

Brumu: Kit educativo de baixo custo para apoiar o desenvolvimento do pensamento computacional em crianças

Bruna Cristhine Amarilha Teles
Faculdade de Computação - FACOM
Universidade Federal de Mato Grosso
do Sul
Campo Grande, MS, Brasil
bruna.amarilha@ufms.br

Munir Souza dos Santos
Faculdade de Computação - FACOM
Universidade Federal de Mato Grosso
do Sul
Campo Grande, MS, Brasil
munir_souza@ufms.br

Luciana Montera Cheung
Faculdade de Computação - FACOM
Universidade Federal de Mato Grosso
do Sul
Campo Grande, MS, Brasil
luciana.montera@ufms.br

ABSTRACT

This work presents the development of Brumu, an educational kit designed to promote Computational Thinking (CT) in children aged 3 to 9. Inspired by the Cubetto educational robot, but with a significantly lower cost, Brumu combines a mobile application and a robot built with widely available components. The solution aims to democratize access to educational robotics, aligning with the principles outlined in Brazil's National Common Curricular Base (BNCC), which emphasizes the importance of CT in basic education.

The app, built with MIT App Inventor, is easy to use and allows kids to program the robot by dragging and dropping commands onto the screen. The bluetooth-controlled robot was built with an Arduino UNO and other components that are cheaper and easier to assemble. Brumu offers a hands-on, fun experience that helps kids learn programming while developing creativity, logical reasoning, and problem-solving skills.

Through this project, Brumu was successfully developed at approximately 91.76% lower cost than Cubetto, accompanied by a detailed manual for assembly and use. This kit represents a scalable and accessible alternative for educational institutions, paving the way for broader adoption of robotics in early education. Future work includes extending the platform with subject-specific maps, lesson plans and usability testing to further enhance its application in diverse educational contexts.

KEYWORDS

Pensamento Computacional, Robótica Educacional, Arduino UNO, MIT App Inventor.

1 INTRODUÇÃO

O termo Pensamento Computacional (PC) foi proposto pela primeira vez em 2006, por Jeannette Wing [1], que afirmou que as técnicas de abstração, resolução de problemas, e implementação algorítmica não deveriam ser de exclusividade dos cientistas da computação, mas sim de toda a população, que

deveria ter a oportunidade de desenvolver essa habilidade começando desde cedo, principalmente nas escolas.

O Pensamento Computacional consiste no uso das técnicas e fundamentos da Ciência da Computação, inicialmente aplicados para soluções de problemas teóricos e práticos com diversos fins, dentre eles o desenvolvimento de software, aplicado também em situações cotidianas por pessoas "comuns" e não por apenas cientistas, professores e acadêmicos. Em seu trabalho, [1] afirma que o PC é uma habilidade fundamental, assim como a leitura, a escrita e a aritmética e, por isso, deve ser estimulada nas crianças tal como essas outras competências.

A autora destaca também que muitos pais têm a visão de que desenvolver o pensamento computacional desde cedo só privilegia aqueles que desejam seguir na carreira da computação ou outro curso da área das ciências exatas. Todavia, Jeannette afirma que essa habilidade é plural e pode ser associada a diversas outras áreas, uma vez que pensar computacionalmente amplia as formas de compreender e resolver problemas e, também, que a tecnologia está associada aos mais diversos setores.

No Brasil, o desenvolvimento do PC é citado na Base Nacional Comum Curricular (BNCC), que é um documento normativo que descreve as "aprendizagens essenciais que todos os alunos devem desenvolver ao longo das etapas e modalidades da Educação Básica, de modo que tenham assegurados seus direitos de aprendizagem e desenvolvimento, em conformidade com o que preceitua o Plano Nacional de Educação (PNE)" [2].

Dentro da BNCC, o PC é definido como a capacidade de identificar, interpretar, estruturar, solucionar, avaliar e automatizar problemas e suas soluções de forma lógica e organizada, por meio da criação de algoritmos. Sendo assim, trata-se de uma competência essencial, que deve ser integrada em diferentes áreas do conhecimento, devendo ser iniciada a partir do Ensino Fundamental. Já no Ensino Médio, destaca-se a importância do uso de recursos como tecnologias digitais e aplicativos tanto para a investigação matemática como para dar continuidade ao desenvolvimento do PC, iniciado no Ensino Fundamental.

Atividades matemáticas como modelagem, resolução de problemas, pesquisa e desenvolvimento de projetos são citadas pela BNCC como fortes influenciadores no desenvolvimento não só do raciocínio lógico e demais fatores do letramento

matemático, mas também como pivôs do desenvolvimento do PC nos estudantes brasileiros.

A capacidade de compreender os números, geometria, probabilidade e estatística e álgebra também acaba contribuindo para o desenvolvimento do PC, pois é preciso traduzir situações que são descritas em forma de texto, na língua materna, para um outro formato como fórmulas, gráficos e tabelas. Além disso, esses procedimentos matemáticos têm muito em comum com implementações algorítmicas como o conceito de variável, identificação de padrões, importância da ordem de execução de tarefas e abstração de problemas [2].

Como estratégia para o desenvolvimento do PC, podemos citar a robótica educacional, que traz uma abordagem lúdica e diferente da tradicional aplicada nas salas de aula, onde os estudantes interagem com um robô ou uma ferramenta robotizada, para explorar e resolver problemas por meio da programação e posterior análise do comportamento do robô [4]. Esse tipo de estratégia é multidisciplinar, podendo ser associada a diversas outras áreas da educação, pois estimula a socialização, por meio do compartilhamento da ferramenta robotizada e explora a criatividade e a capacidade de aprender. A robótica educacional permite também que os professores expliquem conceitos complexos de forma lúdica e mais atrativa para seus alunos.

Um exemplo de robô voltado para a educação é o brinquedo Cubetto, que obteve um dos maiores financiamentos coletivos da história dentro da inovação educacional na tecnologia, cerca de \$1.596.457, sendo criado e comercializado pela empresa *Primo Toys*¹, que surgiu em 2013. De acordo com o próprio fabricante, o Cubetto tem como objetivo ensinar conceitos básicos de programação de computadores para crianças de 3 a 9 anos de idade [3].

O brinquedo é um kit com quatro partes, sendo elas um robô de madeira, blocos de codificação, tabuleiro de controle e um mapa, que juntos funcionam da seguinte maneira: o robô parte de um ponto no mapa e o desafio é que ele chegue em outro ponto (destino). Para isso é preciso dispor os blocos de codificação (que representam os comandos de percurso esquerda, direita, frente e função) no tabuleiro de controle de forma a ensinar/programar o robô para que este chegue ao seu destino.

No entanto, apesar de ser uma ferramenta educacional inovadora e bastante atrativa, seu custo é elevado², cerca de R\$2.200,00, o que pode limitar seu acesso a um público mais amplo e tornar a proposta de robótica educacional menos acessível.

Diante dos benefícios da robótica educacional e da importância do desenvolvimento do Pensamento Computacional na educação básica, o presente trabalho apresenta uma alternativa de baixo custo ao Cubetto, juntamente com um manual de montagem e configuração para que a solução possa ser replicada, com o propósito de tornar a robótica educacional cada vez mais acessível a todos, e promover o desenvolvimento do Pensamento Computacional em todo o tipo de ambiente educacional. A solução proposta (*software* e *hardware*) recebe o nome de Brumu.

As seções seguintes deste artigo estão assim organizadas: na Seção 2 são apresentados trabalhos que também abordam o desenvolvimento de robôs educacionais de baixo custo. Na Seção 3 é descrita a metodologia utilizada no desenvolvimento deste trabalho. A Seção 4 detalha o desenvolvimento do kit educacional proposto. Os resultados alcançados são apresentados na Seção 5. Por fim, na Seção 6, são feitas as considerações finais.

2 TRABALHOS RELACIONADOS

Nesta seção, serão apresentados dois estudos identificados na revisão de literatura que também buscam desenvolver robôs educacionais de baixo custo, além de mais uma ferramenta amplamente utilizada e comercializada no ensino de robótica. O objetivo é destacar as principais diferenças entre o kit educacional desenvolvido neste trabalho e esses outros disponíveis, permitindo a identificação de melhorias, desafios e sugestões baseadas em trabalhos anteriores, reforçando a importância de democratizar o acesso à robótica educacional.

Em [4] é apresentada uma plataforma que oferece uma alternativa de baixo custo ao Cubetto, solução esta que utiliza materiais e componentes mais acessíveis tanto para o desenvolvimento do robô quanto para a interface de controle. Assim como no Cubetto, a comunicação entre o robô e o painel de controle é feita via *bluetooth*. A principal diferença entre este produto e o Brumu, está no tabuleiro e nas peças de controle que são substituídos por uma aplicação *mobile*, reduzindo ainda mais os custos de construção, porém mantendo o mesmo propósito educacional.

O kit didático chamado ROME ([5]), consiste em um projeto *open-source hardware* onde os autores desenvolveram e disponibilizaram para livre reprodução um conjunto de componentes de robótica móvel, bem como arquivos e instruções necessárias para reprodução do modelo proposto (projeto mecânico e eletrônico).

Apesar do custo de produção do ROME ser bastante baixo (cerca de R\$ 100,00 segundo os autores), sua fabricação envolve a produção da maioria dos componentes utilizados no robô. Por exemplo, toda a estrutura do modelo é feita em MDF, mas para obter as peças necessárias, é necessário cortar a placa de MDF com uma máquina CNC de corte a laser e seguir o plano de corte disponibilizado pelos autores. Além disso, o modelo da placa de controle eletrônica do robô também foi desenvolvido pelos autores e precisa ser fabricado separadamente, enquanto o Brumu é totalmente formado por componentes e peças comercialmente disponíveis, o que simplifica consideravelmente sua construção.

Outra ferramenta comercial muito utilizada para o ensino de robótica educacional é o kit *LEGO Mindstorms* ([6]), composto por quatro tipos de componentes distintos: quatro sensores, três motores, um controlador central e diversas peças de lego para construir o robô conforme desejado. Esse kit, lançado em 1998, permite a criação de robôs que executam diversas tarefas por meio da combinação da funcionalidade de seus sensores junto à programação do controlador, que é feita através de sistema próprio, por meio de programação em blocos.

Embora o *LEGO Mindstorms* seja uma ferramenta robusta e amplamente reconhecida no mercado, seu uso é indicado para

¹ <https://www.primotoys.com/>

² valor informado pela representante da Primo Toys no Brasil (<https://www.oclubederobotica.com.br/>)

estudantes que já possuem algum conhecimento prévio de lógica de programação, para que assim consigam explorar as potencialidades que o robô proporciona. O próprio fabricante recomenda o produto para crianças a partir dos 10 anos de idade. Em contrapartida, o Brumu foi projetado para introduzir e desenvolver essas habilidades de lógica de programação desde cedo, atendendo crianças a partir dos três anos, da educação básica.

3 METODOLOGIA

Para o desenvolvimento da solução proposta, optou-se por uma abordagem integrada que combina um aplicativo para celular/*tablet* e um robô, construído com componentes amplamente encontrados no mercado nacional, que se comunicam via *bluetooth*. O objetivo é oferecer uma experiência de aprendizado prática, simples e de baixo custo, por isso a escolha de se desenvolver uma aplicação *mobile*, eliminando assim o custo que seria necessário para desenvolver uma versão física do tabuleiro de controle. Além disso, para montagem do robô, buscou-se utilizar produtos facilmente disponíveis para compra e de baixo custo.

A aplicação *mobile* foi desenvolvida utilizando a ferramenta MIT *App Inventor*³ por ser uma ferramenta online de fácil utilização, mesmo para iniciantes em programação. Desenvolvido e mantido por um grupo do *Massachusetts Institute of Technology* (MIT), o *App Inventor* foi projetado para tornar a criação de aplicativos móveis acessível a um público mais amplo, utilizando uma interface de programação visual baseada em blocos, que facilita o entendimento dos conceitos fundamentais de programação [7].

Por ter uma interface atrativa, intuitiva, motivadora e uma vasta documentação disponível, o *App Inventor* possibilita que os usuários criem seus aplicativos concentrando esforços na lógica do mesmo, sem precisar se aprofundar em alguma linguagem de programação [8]. Assim, o código do Brumu poderá ser facilmente ajustado ou expandido por seus usuários que assim o desejarem.

O aplicativo oferece também uma ferramenta de suporte, que pode ser baixada diretamente das lojas de aplicativos para dispositivos móveis. Essa ferramenta permite ao usuário visualizar, modificar e testar a construção de seu aplicativo em tempo real por meio de uma conexão *Wi-Fi*. Desta forma, um *QR code* é gerado no site oficial do *App Inventor* (plataforma de desenvolvimento) e, ao ser escaneado com o aplicativo de suporte, permite a conexão direta sem a necessidade de configurações, espaço extra no dispositivo ou requisitos técnicos adicionais [7].

No desenvolvimento do robô, o Arduino UNO⁴ foi utilizado como controlador central, programado para receber e executar os comandos enviados pelo aplicativo. Para garantir o sucesso do mesmo, foi realizado um estudo prévio sobre os principais microcontroladores disponíveis no mercado, com foco em como integrá-lo aos diversos componentes eletrônicos e mecânicos necessários para a montagem do robô. Com isso, a escolha do Arduino UNO como plataforma base foi fundamental, visto que

uma de suas principais vantagens em comparação com outras plataformas de desenvolvimento de microcontroladores, é sua facilidade de uso, permitindo que pessoas fora da área técnica aprendam rapidamente o básico e desenvolvam seus próprios projetos em um curto período de tempo [10]. Além disso, como o *hardware* e o *software* do Arduino são de código aberto, qualquer pessoa pode acessar, modificar e utilizar livremente o código, possibilitando realizar adaptações e melhorias contínuas.

Para a construção do robô, foram utilizados diversos componentes eletrônicos e mecânicos essenciais para seu funcionamento. O Arduino UNO R3 é a placa principal, responsável pelo controle de todas as operações do carrinho, permitindo a programação dos comandos que direcionam os motores e recebem comandos do aplicativo. O movimento é garantido por dois motores DC e uma Ponte H L298N, que possibilita ao Arduino controlar a direção e a velocidade dos motores, permitindo deslocamento para frente e mudanças de direção. A comunicação entre o Arduino e o aplicativo é realizada pelo módulo *bluetooth* HC-05, que permite o recebimento de comandos vindos do celular, facilitando o controle do robô e tornando sua interação mais intuitiva.

Para garantir energia suficiente para o funcionamento dos componentes, utilizou-se um suporte com quatro a seis pilhas AA, pois os motores necessitam de uma tensão de 5 a 8 volts para operar adequadamente. Como cada pilha fornece 1,5 volts, a combinação de quatro a seis pilhas garante a energia necessária. Além disso, a alimentação do Arduino Uno, do controlador Ponte H e do módulo *bluetooth* HC-05 também é realizada pelo mesmo suporte de pilhas, garantindo o funcionamento estável de todo o sistema. Foi adicionado também um botão de ligar/desligar para facilitar o uso. Todos esses componentes foram montados sobre um chassi de material acrílico, leve e resistente, proporcionando estabilidade ao robô.

Para a codificação do Arduino, foi utilizada a linguagem de programação C/C++, amplamente empregada na plataforma devido à sua capacidade de controle detalhado e à facilidade de integração com os recursos de hardware do Arduino. A programação utiliza funções e bibliotecas próprias que simplificam o controle dos pinos digitais e analógicos da placa, permitindo que o desenvolvedor programe desde simples sinais de acionamento de LEDs até o controle mais complexo de motores e sensores [11].

O Arduino IDE (*Integrated Development Environment*) foi a ferramenta escolhida para a criação, edição e compilação do código. Essa IDE é especialmente projetada para desenvolvimento com Arduino, fornecendo uma interface amigável que facilita a escrita de códigos e o acesso a bibliotecas integradas [11].

Após a escrita do código, o processo de compilação do Arduino IDE converte as instruções escritas em C/C++ em um código binário que o microcontrolador pode interpretar. Esse binário é, então, carregado na memória flash do Arduino UNO através de um cabo USB. Durante esse processo, a IDE utiliza a comunicação serial para transferir o código e, ao final, o microcontrolador começa a executar automaticamente as instruções carregadas. Dessa forma, o Arduino passa a realizar as operações programadas, como controlar o movimento de motores,

³ <https://appinventor.mit.edu/>

⁴ <https://www.arduino.cc>

receber comandos do módulo *bluetooth*, e outras funções definidas no código.

A Figura 1 ilustra o fluxo de interação entre o aplicativo Brumu, o Arduino UNO, e o carrinho. Nesse sistema, o aplicativo estabelece uma conexão *bluetooth* com o Arduino, permitindo que o usuário envie comandos de movimento diretamente do celular. Após estabelecer a conexão, o aplicativo transmite instruções, que são recebidas pelo módulo *bluetooth* acoplado ao Arduino. O mesmo processa esses comandos e os repassa ao controlador das rodas, que aciona os motores para realizar os movimentos desejados. Esse processo possibilita o controle dos movimentos do carrinho de forma eficiente e intuitiva, facilitando a interação do usuário com o robô educativo.

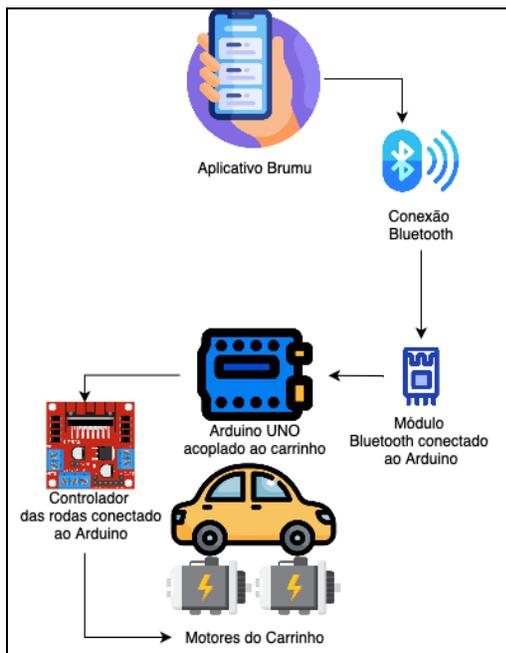


Figura 1: Fluxo de comportamento entre o Aplicativo e Robô/Carrinho.

4 DESENVOLVIMENTO

Esta seção descreve a aplicação da metodologia descrita na seção anterior na elaboração da solução proposta neste trabalho. A descrição é dividida em duas subseções: a primeira detalha o processo de desenvolvimento do aplicativo *mobile*, e a segunda aborda a montagem e a programação do carrinho/robô.

4.1 Aplicativo *Mobile*

O aplicativo desenvolvido foi projetado com interfaces simples e intuitivas, inspiradas nos componentes do tabuleiro de controle do Cubetto. A estrutura da aplicação consiste em três telas: Tela Principal, Tabuleiro Virtual e Tela de Informações.

A Tela Principal, apresentada na Figura 2 (a), exibe um menu contendo os botões Tabuleiro Virtual e Sobre o Projeto, os quais

direcionam para telas correspondentes. Como mostra a Figura 2 (b), a Tela do Tabuleiro Virtual é o núcleo da aplicação e disponibiliza ao usuário os elementos necessários à programação do robô, sendo eles, botão para conexão *bluetooth*; área destinada à disposição de comandos (peças); botão para envio de comandos ao robô; área para definição de funções; e as peças de comandos para frente, esquerda, direita e função.

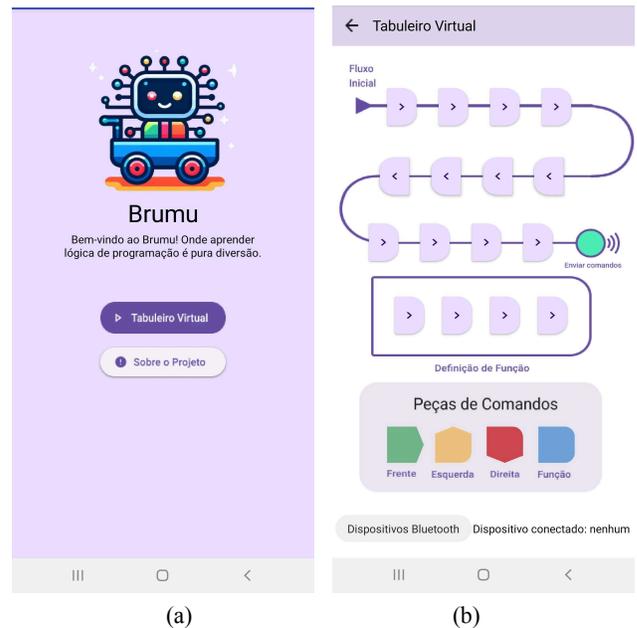


Figura 2: (a) Tela inicial do aplicativo. (b) Tabuleiro Virtual.

Na tela do Tabuleiro Virtual, o usuário pode realizar diversas tarefas, como conectar-se ao carrinho via *bluetooth*, arrastar blocos de codificação (Frente, Esquerda, Direita e Função) para os espaços disponíveis na área de disposição de comandos, conforme a sequência desejada para execução pelo carrinho, e enviar as instruções de movimento para o dispositivo físico. O bloco denominado “Função” permite ao usuário, na área “Definição de Função”, selecionar até quatro blocos de comandos. Quando o bloco de função é incluído no fluxo de comandos, os comandos previamente alocados na área de definição de função são executados de forma indireta, integrando-se automaticamente ao fluxo principal.

A Figura 3 (a) ilustra (por meio das setas vermelhas) como deve ser o percurso percorrido pelo robô, partindo do quadrado com o desenho de um peixe com destino ao quadrado com o desenho de flores, seguindo os comandos selecionados na área de disposição de comandos, mostrado na Figura 3 (b). É importante destacar que pode haver mais de uma seta vermelha dentro de um mesmo quadrado no mapa já que o robô gira para a esquerda ou para a direita em seu próprio eixo. Assim, além de alterar a direção para a qual o usuário deseja que o robô vire, é necessário selecionar o bloco de comando “Frente”, permitindo que alcance o quadrado correspondente à nova direção.

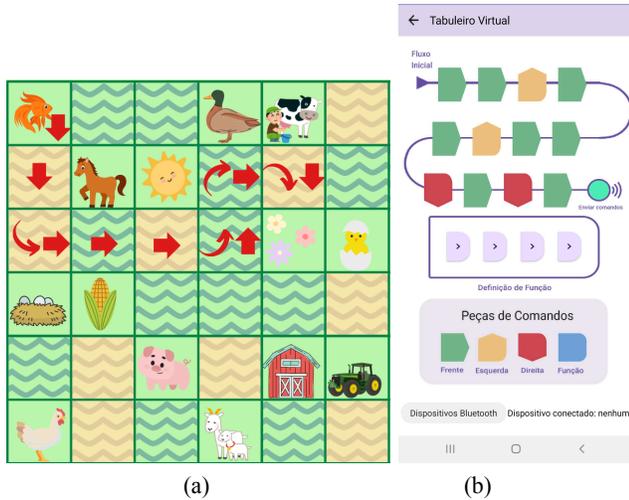


Figura 3: (a) Percurso do robô no mapa. (b) Tabuleiro Virtual com comandos de direção.

Para percursos mais longos, com origem e destino distantes, é necessário utilizar a funcionalidade "Definição de Função". Isso ocorre porque o espaço disponível para comandos suporta apenas doze blocos, o que torna inviável completar o percurso apenas com os blocos Frente, Esquerda e Direita. Ao utilizar funções, a alocação de blocos é otimizada, já que um único bloco de função pode representar até quatro comandos e ainda ser reutilizado em caso de repetição dentro do mesmo fluxo. Como exemplo de reutilização de um mesmo conjunto de peças, a Figura 4 ilustra como o percurso apresentado na Figura 3(a) ficaria utilizando função. Já a Figura 5 (b) ilustra a solução de um percurso mais longo desenhado no mapa da Figura 5 (a).

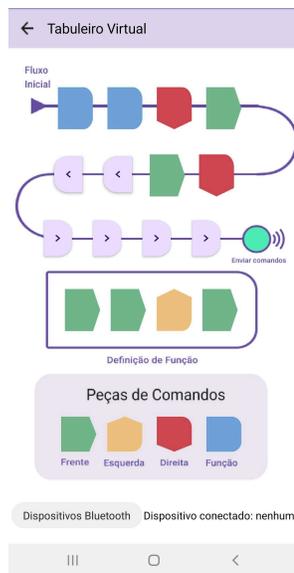


Figura 4: Tabuleiro Virtual com percurso otimizado pelo uso de funções.

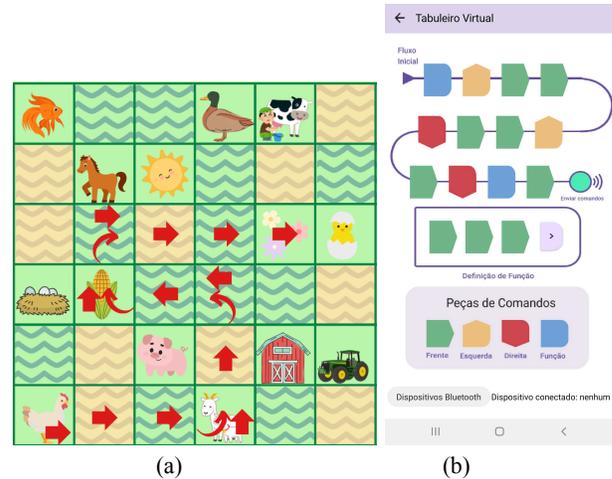


Figura 5: (a) Exemplo de percurso longo no mapa. (b) Tabuleiro Virtual com utilização de função.

Por fim, a Tela de Informações, que é apresentada ao usuário quando o botão "Sobre o Projeto" é acionado pelo usuário na Tela Principal (Figura 2(a)), exibe um texto simplificado de como a aplicação funciona e também disponibiliza um link para o manual completo de montagem, personalização e uso do kit educativo aqui proposto (Figura 6 (a) e (b), respectivamente).

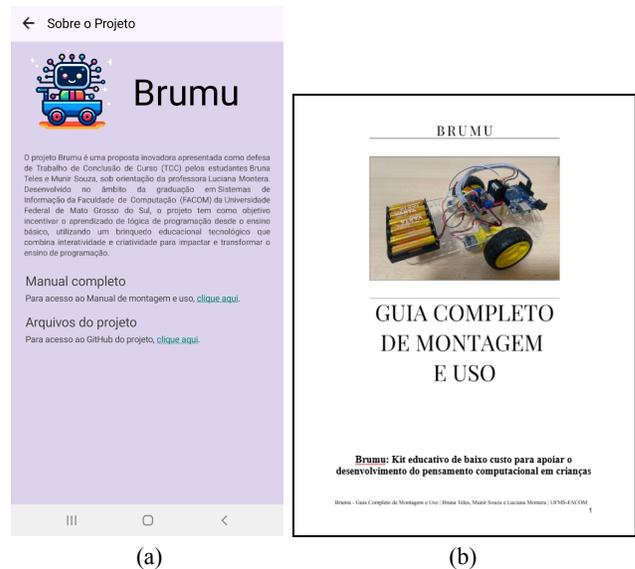


Figura 6: (a) Tela de informações Sobre o Projeto. (b) Print da capa do manual do Brumu.

A Figura 7 apresenta parte da aba de Blocos que representa a lógica e a programação da aplicação. O trecho contém todos os blocos necessários para configurar a conexão da aplicação com dispositivos *bluetooth* e exemplifica a simplicidade do uso do *App Inventor* para criação de aplicativos até com funcionalidades mais complexas.

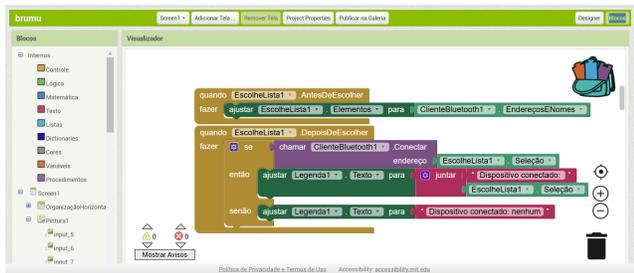


Figura 7: Print de parte do projeto no App Inventor na aba Blocos.

4.2 Carrinho

O desenvolvimento do carrinho em Arduino UNO envolveu tanto a construção física quanto a programação do sistema para a integração dos componentes de *hardware* e *software*. A estrutura física foi baseada em um chassi leve em acrílico, projetado para acomodar os motores e seu controlador, as rodas, o próprio Arduino UNO, a fonte de alimentação, o módulo *bluetooth*, e um botão de ligar/desligar. Essa configuração permitiu uma montagem compacta e funcional, conforme apresenta a Figura 8.

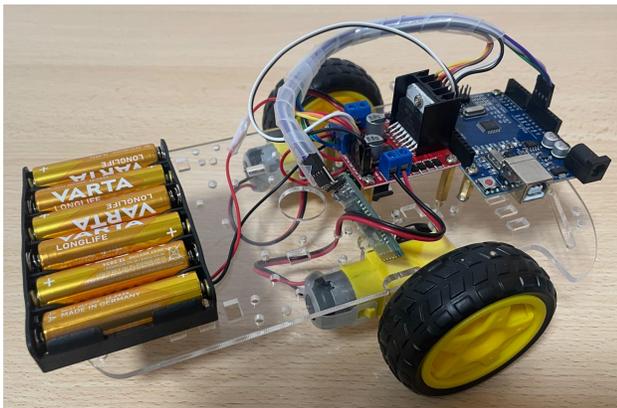


Figura 8: Carrinho construído.

A comunicação remota foi possibilitada pelo módulo *bluetooth* HC-05, conectado ao Arduino, permitindo o envio de comandos via dispositivos móveis para controlar o carrinho.

Os motores das rodas são gerenciados por um *driver* (controlador) Ponte H L298N, que serviu como intermediário entre o Arduino e os motores. O *driver* foi conectado às portas digitais do Arduino, que enviam sinais PWM (*Pulse With Modulation*) para controlar a velocidade e direção. Essa técnica consiste em alternar rapidamente entre os estados ligado e desligado em diferentes proporções, permitindo controlar a velocidade dos motores de forma eficiente e suave, além de determinar sua direção de rotação [11]. Portanto, quanto maior o valor de PWM, mais rápido o motor girará. Quanto menor o valor, mais lento ele girará [12].

A alimentação foi garantida utilizando um suporte com seis pilhas AA conectadas ao *driver* e ao Arduino, juntamente com um botão de ligar/desligar integrado, para segurança e praticidade no controle do estado de alimentação do carrinho.

Para garantir a locomoção precisa do carrinho, tanto nos comandos de avançar quanto nos de girar à direita ou à esquerda (girando em torno de seu próprio eixo), foi necessário fixar a peça chamada "roda boba", que acompanha o kit do chassi (Figura 9). Essa modificação foi essencial para que o carrinho realizasse movimentos mais precisos de direção. Para fixá-la, utilizamos uma fita adesiva, mas outras opções, como supercola, também poderiam ser empregadas.

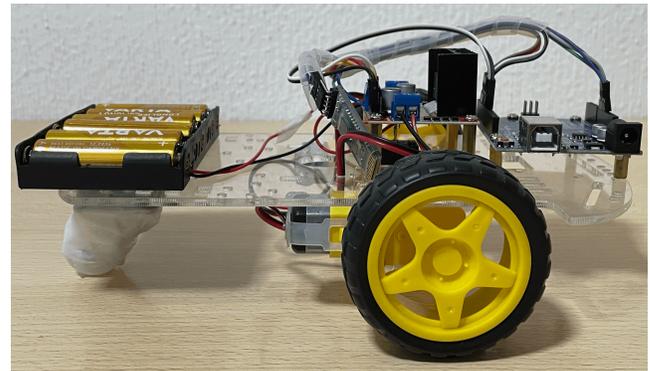


Figura 9: Carrinho.

Por fim, a programação do carrinho foi elaborada em C/C++ usando a IDE Arduino, configurando tanto a comunicação *bluetooth* quanto a lógica de controles dos motores. Na codificação, o Arduino foi programado para receber comandos específicos de direção, como ir para frente, no qual avança 15 cm, ou girar 90 graus em torno do próprio eixo, tanto para a direita quanto para a esquerda. Isso foi implementado calculando a rotação dos motores com base no tempo de acionamento. O código foi projetado para interpretar os comandos recebidos e acionar os motores conforme os parâmetros definidos, garantindo um funcionamento preciso e responsivo.

5 RESULTADOS ALCANÇADOS

Com base nos objetivos deste trabalho foi possível realizar a construção do carrinho desenvolvido em Arduino UNO, juntamente com o aplicativo que envia os comandos de direção para o mesmo. A Tabela 1 apresenta uma descrição detalhada de cada item necessário para a construção do carrinho, incluindo a quantidade requerida, o valor unitário e o custo total de cada componente.

Dada a Tabela 1, podemos concluir que foi possível desenvolver um produto significativamente mais barato em comparação ao Cubetto, considerando que o Brumu apresenta uma redução média de 91,76% quando comparado ao valor que o Cubetto possui atualmente no mercado Brasileiro. Essa diferença destaca o

potencial de criar alternativas mais acessíveis sem comprometer funcionalidades essenciais.

É importante destacar que os valores apresentados foram estimados na data de 17 de novembro de 2024 e não incluem custos de envio, visto que todos produtos foram adquiridos por meio de plataformas online. Esses fatores podem influenciar no custo total e devem ser considerados ao avaliar a viabilidade de sua construção.

Tabela 1: Descrição de valores para construção do carrinho.

Descrição do item	Quantidade	Valor unitário	Valor total
Placa Arduino UNO	1	R\$39,71	R\$39,71
Modulo Bluetooth serial HC-05	1	R\$39,71	R\$39,71
Kit chassi 2WD robô para Arduino	1	R\$48,45	R\$48,45
Módulo Ponte H L298N	1	R\$25,37	R\$25,37
Pilhas AA	6	R\$2,00	R\$12,00
Outros materiais ^a	-	-	R\$15,99
TOTAL	-	-	R\$181,23

^a Parafusos, cabos e fita

Além do robô, o Brumu também inclui um aplicativo *mobile*. Atualmente, para utilizá-lo, é necessário que o usuário faça o download do arquivo .APK, disponibilizado pelos autores, e o instale em um dispositivo *Android*, como celulares ou *tablets*.

Foi também elaborado um material que orienta detalhadamente a construção do carrinho, abrangendo desde a montagem física até a configuração, instalação e uso do aplicativo *mobile* integrado ao mesmo. Todas as instruções estão organizadas em um manual, e os arquivos e códigos-fonte, tanto do carrinho quanto do aplicativo, estão disponíveis publicamente no seguinte repositório do GitHub: <https://github.com/trunabeles/Brumu>.

6 CONSIDERAÇÕES FINAIS

Este trabalho apresenta a construção do Brumu, uma solução de baixo custo composta por um aplicativo *mobile*, um carrinho/robô e seu manual de construção e uso cuja finalidade é colaborar para a democratização da robótica educacional como forma de facilitar o desenvolvimento do Pensamento Computacional em crianças de 3 a 9 anos de idade, uma vez que essa habilidade é crucial para a resolução de problemas de situações cotidianas.

Destaca-se a relevância de soluções de baixo custo, de fácil replicação e simples de serem utilizadas para a popularização da robótica educacional em um número cada vez maior de escolas. Dessa forma, crianças ainda na educação primária podem ter acesso a essas tecnologias, que auxiliam o desenvolvimento do Pensamento Computacional.

O presente trabalho ainda pode continuar em desenvolvimento, explorando a elaboração de mapas para percurso que podem ser aplicados nas diversas áreas da educação como matemática, geografia, química, ciências, história, entre outros. A criação de planos de aula que detalham e orientam como utilizar os componentes do Brumu em sala de aula, consiste também em uma

colaboração futura valiosa capaz de enriquecer ainda mais a experiência de utilização do produto.

Destaca-se, ainda, a relevância de disponibilizar o aplicativo *mobile* nas lojas de aplicativos, facilitando sua instalação e eliminando a necessidade de ajustes manuais nas configurações do dispositivo, o que amplia o alcance e a usabilidade da solução. Além disso, a realização de testes futuros com crianças da educação primária e professores sem experiência prévia em programação permitirá identificar melhorias tanto na interface do aplicativo quanto no desempenho do robô. Essas avaliações contribuirão para tornar o projeto ainda mais acessível, intuitivo e eficaz, proporcionando uma experiência enriquecedora para os usuários finais e consolidando o potencial educacional da ferramenta.

REFERÊNCIAS

- [1] Jeannette M. Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (March 2006), 33–35. DOI: <https://doi.org/10.1145/1118178.1118215>
- [2] Ministério da Educação, 2017. Base Nacional Comum Curricular. MEC, Brasília, DF, Brasil. [Online]. Disponível em: https://www.gov.br/mec/pt-br/escola-em-tempo-integral/BNCC_EI_EF_110518-versaofinal.pdf
- [3] PRIMO. *Cubetto: A toy robot teaching kids code & computer programming*. [Online]. Disponível em: <https://www.primotoys.com>. Acesso em: 3 de novembro de 2024.
- [4] Prabhat, P., Cardoso, F., Sousa, M. e Torres, M. 2020. Plataforma de baixo custo como alternativa para o Cubetto. *Revista Eletrônica de Iniciação Científica em Computação*. 18, 4 (nov. 2020). DOI: <https://doi.org/10.5753/reic.2020.1765>
- [5] MAGRIN, Carlos Eduardo et al. Promovendo a Aprendizagem da Robótica nas Escolas com Metodologias Ativas e o Desenvolvimento de um Robô Móvel Acessível para Redução das Desigualdades Sociais. *Anais do Computer on the Beach*, v.13, p212-219, 2022. DOI: <https://doi.org/10.14210/cotb.v13.p212-219>
- [6] DA LUZ, Marcelo F.; FRANCISCO, Carine S.; FRANCO, Matheus E. Robótica pedagógica aplicada ao ensino-aprendizagem de programação de computadores no ensino médio. *Anais do Computer on the Beach*, v. 10, p. 248-257, 2019.
- [7] David Wolber, Hal Abelson, Ellen Spertus e Liz Looney. 2014. *App Inventor 2: Create Your Own Android Apps*. O'Reilly Media, Inc., Sebastopol, CA, USA. Disponível em: <http://www.appinventor.org/>
- [8] Antonio Finizola, Ewerton Raposo, Maelso Pereira, Wescley Gomes, Ana Araújo e Flávia Souza. 2014. O ensino de programação para dispositivos móveis utilizando o MIT-App Inventor com alunos do ensino médio. *Anais do XX Workshop de Informática na Escola*, novembro 03, 2014, Dourados, Brasil. SBC, Porto Alegre, Brasil, 337-341. DOI: <https://doi.org/10.5753/cbie.wie.2014.337>
- [9] McRoberts. Michael. 2011. *Arduino Básico*. Novatec Editora. São Paulo, SP, Brasil. [Online]. Disponível em: https://edisciplinas.usp.br/pluginfile.php/4287597/mod_resource/content/2/Ard uoC3%ADno%20B%C3%A1sico%20-%20Michael%20McRoberts.pdf
- [10] Arduino. 2024. *Arduino - Home*. Disponível em: <https://www.arduino.cc>. Acesso em: 10 de novembro de 2024.
- [11] Docs Arduino. *Basics of PWM (Pulse Width Modulation)*. [Online]. Disponível em: <https://docs.arduino.cc/learn/microcontrollers/analog-output/>. Acesso em: 10 de novembro de 2024.
- [12] Docs Arduino. *Transistor Motor Control*. [Online]. Disponível em: <https://docs.arduino.cc/learn/electronics/transistor-motor-control/>. Acesso em: 10 de novembro de 2024.