contribuir para que o estudante aprenda a obter, transmitir, analisar e selecionar informações. (BNCC, 2018, p. 50).



Em relação à interação das crianças com a tecnologia, é comum ouvir comentários sobre o suposto controle que os computadores exerceriam sobre as pessoas. Segundo Papert (1980/1986, p. 17),

“Na minha perspectiva é a criança que deve programar o computador e, ao fazê-lo, ela adquire um sentimento de domínio sobre um dos mais modernos e poderosos equipamentos tecnológicos e estabelece um contato íntimo com algumas das ideias mais profundas da ciência, da matemática e da arte de construir modelos intelectuais.”

O discutir o conceito de programação, Papert (1986, p. 18) descreve que “isso envolve, simplesmente, dialogar com o computador em um idioma compreensível tanto para ele quanto para o ser humano”. Utilizando códigos, o programador define comandos a serem executados, elementos a serem contabilizados e comparados, e o computador realiza as tarefas exatamente conforme instruído, seguindo a sequência prescrita. Papert (1986) também ressalta que é viável aprender a interagir com computadores de maneira análoga ao aprendizado da língua nativa, e que essa interação pode alterar o processo de aprendizado em outras áreas. Apesar do crescimento tecnológico ser um vetor importante no mercado atual, observa-se uma alta taxa de evasão nos cursos superiores relacionados à área.

Adicionalmente, “o processo de aprendizado através da programação é frequentemente marcado por tentativas e erros” (CORRÊA, NOTARE, 2019, p. 298). Conforme indicado pelos autores, tais erros são vistos como equívocos conceituais, transformando-se de instrumentos punitivos para momentos de descoberta e compreensão, o que pode enriquecer significativamente o ensino e a formação do saber.

No atual panorama de desenvolvimento computacional, educação e aplicação de tecnologias digitais, a introdução do ensino de lógica de programação e dos princípios da Ciência da Computação desde o nível fundamental tem sido um assunto de debate tanto em estudos acadêmicos quanto em círculos científicos (OLIVEIRA et al., 2014). Paralelamente, Geraldine (2014) menciona que várias campanhas apoiam a integração do ensino de programação em escolas convencionais, sem limitações, contando com o suporte de figuras proeminentes do setor tecnológico como Bill Gates, Mark Zuckerberg e Jack Dorsey, que veem na programação escolar um meio de

promover a inclusão digital. De acordo com Geraldês (2014), para esses influenciadores, a capacidade de compreender e produzir código é tão essencial quanto a habilidade de ler e escrever.

Assim como a Matemática fomenta o desenvolvimento do Pensamento Computacional, este último oferece contribuições significativas para o ensino e compreensão da Matemática. Conforme descrito na BNCC (2018, p. 271),

Associado ao pensamento computacional, cumpre salientar a importância dos algoritmos e de seus fluxogramas, que podem ser objetos de estudo nas aulas de Matemática. Um algoritmo é uma sequência finita de procedimentos que permite resolver um determinado problema. Assim, o algoritmo é a decomposição de um procedimento complexo em suas partes mais simples, relacionando-as e ordenando-as, e pode ser representado graficamente por um fluxograma. A linguagem algorítmica tem pontos em comum com a linguagem algébrica, sobretudo em relação ao conceito de variável.

De acordo com Barichello (2021, p. 4), o “pensamento computacional é definido como a habilidade de abordar problemas utilizando conhecimentos e técnicas da ciência da computação, que inclui organizar, representar e examinar dados e processos”. Neste contexto, a programação de computadores é empregada como ferramenta para organizar, representar e examinar o conhecimento de forma sistemática.

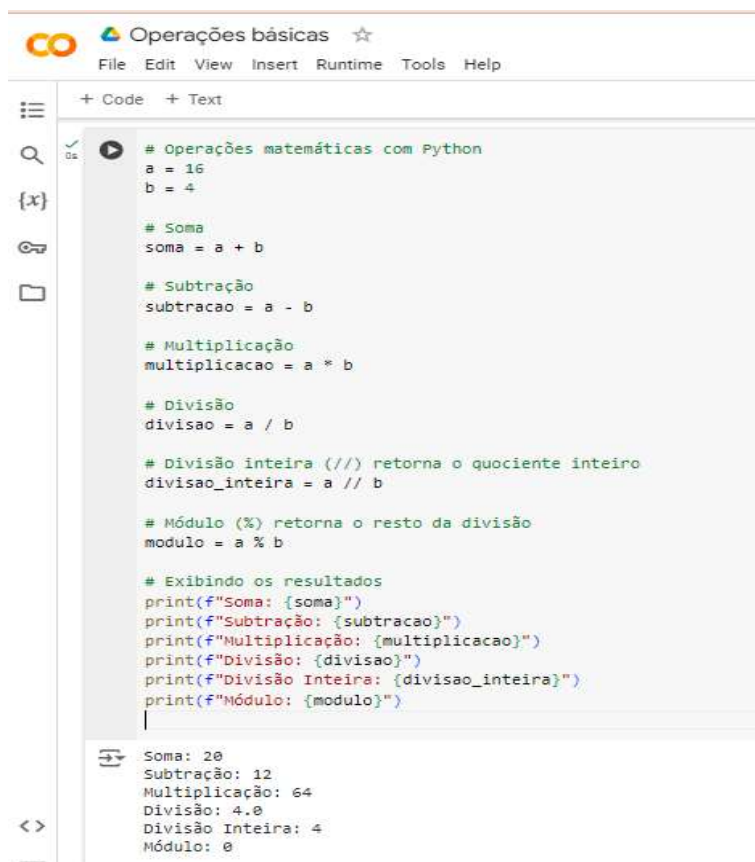
Para a experiência didática em questão, optou-se pela linguagem de programação *Python* como principal alvo de ensino/aprendizagem. Barichello (2021, p. 4) esclarece que,

Linguagem de programação: é uma linguagem usada para descrever algoritmos, escrita por um ser humano, que pode ser compreendida por um computador. Nesse caso, o termo “compreender” está sendo usado no sentido bem restrito de “seguir os passos”. Normalmente, linguagens de programação possuem um conjunto pequeno de comandos disponíveis e sintaxe muito rígida para que não reste qualquer ambiguidade no momento da execução dos comandos. Isso pode comprometer a agilidade do seu uso, pois é necessário conhecer vários detalhes específicos que, eventualmente, variam de uma linguagem para outra.

*Python* é uma linguagem de programação que pode oferecer funcionalidades semelhantes às de uma calculadora, entre outras. Ela possui operadores matemáticos

integrados que facilitam a realização de cálculos e a codificação de expressões algébricas. Além disso, o *Python* conta com funções específicas, como o operador `//`, que retorna o quociente inteiro de uma divisão, e o operador `%`, que fornece o resto da divisão, não a porcentagem, mas sim o valor remanescente após a divisão entre números inteiros. Na (figura 6), é possível observarmos a aplicação das operações básicas utilizando a linguagem de programação.

Figura 6 - Operações matemáticas básicas com *Python*

A screenshot of a Python IDE window titled "Operações básicas". The window has a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". Below the menu bar is a toolbar with icons for file operations and a search icon. The main editor area displays Python code for basic arithmetic operations. The code defines variables 'a' and 'b', calculates their sum, difference, product, and division, and prints the results. The output window at the bottom shows the results of these operations.

```
# Operações matemáticas com Python
a = 16
b = 4

# Soma
soma = a + b

# Subtração
subtracao = a - b

# Multiplicação
multiplicacao = a * b

# Divisão
divisao = a / b

# Divisão inteira (//) retorna o quociente inteiro
divisao_inteira = a // b

# Módulo (%) retorna o resto da divisão
modulo = a % b

# Exibindo os resultados
print(f"Soma: {soma}")
print(f"Subtração: {subtracao}")
print(f"Multiplicação: {multiplicacao}")
print(f"Divisão: {divisao}")
print(f"Divisão Inteira: {divisao_inteira}")
print(f"Módulo: {modulo}")
```

Soma: 20  
Subtração: 12  
Multiplicação: 64  
Divisão: 4.0  
Divisão Inteira: 4  
Módulo: 0

Fonte: O autor (2024)

Figura 7 - Ambiente de trabalho do Google Colab e codificação utilizada



The screenshot shows the Google Colab interface for a file named 'Untitled8.ipynb'. The top menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', 'Help', and 'All changes saved'. Below the menu, there are buttons for '+ Code' and '+ Text'. The left sidebar contains icons for file management, search, and variable inspection. The main area displays two code cells. The first cell, labeled '[6]', contains Python code that prompts the user for their birth year and age, calculates the year, and prints the result. The output shows the user entered '33' for age, and the program calculated '1991'. The second cell, labeled '[7]', contains Python code that prompts the user for their birth year and calculates their age, printing the result. The output shows the user entered '1991', and the program calculated '33'.

```
[6] a=input("Vamos descobrir em que ano você nasceu, informe a sua idade:")
    b=int(a)
    c=2024-b
    print("Você nasceu no ano de",c)

Vamos descobrir em que ano você nasceu, informe a sua idade:33
Você nasceu no ano de 1991

[7] a=input("Em que ano você nasceu?")
    b=int(a)
    c=2024-b
    print("Você tem ou vai fazer",c,"anos")

Em que ano você nasceu?1991
Você tem ou vai fazer 33 anos
```

Fonte: O autor (2024)

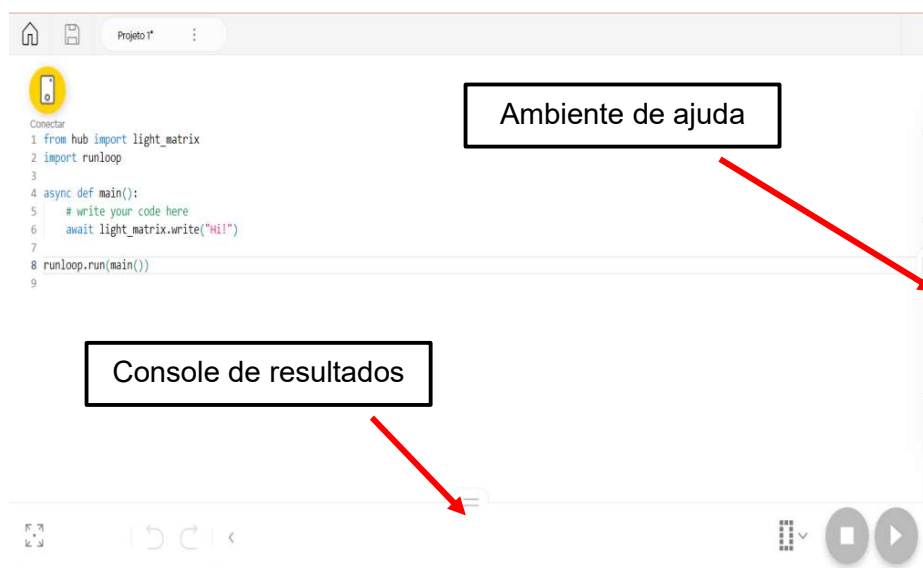
A capacidade de adaptação e personalização dos códigos é um dos aspectos que tornam o *Python* uma linguagem funcional e versátil, superando, em muitos casos, os métodos algébricos tradicionais que são mais rígidos e sequenciais. Existem diversas plataformas para programar em *Python*. O Integrated Development and Learning Environment (IDLE) é o ambiente de desenvolvimento integrado padrão do *Python*, que requer instalação e pode consumir uma quantidade significativa de memória do computador. Para início e utilização do *Python* e simplificar o acesso e uso por estudantes, começamos o trabalho utilizando o uso do Google Colaboratory, conhecido como Colab, que permite ler e executar códigos em *Python* (figura 7) diretamente no navegador, sem a necessidade de instalação local. Logo, após esse primeiro contato migramos para plataforma do LEGO® SPIKETM PRIME App.

Na figura 7, é possível visualizar a interface do Google Colab. Há um espaço de codificação já preenchido onde o programa solicita ao usuário um dado para processar e devolver um resultado diferente. Na linha inicial, a variável “a” é empregada para guardar a resposta do usuário à indagação “Vamos descobrir em que ano você nasceu, informe a sua idade:”. Já que as variáveis recebidas pelo comando input são interpretadas como strings (textos), na linha subsequente ocorre a

transformação da variável “a” em um número inteiro, renomeando-a para “b”. Na terceira linha, a variável c é utilizada para calcular o intervalo entre o ano corrente, 2024, e o ano de nascimento do usuário, que é inferido pela subtração da idade do usuário (entrada) de 2024. A quarta linha contém a instrução de saída de dados, exibindo uma mensagem informativa sobre o ano de nascimento do usuário, baseada no valor obtido na variável “c”.

O ambiente de trabalho do LEGO® SPIKETM PRIME é uma plataforma interativa e intuitiva, projetada para engajar usuários de todas as idades no aprendizado de conceitos fundamentais de programação e robótica. Com uma interface amigável e blocos de código visualmente distintos, o LEGO® SPIKETM PRIME permite que os usuários experimentem com a construção e programação de modelos robóticos de maneira lúdica e educativa. A interatividade é um ponto forte, pois a plataforma responde imediatamente às instruções de código, proporcionando um feedback visual e físico através dos movimentos dos robôs. Isso não só facilita a compreensão dos efeitos do código, mas também estimula a solução de problemas e o pensamento criativo. Na (figura 8), é possível visualizar a interface de programação do LEGO® SPIKETM PRIME no ambiente de codificação *Python*.

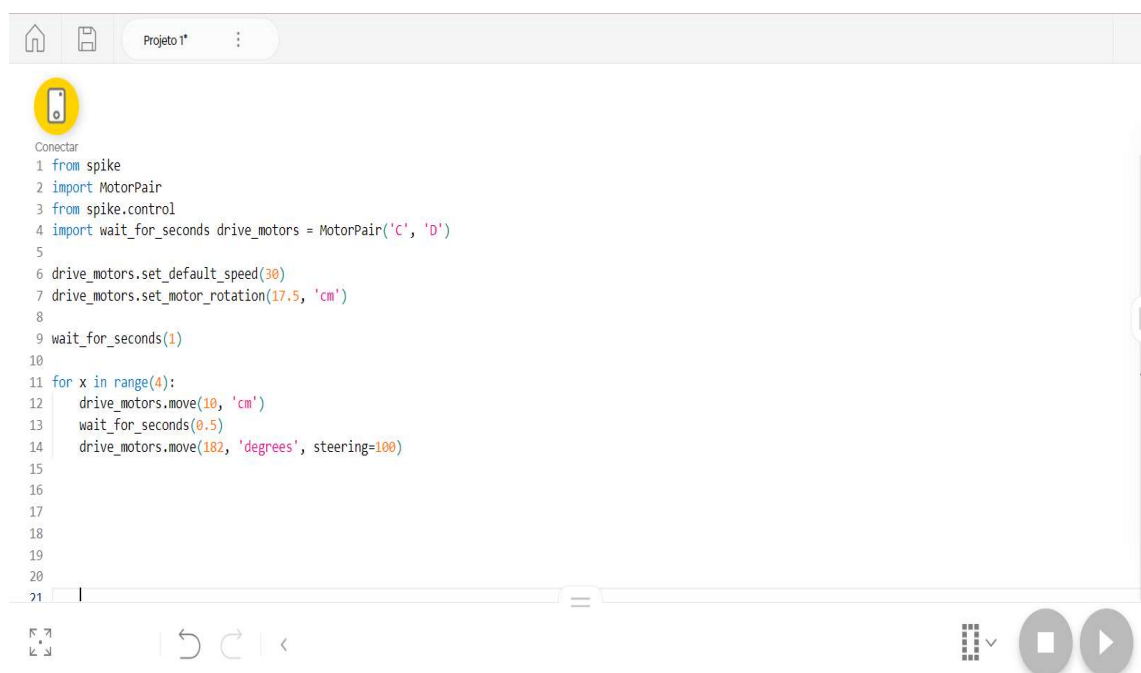
Figura 8 - Ambiente de aprendizagem para *Python*



Fonte: O autor (2024)

É possível visualizar a interface de programação do LEGO® SPIKETM PRIME. Há um espaço de codificação já preenchido onde o programa solicita ao robô para executar movimentos específicos. Na linha inicial, importa-se a classe “MotorPair” do módulo LEGO® SPIKETM PRIME, que é utilizada para controlar um par de motores. Já que as funções de movimento requerem parâmetros específicos, na linha subsequente define-se a velocidade padrão dos motores e a rotação correspondente a uma distância em centímetros. Na sequência do código, a função “wait\_for\_seconds” é usada para criar uma pausa na execução, seguida por comandos que movem os motores por uma distância ou rotação determinada, exemplificando a instrução de saída de dados que ativa o movimento do robô. Na (figura 9) a seguir, podemos observar o código em execução de um dos planos de aula, intitulado "Acampamento de Treinamento 1", parte da PRIME LESSON. Esse plano de aula é um dos muitos recursos educacionais oferecidos para o uso do LEGO® SPIKE™ PRIME, projetado para promover habilidades de programação e pensamento computacional.

Figura 9 - Execução do código “Acampamento de Treinamento 1” *Python*



Fonte: O autor (2024)



É crucial enfatizar a relevância da sintaxe correta nos comandos. Cada parêntese, aspa ou sinal é essencial para que o computador interprete o código adequadamente e execute as operações de maneira precisa.

## **DESCRIÇÃO DA SEQUÊNCIA DIDÁTICA**

As atividades desenvolvidas no contexto das aulas de Programação com blocos de palavras e *Python* foram estruturadas como missões, onde cada tarefa bem-sucedida resultava em uma pontuação registrada em um cartão de pontos, aplicando, assim, técnicas de gamificação no ambiente de aprendizado. Esse conceito envolve quantificação e registro numérico, essenciais para acompanhar o desempenho dos alunos.

Em nossa sequência de atividades e aulas, cada missão representava um conhecimento específico em programação e robótica que precisava ser assimilado pelos alunos, sendo sequenciadas e identificadas como Missão 1, Missão 2, Missão 3, e assim por diante.

Na Unidade 1, os alunos foram introduzidos aos conceitos iniciais, como "Construir um robô" e "Instalando o software & firmware". Essas primeiras missões foram fundamentais para garantir que todos os alunos estivessem familiarizados com os elementos básicos do LEGO® SPIKETM PRIME App. Os alunos foram organizados em grupos de 4 a 5 pessoas e, após uma breve apresentação do tema, iniciaram a montagem do robô base. Durante essa atividade, o professor circulou pela sala, oferecendo suporte e garantindo que todos os grupos estivessem preparados para os desafios subsequentes.

Na segunda unidade, o foco foi em "Conhecendo o software", "Gerenciando projetos", e "Visualizando dados dos sensores". Os alunos, organizados nos mesmos grupos, exploraram o ambiente de programação do LEGO® SPIKETM PRIME App. O professor demonstrou como utilizar as funcionalidades do aplicativo e orientou os alunos na prática dessas operações. Esta etapa foi essencial para capacitar os alunos a coletarem e analisarem dados, conceitos fundamentais da estatística e matemática aplicada.

Na terceira aula, as atividades concentraram-se em "Configurar a movimentação do seu robô", incluindo tarefas como "Andar em linha reta" e "Curvas com giroscópio". Os alunos aplicaram os conhecimentos adquiridos nas aulas anteriores para controlar com precisão o movimento de seus robôs. Nessa atividade, trabalharam com medidas angulares e distância, aplicando conceitos de geometria e cálculo de trajetórias. O professor acompanhou de perto a execução dessas tarefas, oferecendo feedback e auxiliando na correção de eventuais erros. Essa aula foi crucial para o desenvolvimento das habilidades práticas necessárias para as atividades de robótica.

Na quarta aula, os alunos foram introduzidos aos conceitos de "Pseudocódigo" e "Comentando o código". Essa etapa ajudou os alunos a planejar seus algoritmos de forma estruturada e a documentar suas soluções, um passo vital na transição para a programação mais complexa. Os alunos aplicaram esses conceitos na documentação de seus programas, com o professor fornecendo orientações e destacando a importância de um código bem estruturado e documentado.

Na quinta aula, as missões levaram os alunos a explorarem os sensores do KIT LEGO® SPIKETM PRIME, como o "Sensor de força", "Sensor de cor", e "Sensor de distância". Durante essa fase, os alunos aprenderam a utilizar esses sensores para que o robô pudesse interagir com o ambiente. O professor orientou os grupos na programação dos sensores e na integração de dados ao projeto dos alunos, aumentando a complexidade e a interatividade dos robôs construídos.

Na sexta aula, os alunos foram introduzidos aos conceitos de loops e condicionais, utilizando "Blocos de repetição", "Blocos de som & música", "Blocos de luz", e o "Bloco se então". Durante essa aula, os alunos criaram programas mais dinâmicos e responsáveis, aplicando os conceitos apresentados. Aqui, eles trabalharam com lógica condicional e operações booleanas, essenciais para a criação de algoritmos. O professor forneceu suporte contínuo, ajudando os alunos a entenderem as aplicações práticas desses conceitos na criação de programas interativos.

Na sétima aula, os alunos foram desafiados com tarefas práticas que consolidaram o conhecimento adquirido. As atividades incluíram "Detecção de

interrupção" e "Segue linha", culminando nos "Desafios Finais". Os grupos trabalharam de forma colaborativa para resolver os desafios propostos, aplicando todo o conhecimento adquirido até aquele momento. Em particular, na atividade "Segue linha", os alunos aplicaram o conceito de proporções ao programar um "Seguidor de linha proporcional", onde ajustavam dinamicamente os parâmetros do robô com base nas leituras do sensor.

Na oitava aula, os alunos começaram a trabalhar com conceitos mais abstratos, essenciais para a programação em *Python*, como eventos, "Sincronização de eventos", e variáveis. Esses conceitos envolvem o uso de estruturas matemáticas e a manipulação de dados que são fundamentais para o desenvolvimento de algoritmos complexos. Os alunos aplicaram esses conceitos em seus projetos, com o professor orientando sobre as melhores práticas para lidar com abstrações na programação.

Na nona aula, foram introduzidas técnicas avançadas de programação em *Python*, como "Alinhando com a linha" e "Seguidor de linha proporcional". Os alunos aplicaram essas técnicas, que envolvem cálculos de ajustes proporcionais e a resposta dos robôs com base em dados sensoriais. O professor revisou as atividades realizadas, fornecendo feedback detalhado sobre o desempenho dos alunos e destacando as melhores práticas em programação avançada.

Na décima e última aula, o foco foi em ensinar técnicas de depuração e confiabilidade para melhorar a robustez dos programas desenvolvidos. Os alunos aprenderam e aplicaram técnicas de "Depuração (debug)" e como melhorar a confiabilidade dos seus programas. O professor guiou os alunos durante a fase de testes e ajustes, discutindo a importância da depuração no desenvolvimento de software e como garantir que os programas funcionem de forma confiável em diferentes situações.

Este planejamento didático permitiu que os alunos progredissem de forma estruturada, começando com os fundamentos da montagem e programação com blocos e avançando para habilidades complexas de programação em *Python*, utilizando o LEGO® SPIKETM PRIME App como uma ferramenta poderosa para aprendizado prático e interativo.

## RESULTADOS E DISCUSSÃO

Como conclusão do trabalho de pesquisa, os seguintes resultados foram obtidos: As missões 5, 7 e 9 destacaram-se como os desafios mais complexos, sendo as únicas que não foram resolvidas por todas as equipes. Nos demais desafios, todas as equipes conseguiram concluir as missões, resultando na pontuação final apresentada no (tabela 4). Esses resultados sugerem que, embora a maioria das missões tenha sido acessível para os alunos, algumas exigiram um nível mais avançado de compreensão e habilidades em programação, indicando áreas para futuras revisões e aprimoramentos no currículo. Contudo, o objetivo principal que era o contato e aquisição de conhecimento foram alcançados pelos educandos.

Tabela 4 – Resultado dos questionários aplicados em cada aula.

<b>Equipes</b>	<b>Resultado geral</b>
RobMuts	90
PinkBots	55
Mamutes	85
RobotMar	70

Fonte: O autor (2024)

Ao analisar os resultados computados a partir da (tabela 4), observou-se que, quando questionados se os conhecimentos adquiridos poderiam ser aplicados em outras atividades, a maioria dos alunos demonstrou insegurança, com 8 dos 10 desafios, optando por "Não tenho certeza." No entanto, ao serem perguntados sobre a continuidade das aulas de robótica, todos os alunos expressaram um desejo claro de que elas continuassem mostrando um grande entusiasmo pelo conteúdo oferecido.

A partir das observações diretas, notou-se que os alunos se divertiam ao tentar resolver os problemas, especialmente devido ao aspecto lúdico introduzido pelas aulas. Contudo, esse mesmo elemento de jogo parece ter incentivado uma abordagem mais apressada, onde os alunos priorizavam a velocidade em detrimento de uma análise cuidadosa e estratégica dos desafios. Assim, embora a gamificação tenha motivado os alunos, ela também desviou o foco da resolução correta dos desafios, levando-os a adotar uma atitude mais impulsiva, com maior uso de tentativa e erro.

As respostas dos questionários estruturados também indicaram que os alunos apreciaram o aspecto divertido das aulas e recomendariam as aulas de robótica a outros colegas. Todos concordaram que as aulas deveriam continuar, o que reforça o valor dessas atividades no engajamento dos alunos. No entanto, ficou claro que, na percepção dos participantes, os conhecimentos de lógica de programação trabalhados durante as aulas de robótica não foram vistos como diretamente aplicáveis a outras áreas do conhecimento. Isso sugere que as aulas precisam incorporar de forma mais explícita elementos de interdisciplinaridade, especialmente considerando a faixa etária dos alunos, de 14 a 16 anos, onde a conexão entre diferentes áreas do conhecimento ainda não está claramente estabelecida. Esse aspecto foi subestimado no planejamento das aulas, indicando uma área para aprimoramento em futuros projetos educacionais.

## **CONSIDERAÇÕES FINAIS**

O trabalho de pesquisa, com enfoque teórico-prático e utilizando um estudo de caso como técnica de investigação, demonstrou ser altamente eficaz ao permitir que os conceitos teóricos fossem aplicados na prática, dentro de um contexto específico. Os resultados indicam que aulas práticas envolvem os alunos em um processo de imersão e vivência das teorias. No caso da robótica educacional, ficou evidente como os alunos aprendem uns com os outros, em seu próprio ritmo e motivados pelo interesse despertado pela novidade, que neste contexto é a introdução à lógica de programação.

Na tentativa de superar seus colegas, os alunos tendem a abandonar o processo de análise e planejamento, buscando resolver os problemas de maneira apressada e por tentativa e erro. Isso não garante que os alunos realmente compreendam a solução, mas sim que um de seus “chutes” acabe sendo acertado. Essa tendência foi confirmada quando solicitamos que os alunos reescrevessem suas respostas sem a pressão da competição, realinhando os blocos do programa ou transcrevendo o algoritmo em papel. Poucos conseguiram expressar novamente suas respostas de forma clara.

A utilização de uma linguagem visual, como os blocos de programação, para o ensino de lógica de programação, mostrou-se extremamente eficiente, pois concentra o processo de criação de algoritmos nos conceitos fundamentais, sem a preocupação com a sintaxe textual, permitindo uma maior ênfase no raciocínio lógico correto.

Recomenda-se que a gamificação não seja utilizada em todas as aulas. É importante esperar até que os conceitos essenciais estejam bem assimilados pelos alunos antes de introduzir a competição nas atividades. No entanto, manter a gamificação como uma estratégia de ensino é valioso, pois mesmo em um ambiente de aprendizagem já motivador, como o proporcionado pela robótica educacional, as aulas adicionam alegria e fluidez ao trabalho em equipe, promovendo a colaboração natural entre os alunos.

Os resultados obtidos a partir da aplicação do questionário estruturado na tabela 3 mostram que os alunos não percebem claramente que estão utilizando outros conhecimentos ou habilidades além daquelas ensinadas nas aulas de programação com robótica educacional. Portanto, é necessário contextualizar os conteúdos focados no funcionamento e comandos dos robôs com temas de outras disciplinas, como ciências e matemática, para incentivar uma conexão interdisciplinar clara. Revisar ou introduzir tópicos relacionados pode ser uma abordagem eficaz; por exemplo, ao programar utilizando comparações, seria adequado revisar conceitos como razão e proporção, geometria plana (cálculo de área do círculo e comprimento). Assim como outros aspectos.

Concluiu-se que as aulas de robótica educacional contribuíram para o ensino de lógica de programação, utilizando uma linguagem de programação visual em blocos, são altamente eficientes e empolgantes. Elas contribuem significativamente para o desenvolvimento do raciocínio lógico sequencial, promovem o trabalho em equipe e proporcionam uma experiência prática dos conteúdos abordados em sala de aula. Isso resulta em uma aprendizagem significativa para os alunos, pois colocam em prática habilidades e conhecimentos na solução de problemas, ao mesmo tempo que o aspecto lúdico gera engajamento e favorece a socialização entre os alunos.



## REFERÊNCIAS

ARIMOTO, M. M.; OLIVEIRA, T. D. O. **Dificuldades no processo de aprendizagem de programação de computadores: um survey com estudantes de cursos da área de computação**. In: XXVII Workshop sobre Educação em Computação, 2019, Belém. Anais [...]. Belém: [s.n.], 2019. p. 244-254.

BACICH, L; MORAN, J. **Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática**. Penso Editora, 2018.

BENNEDSEN, J.; CASPERSEN, M. E. **Failure rates in introductory programming: 12 years later**. ACM Inroads, New York, v. 10, n. 2, p. 30-36, 2019.

BORBA, M. de C.; ALMEIDA, H. R. F. L. de; GRACIAS, T. A. de S. **Pesquisa em ensino e sala de aula: diferentes vozes em uma investigação**. Belo Horizonte: Autêntica, 2018.

BORBA, M. de C.; ALMEIDA, H. R. F. L. de; GRACIAS, T. A. de S. **Pesquisa em ensino e sala de aula: diferentes vozes em uma investigação**. 2. ed. Belo Horizonte: Autêntica Editora, 2019. (Coleção Tendências em Educação Matemática).

BOSSE, Y.; GEROSA, M. A. **Why is programming so difficult to learn?: Patterns of Difficulties Related to Programming Learning Mid-Stage**. ACM SIGSOFT Software Engineering Notes, New York, v. 41, n. 6, p. 1-6, 2016.

BRACKMANN, C. P. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica**. 2017. 226 f. Tese (Doutorado) - Programa de Pós-graduação em Informática na Educação, Centro de Estudos Interdisciplinares em Novas Tecnologias na Educação, UFRGS, Porto Alegre, 2017.

BRASIL. Ministério da Educação e Cultura. **LDB - Lei nº 9394/96, 20 dezembro de 1996**. Estabelece as diretrizes e bases da Educação Nacional. Brasília: [s.n.], 1996.  
BRASIL. Secretaria de Educação Média e Tecnológica. **Parâmetros Curriculares Nacionais: Ensino Médio. Parte III – Ciências da Natureza, Matemática e suas Tecnologias**. Brasília: MEC/SEMT, 2000.

BUNDY, A. Computational Thinking Is Pervasive. **Journal of Scientific and Practical Computing Noted Reviews**, v. 1, n. 2, p. 67–69, 2007.

CAMBRUZZI, E.; SOUZA, R. **Robótica Educativa na aprendizagem de Lógica de Programação: Aplicação e análise**. Anais do Workshop de Informática na Escola, Porto Alegre, v. 21, n. 1, p. 21-28, 2015. DOI: 10.5753/cbie.wie.2015.21.

CAMPOS, F. R. Robótica Educacional no Brasil: Questões em Aberto, Desafios e Perspectivas Futuras. **RIEEE – Revista Ibero-Americana de Estudos em Educação**, Araraquara, v. 12, n. 4, p. 2108-2121, out./dez. 2017. DOI: 10.21723/riae.v12.n4.out./dez.2017.8778.

CARNEIRO, H. G. S.; JUNIOR, A. J. S. **Conhecimento de Robótica e de Matemática na formação inicial de professores no Estágio Supervisionado**. Educação Matemática Debate, Montes Claros, v. 7, n. 13, p. 1-20, 2023. DOI: 10.46551/emd.v7n13a20.

CASTRO, A. de. O uso da programação scratch para o desenvolvimento de habilidades em crianças do ensino fundamental. **Revista Tecnologias na Educação**, Ponta Grossa, v. 9, n. 19, 2017. Disponível em: <http://tecedu.pro.br/ano9-numerovol19/>. Acesso em: 18 mai. 2024.

CHAVES, C. D. **O uso de atividades de robótica e Linguagem de Programação para o desenvolvimento do Pensamento Computacional**. Presidente Prudente, 2023.

FEISTLER, T. N. F.; SCHEFFER, D. da C. D.; SILVEIRA, D. P. da. **A Robótica Pedagógica como Ferramenta na Aprendizagem da Educação Infantil e Ensino Fundamental I: Um Estudo em Referenciais Bibliográficos**. In: Congresso Nacional de Educação Infantil e Ensino Fundamental I, Alta Cruz, 2020, p. 221-232.

GIL, A. C. **Métodos e técnicas de pesquisa social**. 7. ed. São Paulo: Atlas, 2019.

LEGO. **The Lego, Mindstorms EV3**. Disponível em: <https://www.lego.com/en-us/mindstorms>. Acesso em: 05 mai. 2024.

LEGO. **The Lego, Spike Prime**. Disponível em: <https://spike.legoeducation.com/prime/project>. Acesso em: 15 mai. 2024.

LIMA, C. C. de; PREUSS, J. O. Ferramentas Online na Aprendizagem de Programação de Computadores no Contexto do Ensino Remoto. **Revista Brasileira de Informática na Educação**, v. 31, p. 790-813, 2023. DOI: 10.5753/rbie.2023.2867. Disponível em: <https://journals-sol.sbc.org.br/index.php/rbie/article/view/2867>. Acesso em: 30 jul. 2024.

LIMA, F. R. M. **LEGO® ZOOM: ferramenta para obtenção de dados experimentais na Física para o Ensino fundamental**. 2017.

LIMA, J. R. T.; FERREIRA, H. S. **Uma Revisão das Produções Científicas Nacionais sobre o uso da Robótica no Ensino da Física**. In: X Encontro Nacional de Pesquisa em Educação em Ciências, Águas de Lindóia-SP, 2015, p. 1-8.

LÜDKE, M.; ANDRÉ, M. E. D. A. **Pesquisa em educação: Abordagens qualitativas**. São Paulo: EPU, 1986.

LÜDKE, M.; ANDRÉ, M. E. D. A. **A pesquisa em educação: abordagens qualitativas**. 2. ed. Rio de Janeiro: E.P.U., 2013.

LUZ, A. M. R.; ALVARES, B. A. **Física contexto & aplicações**. São Paulo: Scipione, 2013.

MALHEIROS, A. P. dos S. **Educação matemática online: a elaboração de projetos de modelagem**. 2008.

MAIA, L. D. O.; SILVA, V. J. da; ROSA, R. E. V. de S.; LUCENA JUNIOR, V. F. de; QUEIROZ NETO, J. P. de. **A Robótica como Ambiente de Programação Utilizando o Kit Lego Mindstorms**. In: Simpósio Brasileiro de Informática na Educação, 2008, Brasil.

MARQUES, D.; COSTA, L.; SILVA, M.; REBOUÇAS, A. **Atraindo Alunos do Ensino Médio para a Computação: Uma Experiência Prática de Introdução a Programação utilizando Jogos e Python**. In: Workshop de Informática na Escola, 2011.

MORAES, M. C. **Ambientes de aprendizagem como expressão de convivência e transformação**. In: MORAES, M. C.; BATALLOSO NAVAS, J. M. (Orgs). Complexidade e transdisciplinaridade em educação: teoria e prática docente. Rio de Janeiro: Wak Editora, 2010.

PERIN, A. P.; CAMPOS, C. R. **Interfaces entre Modelagem Matemática, Raciocínio e Pensamento Estatístico**. Educação Matemática Debate, Montes Claros, v. 4, n. 10, p. 1-22, 2020. DOI: 10.46551/emd.e202032. Disponível em: <https://www.periodicos.unimontes.br/index.php/emd/article/view/2724>. Acesso em: 30 jun. 2024.

SIQUEIRA, P. V. R. **Volta, Lin!: Brinquedo maker**. UFRJ, 2021. 213f.

SILVA, M. A. de F. da; OLIVEIRA, M. **A Robótica Educacional na Perspectiva das Metodologias Ativas**. In: WORKSHOP DE INFORMÁTICA NA ESCOLA (WIE), 25., 2019, Brasília. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019, p. 1289-1293. DOI: 10.5753/cbie.wie.2019.1289.

SILVA NETO, V. J. da; BONACELLI, M. B. M.; PACHECO, C. A. O Sistema Tecnológico Digital: inteligência artificial, computação em nuvem e Big Data. **Revista Brasileira de Inovação**, v. 19, p. e0200024, 2020. DOI: 10.20396/rbi.v19i0.8658756.

VIEIRA, K. D.; HAI, A. A. O pensamento computacional na educação para um currículo integrado à cultura e ao mundo digital. **Acta Scientiarum. Education**, v. 45, n. 1, p. e52908, 6 out. 2022.

WING, Jeannette M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33, 2006. ISSN 00010782.

WING, Jeannette M. **Computational thinking and thinking about computing**. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 366, n. 1881, p. 3717–3725, 2008. ISSN 1364503X.

YIN, Robert K. **Estudo de caso: planejamento e métodos**. 5. ed. Porto Alegre: Bookman, 2015.

#### **AUTORES:**

**Flávia Gonçalves Fernandes**, Professora do Ensino Básico, Técnico e Tecnológico no IFMS – Campus Dourados, atua nas áreas de Informática, Desenvolvimento Web e Design Gráfico. É bacharel em Engenharia da Computação, mestre em Engenharia Biomédica, doutora em Ciências Exatas e Tecnológicas, com pesquisa voltada a métodos teórico-computacionais em ciência de materiais, e possui especializações em formação pedagógica e constelação sistêmica familiar.

**Rodrigo Felipe da Silva Mendes**, Professor, licenciado em Matemática e mestre em Modelagem Matemática e Computacional, possui especializações voltadas à educação básica, robótica educacional e uso de tecnologias no ensino. Atua no desenvolvimento de projetos inovadores com programação e robótica, integrando recursos tecnológicos ao currículo escolar, com experiência em pesquisa, publicações científicas e planejamento educacional estratégico.