

SKYE: Proposta de uma Linguagem de Programação Estruturada em Blocos para o Ensino Básico

**Vinicius M. L. Santos¹, Lidy Emanuelle M. Santos¹, Carlos A. O. de Freitas¹,
Anacilia M. C. de A. P. Vieira¹**

¹Instituto de Ciências Exatas e Tecnologia - Universidade Federal do Amazonas
(UFAM)
Rua Nossa Senhora do Rosário, 3863 - CEP 69.103-128 - Itacoatiara - AM

{vinicius.santos, lidy.santos, carlosfreitas,
anaciliacavalcante}@ufam.edu.br

Abstract. *Technology demands qualified professionals in computing, but basic education still lacks the development of computational thinking, especially in low socioeconomic contexts. The difficulty of abstraction by students and educators goes beyond the lack of resources. To address this challenge, this work proposes Skye, a visual programming language in blocks. The tool aims to facilitate the teaching of computational and mathematical concepts, allowing the creation of low-level algorithms in an intuitive and accessible way.*

Resumo. *A tecnologia exige profissionais qualificados em computação, mas o ensino básico ainda carece do desenvolvimento do pensamento computacional, especialmente em contextos socioeconômicos baixos. A dificuldade de abstração por alunos e educadores vai além da falta de recursos. Para enfrentar esse desafio, este trabalho propõe a Skye, uma linguagem de programação visual em blocos. A ferramenta visa facilitar o ensino de conceitos computacionais e matemáticos, permitindo a criação de algoritmos de baixo nível de forma intuitiva e acessível.*

1. Introdução

A tecnologia tem se estabelecido como um recurso essencial que tem um impacto significativo em várias áreas da vida diária, desde a indústria até a ciência. A crescente necessidade de profissionais com habilidades e competências em computação reflete a importância desse campo na sociedade contemporânea. No entanto, o ensino do pensamento computacional nas escolas ainda enfrenta obstáculos consideráveis, especialmente em comunidades com poucos recursos econômicos. Esses desafios vão além da falta de financiamento adequado ou do desinteresse dos alunos. Muitas vezes, estão relacionados à dificuldade de compreensão e abstração de conceitos tecnológicos, afetando tanto os professores quanto os estudantes.

Conforme destacado por Wing (2006 p.34), o conceito de pensamento computacional é um método que “se concentra em resolver problemas e em entender o comportamento humano, inspirando-se em conceitos fundamentais da ciência da computação”. Considerando o ambiente de sala de aula, desde o jardim de infância, o pensamento computacional pode ser uma técnica valiosa de solução de problemas que beneficia os alunos diante de variados desafios na vida futura. No entanto, a aquisição

de ferramentas ou o uso isolado delas têm eficácia limitada na motivação de alunos e estudantes.

O pensamento computacional é um conceito fundamental que se baseia no poder e nos limites dos processos de computação, sejam eles executados por humanos ou máquinas. Segundo Wing (2021 p.2-4), “o pensamento computacional confronta o enigma da inteligência da máquina: O que pode o ser humano fazer melhor do que os computadores? e O que é que os computadores podem fazer melhor do que os humanos? Mais fundamentalmente aborda a questão: O que é computável?”. Este conceito envolve a resolução de problemas, a concepção de sistemas e a compreensão do comportamento humano, utilizando conceitos fundamentais da ciência da computação. Wing (2021, p.2-4) argumenta que o pensamento computacional “é uma capacidade fundamental para todos, e não apenas para cientistas da computação. Ele deve ser adicionado às competências analíticas básicas de cada criança, junto com a leitura, escrita e aritmética. A informática e os computadores facilitam a disseminação do pensamento computacional, que inclui uma variedade de ferramentas mentais refletindo a amplitude das ciências da computação”.

2. Objetivos

O presente trabalho propõe a implementação da Skye, uma linguagem de programação visual baseada em blocos, como uma alternativa para democratizar o acesso ao pensamento computacional no ensino básico. A Skye foi desenvolvida para explorar conceitos científicos e matemáticos com a finalidade de capacitar os alunos e professores a construírem algoritmos de baixo nível por meio de uma interface amigável de blocos arrastáveis. A expectativa quanto a essa abordagem envolve não apenas reduzir a complexidade do ensino de algoritmos, mas também criar um ambiente mais fácil e motivador para os alunos. A ideia é que a Skye seja usada em ambientes formais e não formais de sala de aula para estender o ensino da programação em termos de alcance e eficácia.

3. Fundamentação Teórica

O desenvolvimento do pensamento computacional no ensino básico é um objetivo educacional crucial, considerando a crescente importância da tecnologia na sociedade contemporânea. De acordo com Resnick et al. (2009), o uso de ferramentas de programação visual, como o Scratch, tem se mostrado eficaz na introdução de conceitos de programação para crianças e adolescentes. O Scratch, desenvolvido pelo MIT Media Lab permite que os usuários programem arrastando e soltando blocos de código em vez de digitar comandos de texto, o que simplifica a compreensão de lógica e algoritmos, onde essa abordagem reduz a barreira de entrada para iniciantes, tornando a programação mais acessível e menos intimidante. Essas ferramentas facilitam a compreensão de lógica e algoritmos ao permitir a manipulação de blocos de código em uma interface intuitiva e amigável.

O estudo de Papert (1980) reforça essa abordagem ao argumentar que a construção do conhecimento é melhor facilitada quando os alunos podem interagir com materiais concretos de maneira lúdica e exploratória.

Por conseguinte, uma abordagem estruturada é necessária, que inclua ferramentas ao lado de métodos pedagógicos eficazes para manter o público-alvo envolvido, tal como mostrado no estudo de Stella (2016), onde foram analisados a utilização de recursos tecnológicos no ensino fundamental com a introdução de conceitos de programação utilizando o Scratch por meio de atividades lúdicas, no qual a avaliação do projeto mostrou que as crianças aprovaram a utilização dos recursos, mas tendiam a se interessar mais por atividades que envolviam a gravação com voz, imagens e fotos.

4. Metodologia

A metodologia utilizada neste trabalho é baseada no modelo cascata dividida em 7 etapas, uma abordagem sequencial para o desenvolvimento de sistemas, originalmente descrita por Royce (1970). O modelo cascata organiza o processo de desenvolvimento em uma série de etapas lineares e sequenciais, cada uma das quais deve ser concluída antes que a próxima comece. A Figura 1 mostra o fluxograma da Metodologia.

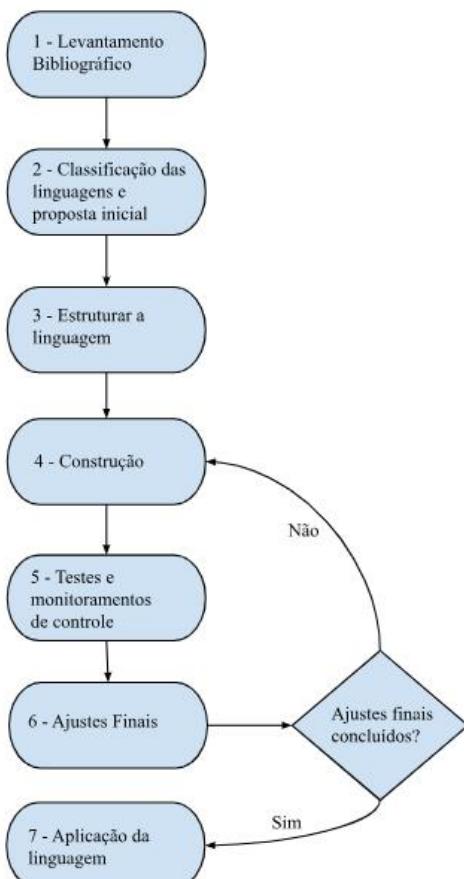


Figura 1. Fluxograma da Metodologia

A primeira etapa da metodologia consiste no levantamento bibliográfico, para este fim pesquisas em plataformas científicas como Scielo (Scientific Electronic Library Online), Google Acadêmico, Portal Capes foram realizadas para desenvolver a fundamentação teórica e fornecer uma estrutura conceitual para o desenvolvimento da linguagem de programação estruturada em blocos.

A etapa 2 é caracterizada pela classificação das linguagens obtida por meio do levantamento bibliográfico que permitiu também a identificação de trabalhos relacionados e possibilitou o desenvolvimento de uma proposta inicial com especificações para a construção da linguagem.

Na terceira etapa, a partir das especificações obtidas na etapa anterior deu-se início a documentação e estruturação da linguagem juntamente com o levantamento de requisitos iniciais.

A quarta etapa consistiu na construção da linguagem de programação estruturada com base nas pesquisas realizadas e nos requisitos iniciais levantados. Esta etapa possui três atividades distintas: (i) o desenvolvimento do módulo front-end (interface da linguagem com o usuário), (ii) o desenvolvimento do módulo back-end (interface da linguagem com os equipamentos e sensores e (iii) a integração destes dois módulos (front-end/back-end).

Após a etapa 4, deu-se início a quinta etapa que corresponde a execução de testes e monitoramento de controle, onde realizou-se testes de verificação para garantir que a linguagem de programação estruturada construída estava conforme a especificação da documentação dos requisitos.

A sexta etapa é caracterizada como uma etapa necessária para a correção de possíveis erros e não conformidades às especificações dos requisitos. Nesta etapa há um procedimento decisório, pois é nesta fase após a execução dos testes, que caso seja detectado erros significa que a linguagem não está “Ok”, então, retorna-se para a etapa de construção para fazer os ajustes e submeter novamente a linguagem a etapa de testes e monitoramento, em caso positivo, ou seja se os testes realizados indicarem que está tudo “Ok” então, passa-se para a próxima etapa da metodologia proposta.

Por fim, a sétima etapa corresponde à aplicação da linguagem de programação estruturada em blocos em alguns cenários para a validação da solução tecnológica desenvolvida.

5. Resultados e Discussão

A Skye foi proposta como uma linguagem de programação visual criada com o objetivo de facilitar o acesso do pensamento computacional para os mais variados cenários educacionais. Desta forma, a Skye foi desenvolvida para ser uma ferramenta de fácil acesso e utilização e usa uma interface intuitiva baseada em blocos móveis para que pessoas sem ou com pouca experiência com programação possam criar programas de forma simplificada.

De modo geral, a Skye é composta por dois módulos designados de módulo front-end e módulo back-end. O módulo front-end é desenvolvido utilizando HTML,

CSS e JavaScript, possibilitando assim a criação de uma interface de usuário que permita a manipulação dos blocos da Skye. O módulo back-end, por sua vez, é desenvolvido com Python e utiliza o framework Django. Isso permite coletar os dados provenientes de sensores e dispositivos, pois a linguagem permite a coleta de dados físicos, como temperatura e pressão, aumentando as aplicações de utilização da Skye em contexto educacional. Nas subseções seguintes mostra-se os resultados e discussões pertinentes a este trabalho.

5.1. A linguagem Skye

A Skye é uma ferramenta para a criação de algoritmos de baixo nível através de uma interface visual, permitindo que os usuários arrastem e soltem blocos de código em vez de escrever linhas de código manualmente.

Sendo assim, a Skye organiza seus blocos em várias categorias, facilitando a navegação e a seleção dos blocos necessários para a criação de programas. As categorias da Skye são mostradas na Tabela 1.

- Bibliotecas: Blocos que permitem a importação e a utilização de diversas bibliotecas de código, expandindo as funcionalidades disponíveis na linguagem.
- Lógica: Blocos que incluem operadores lógicos e estruturas condicionais, que permitem a criação de lógica de controle no programa.
- Laços de repetição: Blocos que permitem a execução repetida de um conjunto de instruções para facilitar a implementação de iterações.
- Variáveis: Blocos que permitem a criação, a modificação e a utilização de variáveis para armazenar e manipular dados no programa.
- Sensores: Blocos que permitem a leitura de dados físicos, como temperatura, umidade e distância.
- Funções: Blocos que permitem a definição e a chamada de funções personalizadas, promovendo a reutilização de código e a organização modular do programa.
- Conexão: Blocos que facilitam a comunicação com outros dispositivos, permitindo a troca de dados, blocos essenciais para o funcionamento da rede de dispositivos IoT.

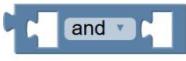
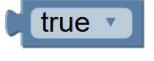
5.2. Dicionário de blocos

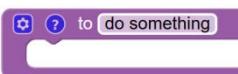
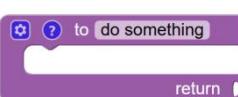
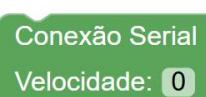
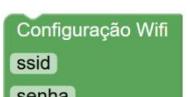
A Skye é formada por vários blocos distribuídos em categorias específicas, onde o dicionário de blocos aborda cada bloco com detalhes e possui as categorias: Bibliotecas, Lógica, Laços de repetição, Variáveis, Sensores, Funções e Conexão.

A Tabela 1 mostra os blocos da linguagem Skye, suas respectivas categorias e funcionalidades.

Tabela 1. Dicionário da linguagem

Bloco	Categoria	Descrição
Importar Sensor DHT	Bibliotecas	Bloco de importação da biblioteca DHT para os sensores DHT11 e DHT22.

	Bibliotecas	Bloco de importação da biblioteca ESP para placas ESP8266 com suporte à conexão wifi.
	Bibliotecas	Bloco de importação da biblioteca ESP para placas ESP32 com suporte à conexão wifi.
	Bibliotecas	Bloco de importação da biblioteca SoftwareSerial que permite a comunicação serial em qualquer pino digital do microcontrolador, para comunicar com múltiplos dispositivos seriais ao mesmo tempo.
	Logica	Bloco de inicialização permite que configure o hardware e prepare qualquer inicialização necessária, e a execução continua do algoritmo.
	Logica	Bloco de controle de fluxo, que permite executar blocos de código condicionalmente.
	Logica	Bloco de operadores aritméticos são usados para realizar operações básicas.
	Logica	Bloco de operadores lógicos utilizados para combinar expressões booleanas em condições mais complexas
	Logica	Bloco comum de negação de sentença.
	Logica	Bloco comum de valores booleanos(verdadeiro/falso).
	Laços de repetição	Bloco de repetição para iterar sobre uma sequência de elementos.
	Laços de repetição	Bloco de repetição para executar repetidamente um bloco de código enquanto uma condição especificada for verdadeira.
	Variáveis	Bloco comum para armazenar dados que podem ser manipulados e utilizados ao longo da execução.
	Sensores	Bloco do sensor DHT11 para coleta de dados de temperatura e/ou umidade.
	Sensores	Bloco do sensor ultrassónico HC-sr04 para a coleta de distância, com os parâmetros de pinagem trig e echo.

	Sensores	Bloco de som para fornecer feedback sonoro.
	Funções	Bloco de função que realiza um conjunto de instruções sem retornar um valor específico ao final da execução.
	Funções	Bloco de função que executa uma tarefa específica e retorna um valor como resultado desse processamento
	Funções	Bloco comum para controlar o fluxo do programa e retornar valores condicionalmente
	Conexão	Bloco de conexão serial com o dispositivo que recebe a velocidade da porta de conexão.
	Conexão	Bloco de conexão wifi que conecta um dispositivo à rede Wi-Fi usando os parâmetros ssid e senha.
	Conexão	Bloco de leitura serial com os dispositivos conectados.

5.3. Exemplificação da linguagem

Nesta subseção, demonstra-se a linguagem, destacando sua sintaxe e recursos através de um exemplo prático. A exemplificação visa facilitar a compreensão dos conceitos teóricos apresentados anteriormente, demonstrando como a linguagem pode ser aplicada em diferentes contextos.

5.3.1. Objetivo do experimento

O objetivo deste experimento é demonstrar a comunicação entre dois Arduino de forma serial, onde um Arduino atua como transmissor de dados de um sensor ultrassônico e o outro Arduino como receptor, exibindo os dados recebidos juntamente com informações de um sensor de temperatura e umidade (DHT11).

Para este experimento foram necessários os seguintes materiais: 2 placas Arduino (pode ser Uno, Nano, etc.), Sensor ultrassônico HC-SR04, Sensor de temperatura e umidade DHT11, Buzzer, Cabos jumpers e uma Protoboard (opcional).

5.3.2. Interface inicial

Logo ao acessar a interface gráfica da linguagem, tem-se acesso a barra superior de navegação que permite navegar entre o editor de blocos, o console da linguagem, a conexão e o dashboard.

No lado esquerdo da interface estão localizados os blocos divididos por categoria conforme mostrado no tópico 3.2. Por fim no canto inferior direito tem-se a lixeira para o descarte de blocos conforme mostra a Figura 2.

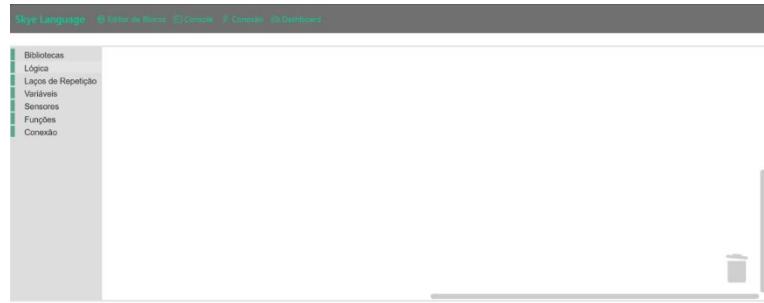


Figura 2. Interface inicial

5.3.3. Módulo receptor

Para construção do módulo receptor utilizou-se o bloco lógico de configuração e execução. Dessa forma, importa-se a biblioteca DHT para o uso do sensor DHT11 e, após a importação realiza-se a configuração da conexão serial com o bloco correspondente passando como parâmetro a velocidade 115200 e por fim usa-se o bloco de leitura de serial para receber os dados enviados pelo modulo transmissor (Figura 3).

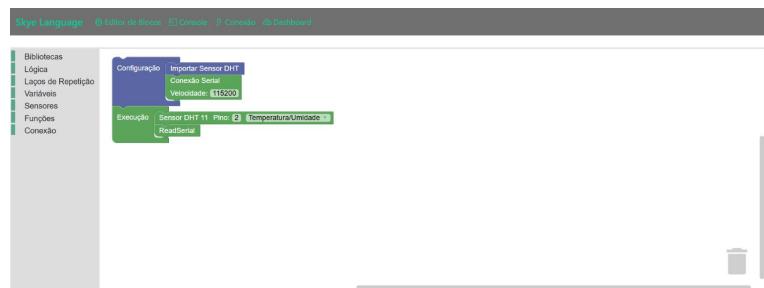


Figura 3. Construção do módulo receptor

Ao acessar o console pela barra de navegação pode-se copiar o código gerado, mas também há a possibilidade de o editar conforme demonstrado na Figura 4.

```
#include "DHT.h"
#define DHTTYPE DHT11

void setup() {
    const int DHTPIN = 2;
    DHT dht(DHTPIN, DHTTYPE);
    Serial.begin(9600);
    dht.begin();
}

void loop() {
    float humidity = dht.readHumidity();
    float temperature = dht.readTemperature();

    Serial.print("Umidade: ");
    Serial.print(humidity);
    Serial.print("\n");
}
```

Figura 4. Algoritmo do módulo receptor

5.3.4. Módulo transmissor

Na construção do módulo de transmissão utilizou-se bloco lógico de configuração e execução, após isso é realizado a configuração da conexão serial com o bloco correspondente passando como parâmetro a velocidade 115200. Ainda no bloco de configuração é criado a variável ‘duration’ que é usada como parâmetro para definir a

duração de 1000 milissegundos para a coleta da distância, passando para o bloco de execução é criado um fluxo com o bloco condicional ‘if ...do’ que caso a distância coletada pelo sensor ultrassônico seja igual ou inferior a 5 centímetros é acionado um buzzer conectado no pino 6 para produzir um som. Após a construção e acessar o console, tem-se o algoritmo resultante os parâmetros estabelecidos, conforme demonstrado nas Figuras 5 e 6.



Figura 5. Construção do módulo transmissor

```

void setup() {
    const int trigPin = 7;
    const int echoPin = 8;
    int duration;
    int distance;
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.begin(115200);
}

void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
}

```

Figura 6. Algoritmo do módulo transmissor

Finalizado a etapa de construção é necessário descarregar os algoritmos gerados nos Arduinos, e aguardar que a conexão seja estabelecida e por fim acessar o dashboard pela barra de navegação e aguardar que a coleta de dados seja recebida e mostrada conforme é demonstrado na Figura 7.

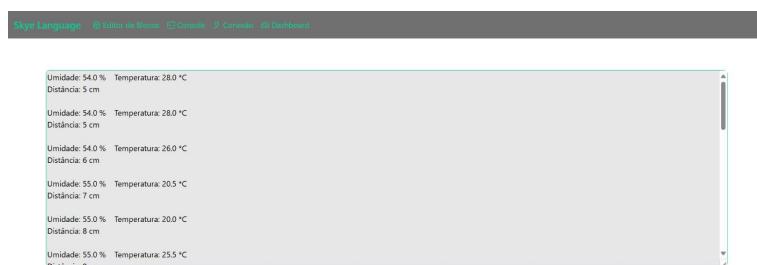


Figura 7. Tela de resultados de coleta

5.3.5. Experimento

A Figura 8 demonstra o esquemático do sistema montado. Nas Figuras 9 e 10 estão respectivamente o circuito montado e o resultado do experimento realizado com a linguagem.

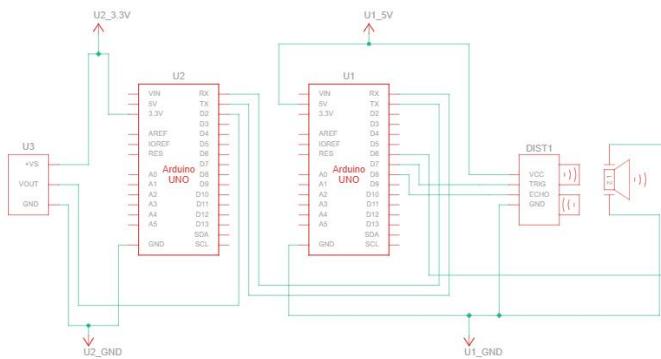


Figura 8. Esquemático do experimento

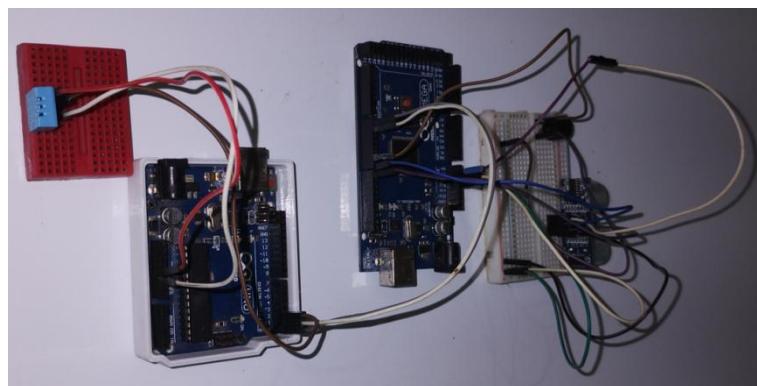


Figura 9. Circuito montado

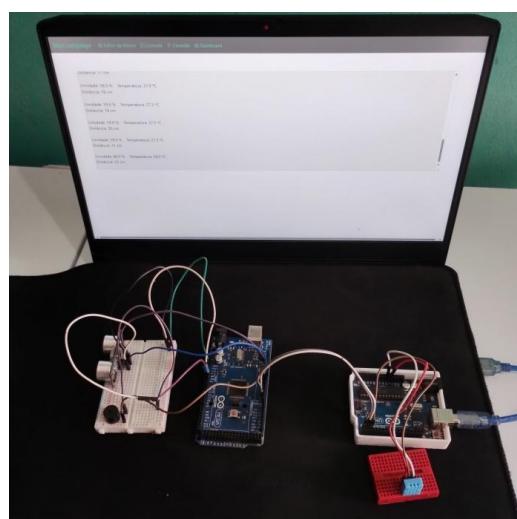


Figura 10. Experimento realizado

6. Considerações Finais

A proposta da linguagem de programação visual Skye mostrou-se uma aplicação interessante e promissora para a disseminação do pensamento computacional no ensino básico. A Skye oferece um ambiente simples e didático para a compreensão de conceitos fundamentais de programação e algoritmos de baixo nível, usando uma interface visual com blocos arrastáveis que a torna acessível a estudantes e professores, mesmo em ambientes com recursos limitados. A exemplificação da Skye mostra que a linguagem pode ser uma ferramenta de grande utilidade para o ensino de programação nos níveis mais adiantados.

O uso de blocos de código pode simplificar a compreensão de conceitos mais complexos, pode estimular a participação dos estudantes, e aprimorar o desenvolvimento de habilidades relacionadas à computação, à lógica e à matemática, fundamentais para suas carreiras futuras.

Em síntese, com a Skye, temos uma alternativa promissora na abordagem adotada para ensinar o pensamento computacional, uma vez que dispomos de uma plataforma intuitiva e simples, eficaz e que pode ser inserida em ambientes formais e informais de ensino. Além disso, caso o seu desenvolvimento proporcione uma ampliação das suas funcionalidades, pode ajudar na formação de uma nova geração de profissionais que, em ciências da computação, estão preparados para os desafios tecnológicos.

Referências

- Papert, S. (1980) *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.
- Resnick, M. e Maloney, J. e Monroy-Hernández, A. e Rush, S. e Eastmond, E. e Brennan, K. e Millner, A. e Rosenbaum, E. e Silver, J. e Silverman, B. e Kafka, Y. (2009) Scratch: Programming for All. *Communications of the ACM*, 52(11), 60–67.
- Royce, W. W. (1970) Managing the development of large software systems.
- Stella, A. L. (2016) Utilizando o pensamento computacional e a computação criativa no ensino da linguagem de programação Scratch para alunos do ensino fundamental. Dissertação, 2016 (Mestrado), Faculdade de Tecnologia – Universidade Estadual de Campinas. Limeira.
- Wing, J. M. (2006) Computational Thinking. *Communications of the ACM*, vol. 49, nº3. Disponível em: <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>. Acesso em: 20 jul. 2024.
- Wing, J. M. (2021) Pensamento computacional. *Educação e Matemática*, n. 162, p. 2-4.