



Serviço Público Federal
Ministério da Educação

Fundação Universidade Federal de Mato Grosso do Sul



Plataforma alternativa para o Sphero Indi

Gabriel Spressola Ziviani

Campo Grande

2024

Agradecimentos

A jornada da graduação foi repleta de desafios, aprendizados e superações, e nada disso seria possível sem o apoio de pessoas fundamentais que estiveram ao meu lado.

À minha mãe, Nilvania, meu agradecimento mais sincero por todo o apoio emocional, pelas palavras de conforto nos momentos difíceis e por acreditar em mim mesmo quando eu duvidava. Sua força me inspirou a continuar.

Ao meu pai, Marcelo, agradeço profundamente pelo apoio incondicional, tanto emocional quanto financeiro, que me permitiu seguir em frente com segurança. Seus conselhos valiosos me ajudaram a tomar decisões importantes ao longo da caminhada.

À Bruna, minha namorada, minha eterna gratidão. Obrigado por estar comigo todos os dias, até o fim. Por suportar minhas crises, por ouvir incansavelmente todas as minhas ideias, angústias e teorias sobre este TCC. Seu amor e paciência foram fundamentais para que eu não desistisse.

E ao meu orientador, muito obrigado por acolher um orientando tão doido como eu. Sua orientação, seus insights e sua escuta aberta deram direção e profundidade a este trabalho. Sem sua orientação generosa, este texto certamente não teria alcançado o que alcançou.

A todos vocês, meu muito obrigado. Este trabalho é também de vocês.

SUMÁRIO

| | |
|--|----|
| Agradecimentos | 1 |
| 1 Introdução | 4 |
| 1.2 Justificativa | 6 |
| 1.3 Objetivos | 7 |
| 2 Fundamentação Teórica | 8 |
| 3 Metodologia | 12 |
| 3.1 Revisão de Tecnologias | 12 |
| 3.2 Escolha de Componentes Econômicos e Pré-Requisitos | 14 |
| 3.3 Aquisição de Componentes Eletrônicos e Montagem | 16 |
| 3.4 Desenvolvimento do Código Arduino | 18 |
| 3.5 Validação do Protótipo | 27 |
| 4 Resultados | 28 |
| 5 Trabalhos Futuros | 31 |
| Referências Bibliográficas | 32 |

Abstract. The educational field has proven to be highly promising for the incorporation of computational technologies due to the wide range of opportunities these tools offer, making the teaching process more engaging and efficient. Among these innovations, Educational Robotics stands out by promoting the development of skills such as conducting research, critical thinking, problem-solving, and the enhancement of logical reasoning. With the aim of promoting robotics education on a national level, this work proposes the creation of a kit for building a more accessible alternative prototype to the Sphero Indi toy, which is a robust tool for educational robotics, designed to teach robotics concepts to children as young as four years old. The developed robot consists of a scaled-down model of a motorized car, with a 3D-printed chassis. Furthermore, it features control based on color identification through a sensor attached to the bottom of the vehicle, using an Arduino UNO R3 controller.

Resumo. O campo educacional tem se mostrado altamente promissor para a incorporação de tecnologias computacionais, devido à ampla variedade de oportunidades que essas ferramentas oferecem, tornando o processo de ensino mais envolvente e eficiente. Dentre essas inovações, a Robótica Educacional se sobressai ao fomentar o desenvolvimento de competências como a realização de pesquisas, o pensamento crítico, a resolução de problemas e o aprimoramento do raciocínio lógico. Com o objetivo de alavancar o ensino de robótica no cenário nacional, este trabalho propõe a criação de um kit para a construção de um protótipo alternativo, mais acessível financeiramente, do brinquedo Sphero Indi, que é uma ferramenta robusta para robótica educacional, projetada com o intuito de ensinar conceitos de robótica a crianças a partir dos quatro anos de idade. O robô desenvolvido consiste em um modelo reduzido de um carro motorizado, com chassi impresso em 3D. Ademais, conta com controle baseado na identificação de cores, por meio de um sensor acoplado na parte inferior do veículo, utilizando um controlador Arduino UNO R3.

1 Introdução

A aplicação de tecnologias computacionais no contexto educacional tem revelado um imenso potencial, proporcionando um amplo espectro de possibilidades que tornam o processo de ensino-aprendizagem mais dinâmico e eficiente. Entre essas inovações tecnológicas, destaca-se a Robótica Educacional, que se configura como uma ferramenta essencial ao estimular o desenvolvimento de uma série de habilidades e competências nos alunos. Essas habilidades estão intrinsecamente ligadas à capacidade de conduzir pesquisas, ao fortalecimento do pensamento crítico e à resolução de problemas de forma lógica e estruturada, promovendo, assim, uma formação mais completa e adequada às demandas do século XXI.

Com a publicação da Lei 14.533, que instituiu a Política Nacional de Educação Digital (PNED), reforça-se a necessidade de integrar essas tecnologias ao ambiente escolar. A Robótica Educacional, ao viabilizar a aprendizagem prática de computação e programação, alinha-se diretamente ao Complemento da Base Nacional Comum Curricular (BNCC), que incorporou a Educação Digital como componente curricular a ser inserido na educação básica. Assim, a robótica não apenas responde às diretrizes da PNED, mas também contribui para uma educação mais inclusiva e alinhada às exigências tecnológicas da sociedade contemporânea.

Nesse contexto, o uso de jogos e brinquedos educacionais que incorporam a tecnologia como base se mostra como uma ferramenta que pode proporcionar maior engajamento e interesse dos estudantes. Robôs didáticos como o *Cubetto*, que permite introduzir conceitos básicos de programação por meio de blocos físicos, ou a *Codipéia*, que ensina lógica e estrutura de comandos de maneira intuitiva, destacam-se como recursos lúdicos que facilitam a compreensão de temas complexos desde a educação infantil. Além disso, robôs como o *Sphero SPRK* e o *Sphero Indi* oferecem oportunidades para uma abordagem mais prática e exploratória, permitindo que os alunos programem trajetórias e interajam com cores

e padrões, desenvolvendo habilidades de resolução de problemas e pensamento computacional.

Sendo assim, esta pesquisa detalha os procedimentos para a construção de um protótipo inspirado no *Sphero Indi*, empregando materiais e componentes de baixo custo e de fácil obtenção, de modo a facilitar sua montagem e implementação em diferentes contextos educacionais, tornando mais barato e acessível.

O *Sphero Indi* é um robô que se movimenta por meio da leitura de cartões coloridos, como exemplificado na figura 1, a seguir, permitindo a combinação de diversos cartões para a execução de trajetos programados. Inicialmente, sua operação não demanda o uso de dispositivos eletrônicos adicionais, como computadores, tablets ou celulares, embora estes possam ser utilizados de forma complementar com o aplicativo *Sphero Edu Jr*.



Figura 1. Cartões coloridos para comandos ao Sphero Indi

Fonte: <https://sphero.com/pages/sphero-indi>, Acesso em: 25 mar. 2025.

O protótipo utiliza uma placa microcontroladora Arduino UNO R3 para gerenciar os componentes, incluindo um sensor de leitura de cores TCS 3472, um par de servomotores e uma bateria de 9V.

Este trabalho se organiza da seguinte forma: esta seção apresenta a introdução, a justificativa e os objetivos, sendo estes separados em gerais e

específicos. Na Seção 2 é apresentada a fundamentação teórica, que explora o contexto científico no qual este se fundamenta. A Seção 3 descreve a metodologia e todo o processo de revisão das tecnologias e componentes utilizados, bem como o desenvolvimento do protótipo e sua validação. Na Seção 4, são apresentados os resultados, incluindo uma tabela de custos.

1.2 Justificativa

O robô educacional *Sphero Indi* é vendido nos Estados Unidos por US\$150, porém no Brasil, dada a realidade atual econômica, a aquisição do robô por famílias de classe média e por escolas com orçamento modesto se torna dificultoso, por conta do preço alto causado por importação e revenda, sendo vendido na faixa de R\$ 3.335,51 (Ri Happy¹) a R\$ 3.820,78 (Carrefour²) a unidade.

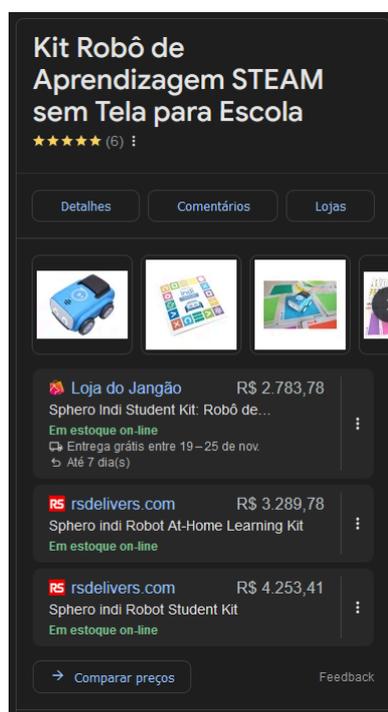


Figura 2: pesquisa de preços relacionadas ao Sphero Indi.

¹
<https://www.rihappy.com.br/kit-robo-de-aprendizagem-steam-sem-tela-para-escola-sphero-azul-1002852156/p?idsku=1002718498&srsIid=AfmBOoo12tETxsEQqhRxSIACsvEAugQT0XNhOQkDTdFEZfNf4ZzR7AoqFLY>

²
<https://www.carrefour.com.br/kit-robo-de-aprendizagem-steam-sem-tela-para-escola-sphero-azul-mp933142256/p>

Assim, como apresentado na figura 2, a aquisição do produto se dificulta, tanto por instituições de ensino, quanto para a maioria das famílias. Além disso, há pouca quantidade de artigos e trabalhos desenvolvidos no Brasil que propõem a produção de robôs que possam ser utilizados para robótica didática, tais como (OLIVEIRA, Prabhát K. et al), que foi inspiração para a metodologia utilizada neste, e (VEIGA, Ernesto Fonseca et al), que sugere um robô para uso genérico.

1.3 Objetivos

Considerando que as tecnologias educacionais, especialmente aquelas relacionadas à robótica, são frequentemente associadas a altos custos, torna-se essencial adaptar tais tecnologias às realidades financeiras do contexto brasileiro, sem comprometer a eficácia no aprendizado.

O objetivo geral deste estudo é desenvolver um protótipo baseado no Sphero Indi que possibilite a aplicação dos mesmos conceitos educativos, mas utilizando componentes que reduzam o custo do hardware sem perder o foco nas competências pedagógicas que a robótica educativa oferece.

Para alcançar esse objetivo, os seguintes objetivos específicos foram definidos:

- Realizar um estudo aprofundado sobre o funcionamento do robô Sphero Indi, incluindo suas interfaces, sensores de cor e outros componentes responsáveis pelo seu movimento e interação.
- Selecionar componentes de baixo custo e alta disponibilidade no mercado nacional. Serão considerados componentes de fácil substituição, visando manter a funcionalidade do protótipo semelhante ao Sphero Indi original.
- Construir um robô utilizando componentes acessíveis e o desenvolvimento de um sistema que permita simular as funcionalidades educativas do Sphero Indi, especialmente no que diz respeito ao aprendizado de lógica e programação.

2 Fundamentação Teórica

Nos últimos anos, o uso de tecnologias educacionais tem ganhado cada vez mais espaço nas escolas, especialmente com a crescente demanda por metodologias que integrem as áreas de Ciência, Tecnologia, Engenharia e Matemática (STEM). Além disso, surge a inclusão do termo "Artes" nessa metodologia, ampliando o conceito para STEAM. A inserção da Arte visa a estimular a criatividade e a expressão individual, proporcionando um equilíbrio entre habilidades técnicas e criativas, essenciais para resolver problemas de maneira inovadora e abrangente.

Essas metodologias buscam preparar os estudantes para os desafios do século XXI, desenvolvendo habilidades como pensamento crítico, resolução de problemas e criatividade, de forma prática e interativa. Nesse contexto, ferramentas de robótica educacional, como o Sphero Indi, têm se destacado como uma abordagem inovadora e acessível para a introdução de conceitos básicos de programação e lógica desde as primeiras etapas da educação. O Sphero Indi, projetado para crianças a partir de 4 anos, utiliza o aprendizado baseado em brincadeiras e a construção de labirintos para estimular a curiosidade e a capacidade de resolver problemas, ao mesmo tempo em que desenvolve habilidades de pensamento computacional. De acordo com a empresa Sphero:

"O Indi inspira o aprendizado imaginativo baseado em brincadeiras, permitindo que as crianças desenhem e construam seus próprios labirintos, enquanto cria oportunidades para os alunos aprenderem os fundamentos da codificação, resolverem problemas e desenvolverem habilidades de pensamento computacional" (SPHERO, 2024).

A utilização de uma abordagem construtiva física, como a aplicada no Sphero Indi, vem sendo abordada desde 1980, vista pela primeira vez em no livro de (PAPERT, Seymour, 1908). O mesmo foi fortemente influenciado por Piaget, com quem trabalhou em Genebra. Ambos compartilham a visão de que as crianças constroem ativamente seu conhecimento, sendo protagonistas no processo de aprendizagem. Piaget propõe a teoria do construtivismo, onde o conhecimento é construído pela

criança à medida que ela interage com o mundo, desenvolvendo esquemas mentais e adaptando-se por meio da assimilação e acomodação.

Papert ampliou essa ideia com o construcionismo, que envolve o aprendizado não apenas como uma construção mental, mas também como uma construção física e prática. Para Papert, a criança não apenas constrói internamente o conhecimento, mas o faz de forma mais eficaz quando está construindo algo tangível, como um robô ou um programa de computador. De acordo com Papert:

"Se realmente olharmos para a 'criança como construtora', estamos a caminho de uma resposta. Todos os construtores precisam de materiais para construir. Onde eu me afasto de Piaget é no papel que atribuo às culturas circundantes como fonte desses materiais. Em alguns casos, a cultura os fornece em abundância, facilitando o aprendizado construtivo piagetiano" .(PAPERT, Seymour.1980, p. 21, Traduzido pelo autor).

Posteriormente, a utilização de tecnologias ligadas à computação na educação tem sido abordada por diversos pesquisadores, e como alguns resultados já demonstram, durante a faixa etária onde há melhora nos campos de atenção, capacidade motora, e capacidade lógica, há maior interesse em robótica didática quando aplicada no ensino básico, como denotado no artigo de (ESQUICHA-TEJADA et al., 2020).:

"[...] a Escola 1 das Meninas e a Escola 2 dos Meninos apresentam um aumento em lógica, memória e atenção de acordo com o grau em que estão estudando (5º e 6º ano do ensino fundamental), isso porque as crianças estão em constante desenvolvimento cerebral ao longo dos anos e a tecnologia promove isso [16]. [...] na Escola Mista, há a mesma tendência, com uma visão geral de que os anos seguintes apresentam percentuais mais altos do que seus predecessores. [...] Finalmente, o total de estudantes que responderam a um conjunto de perguntas relacionadas ao brinquedo interativo mostra que 58% o consideram muito interessante, 38% interessante e apenas 4% indiferente, refletindo o interesse que o jogo desperta."(ESQUICHA-TEJADA et al., 2020, p. 6-7, traduzido pelo autor).

Também há, em pesquisas, a preocupação quanto a adaptabilidade e compatibilidade de plataformas de robótica didática, e suas capacidade didática como demonstrado no estudo de (BAKALA, Ewelina et al., 2022):

"Analisamos três ferramentas em conjunto: KIBO, KIWI e CHERP. KIBO é um robô atualmente disponível no mercado, anteriormente conhecido como 'KIWI' ou Kids Invent with Imagination (Tufts University, 2022). CHERP é uma linguagem de programação usada para programar o KIBO e o KIWI, então avaliar o CHERP é equivalente a avaliar o KIBO e o KIWI. Em alguns casos, a interface de programação continha imagens, o que a tornava acessível para crianças pré-alfabetizadas, mas tivemos a impressão de que os sistemas foram projetados para crianças mais velhas que a nossa faixa etária alvo." (BAKALA, Ewelina et al., 2022, p. 6, traduzido pelo autor).

Desta maneira, fica nítido que ao desenvolver tais ferramentas didáticas é necessário atenção extra para os quesitos de facilidade de utilização e entendimento, com foco na faixa etária alvo. No que se refere à adaptabilidade, o trabalho de (D'ALESSANDRO, Giulia. 2022) foi realizada uma comparação entre vários modelos de robôs didáticos, gerando o comparativo mostrado na Figura 3 a seguir.

São considerados os seguintes aspectos: facilidade de utilização de forma individual e em grupo, capacidade de modificar a programação em caso de erro, facilidade e custo de produção, precisão na decodificação dos comandos, suporte a diferentes tipos de comandos, e por final, espaço físico necessário para a programação. Todos os quesitos classificados de um a cinco de maneira pictográfica.

| | Exemplo | Facilidade de Utilização Sozinho | Facilidade De Utilização Em Grupo | Facilidade de Modificação Da Programação Em Caso De Erro | Facilidade de Produção | Custo de Produção | Precisão da Decodificação | Suporte de Comandos | Espaço Utilizado Pelo Código |
|---------------|--|----------------------------------|-----------------------------------|--|------------------------|-------------------|---------------------------|---------------------|------------------------------|
| Cubetto |  | ●●●●● | ●●●●○ | ●●●●● | ●●●●○ | ●●●●○ | ●●●●● | ●●●●○ | ●●○○○ |
| mTiny |  | ●●●●○ | ●●●●● | ●●○○○ | ●●●●● | ●○○○○ | ●●○○○ | ●●●●● | ●●●●○ |
| Kibo |  | ●●●●○ | ●●●●● | ●●○○○ | ●●●●○ | ●●●●○ | ●●○○○ | ●●●●● | ●●●●○ |
| Bee Bot |  | ●●●●● | ●○○○○ | ●●●●● | ●●●●○ | ●●○○○ | ●●●●● | ●○○○○ | ●○○○○ |
| Code a Pillar |  | ●●●●○ | ●●●●○ | ●●●●○ | ●○○○○ | ●●●●● | ●●●●● | ●●●●○ | ●●●●○ |
| Sphero Indi |  | ●●●●● | ●●●●● | ●●●●● | ●●●●● | ●○○○○ | ●●●●● | ●●●●● | ●●●●○ |
| MatataBot |  | ●●●●● | ●●●●○ | ●●●●● | ●●●●○ | ●●●●○ | ●●●○○ | ●●●●○ | ●●○○○ |

Figura 3. Tabela de comparação avaliativa entre robôs didáticos

Fonte: Traduzido de (D’ALESSANDRO, Giulia. 2022), p.1. Adaptado pelo autor.

Pode-se observar na figura 3 que o Sphero Indi, que aparece na penúltima linha da tabela, possui um desempenho consistente em todos os aspectos avaliados. Ele é destacado como um dos robôs com maior facilidade de utilização tanto individualmente quanto em grupo, além de permitir modificações rápidas na programação em caso de erro, o que é fundamental em um ambiente educacional. Outro ponto relevante é sua precisão e capacidade de suportar uma ampla variedade de comandos, o que o torna extremamente versátil. Essa combinação de características, aliada à facilidade de produção e ao baixo custo associado, torna o Sphero Indi perfeito para ser a base desta plataforma educacional.

3 Metodologia

Este trabalho segue, de maneira adaptada, a metodologia aplicada no artigo de (OLIVEIRA, Prabhát K. et al). A metodologia foi organizada em cinco fases distintas, cada uma com um papel essencial para o progresso eficiente do projeto. Na Seção 3.1, é realizada a *revisão das tecnologias*, onde se examinam as peças e sensores existentes, identificando suas características, vantagens e limitações. Na Seção 3.2, ocorre a *escolha de componentes econômicos*, na qual são selecionados materiais e tecnologias acessíveis, de baixo custo e com potencial de atender às necessidades do projeto. Na Seção 3.3, é detalhada a *aquisição de componentes e montagem*, onde os materiais escolhidos são adquiridos e o protótipo é montado de acordo com o projeto técnico estabelecido. Na Seção 3.4, é descrito o *desenvolvimento do código para o arduino* parte por parte. Por fim, na Seção 3.5, ocorre a *validação do protótipo*, etapa em que o sistema é testado em um ambiente controlado para verificar sua eficiência, precisão e facilidade de uso.

3.1 Revisão de Tecnologias

A primeira etapa envolveu a análise aprofundada das tecnologias empregadas no desenvolvimento do projeto Sphero Indi, com o intuito de identificar como essas soluções poderiam ser adaptadas para o novo contexto, com a finalidade de assegurar simplicidade e acessibilidade para o público-alvo.

Dado o alto custo do brinquedo, foi empregada uma análise do funcionamento via vídeos e descrições no site do fabricante para o melhor entendimento possível do funcionamento e capacidades do robô a ser replicado.

Para a produção de um protótipo inspirado na funcionalidade do sphero indi é necessário que o robô possua a funcionalidade de deslocamento por motores, leitura e interpretação de cartões coloridos. Para tais funcionalidades fora suposto necessário um sensor capaz de diferenciar valores dentro de um espectro de cores,

uma placa controladora e apenas um par de motores de passo e seus controladores, ou “drivers”.

Dentre as placas controladoras disponíveis, destacam-se as placas da linha Arduino, que possui várias placas de extrema acessibilidade, tais como o Arduino Nano, que é uma alternativa compacta e eficiente, ideal para projetos onde o espaço é limitado, Arduino Leonardo que é conhecido por sua capacidade de comunicação USB, podendo ser reconhecido como um dispositivo de entrada (como teclado ou mouse) diretamente pelo computador, e por fim, a placa Arduino Uno R3 se destaca por sua versatilidade, robustez e grande quantidade de documentação e suporte na comunidade. É capaz de controlar motores de passo e de lidar com sensores de cor com facilidade, graças ao seu número de pinos de entrada e saída digitais e analógicos.

Existem outras placas como o Arduino Mega, que oferece uma maior quantidade de pinos e maior memória, sendo interessante para projetos mais complexos com muitos periféricos, ou ESP32 que possuem recursos avançados como conectividade Wi-Fi e Bluetooth, permitindo possibilidades adicionais de comunicação. No entanto, para um robô inspirado no Sphero Indi, com funcionalidades específicas e controláveis por uma placa de médio porte, o Arduino Uno R3 atende perfeitamente às necessidades sem o custo adicional de uma placa de maior potência.

Assim, o Arduino Uno R3 é a melhor escolha para este protótipo devido à sua simplicidade, custo acessível e capacidade de controlar motores e comunicar-se com sensores, e por fim oferece uma excelente base para desenvolvimento educacional por conta de sua documentação e suporte amplamente acessíveis.

Além disso, a disponibilidade de sensores RGB³ compatíveis com a plataforma Arduino é limitada, sendo que os modelos mais utilizados incluem o TCS3472, TCS230 e TCS3200. Esses sensores foram desenvolvidos pela TAOS (Texas Advanced Optoelectronic Solutions). No entanto, até o momento de escrita, o sensor TCS3472 destaca-se como a melhor opção entre eles por uma série de fatores técnicos e econômicos que atendem aos requisitos de projetos acadêmicos e

³ (diminutivo para “*Red, Green, Blue*”, ou “*Vermelho, Verde, Azul*”, sendo as estas as 3 faixas de cores captadas pelo sensor)

educacionais, oferecendo um excelente custo-benefício, sendo frequentemente encontrado no mercado a preços acessíveis quando comparado a outros sensores RGB.

Já quanto ao movimento, motores simples com caixa de redução poderiam ser utilizados, pois oferecem torque suficiente para a movimentação, mas apresentam baixa precisão, o que comprometeria a estabilidade e o controle exato do trajeto. Por outro lado, motores de passo, como os empregados no estudo de (OLIVEIRA, Prabhát K. et al), garantiriam um controle preciso da posição, mas a baixa velocidade inerente ao seu funcionamento os tornaria inadequados para um sistema que exige deslocamentos dinâmicos e eficientes.

Diante dessas limitações, a melhor alternativa consiste em um meio-termo entre as duas opções mencionadas. Assim, optou-se por uma adaptação dos motores Tower Pro, comumente usados como servomotores com mobilidade de 180 graus em projetos de arduino, possibilitando quando modificado, um movimento contínuo de 360 graus do eixo do motor mantendo a precisão necessária.

3.2 Escolha de Componentes Econômicos e Pré-Requisitos

Com base na revisão inicial, a fase seguinte centrou-se na busca e seleção de materiais de baixo custo, capazes de compor o protótipo sem comprometer a funcionalidade ou a qualidade do resultado final. Por mais que a viabilidade financeira foi uma prioridade nesta etapa, também foi levada em conta o formato de viabilidade do produto final.

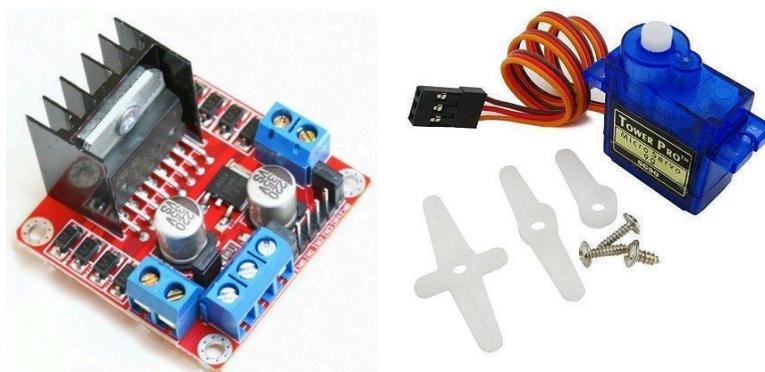


Figura 4. Ponte H e Motor Tower Pro 9g

Fonte: Eletrogate.com

Dentre as possibilidades, as de melhor acesso e praticidade de aquisição e montagem são o Arduino Uno R3 como placa controladora, sensor TCS 3472 para reconhecimento de cores, e Tower Pro 9g modificados para funcionamento 360 graus, auxiliados por um controlador de voltagem, também conhecido como ponte H, ilustrados na figura 4.

Para o chassi, ou corpo do brinquedo, analisando as alternativas de compra se torna mais proveitoso a impressão de um modelo 3D, reduzindo ainda mais os custos do projeto. Para isso, um chassi foi desenvolvido com o software 3D Builder, da Microsoft, com auxílio de modelos 3D das peças internas também modeladas pelo autor, criando um esquema 3D do projeto eletrônico em escala, como demonstrado nas figuras 5 e 6 a seguir.

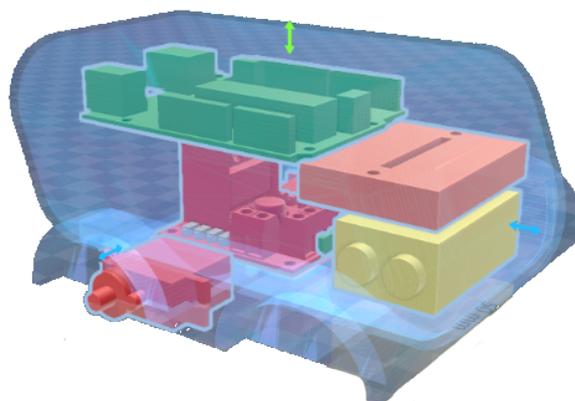


Figura 5. Esquema 3d dos componentes internos do protótipo.

Fonte: Elaboração própria.

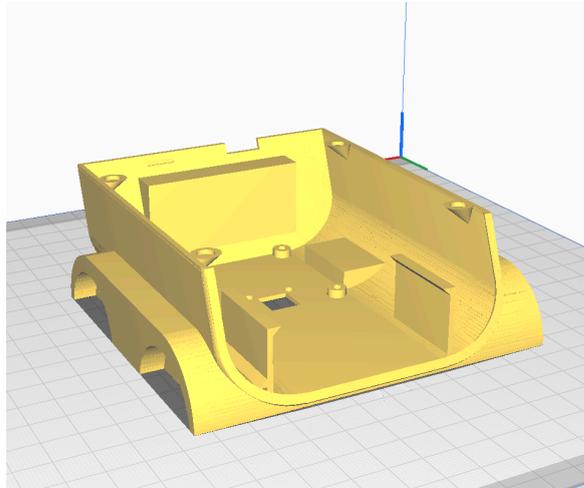


Figura 6. Modelo do chassi finalizado.

Fonte: Elaboração própria.

3.3 Aquisição de Componentes Eletrônicos e Montagem

A terceira etapa consistiu na compra de componentes eletrônicos, como o controlador Arduino, motores, e sensor e também a montagem do protótipo do veículo. Assim, levando em conta as necessidades citadas anteriormente, todos componentes utilizados neste trabalho podem ser facilmente adquiridos por sites como MercadoLivre, Shopee, ou outro site de compras de fácil acesso.

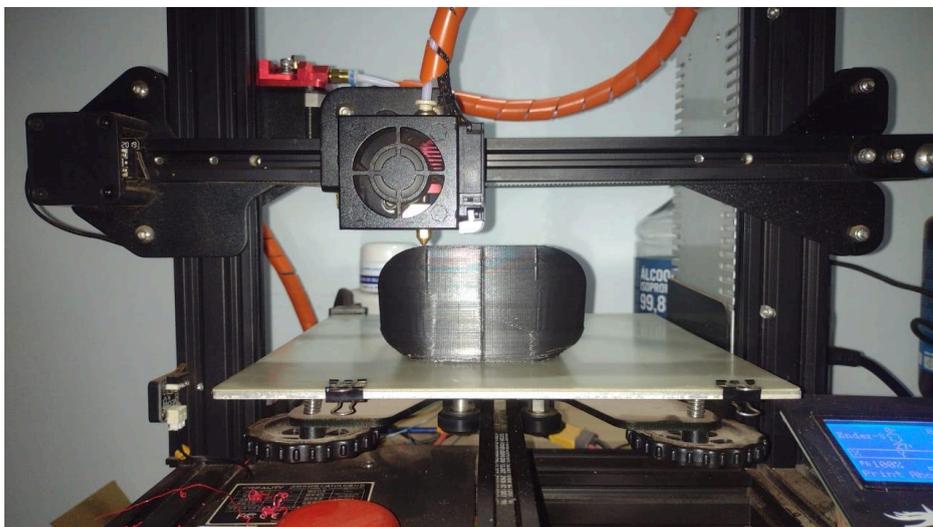


Figura 7. Impressão 3D do Chassi.

Fonte: Elaboração própria.

Também foi realizada a impressão do chassi, como exibido na figura 7, usando *PLA* como material e uma impressora 3D modelo *Ender 3*. O arquivo de impressão foi gerado pelo programa *Ultimaker Cura* e utilizado as configurações padrão oferecidas.

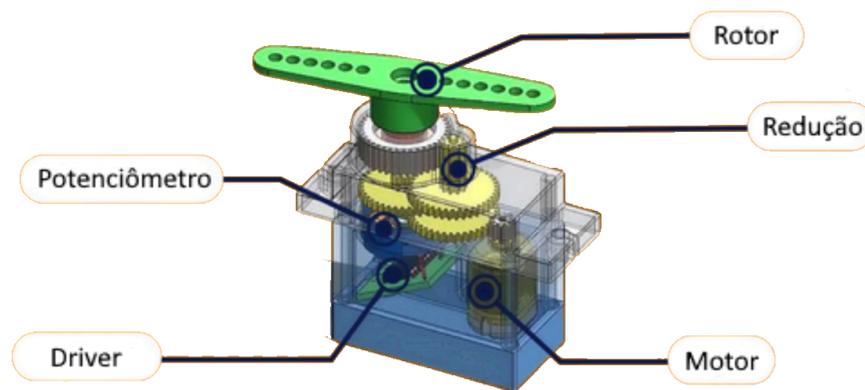


Figura 8. Esquema 3d dos componentes do motor.

Fonte: Elaboração própria.

Além disso foi realizada nesta etapa a modificação dos motores Tower Pro, removendo a faixa de resistência do potenciômetro que limita o motor, e também soldando os fios diretamente nos terminais do motor, levando em conta o diagrama na figura 8, e como demonstrado a seguir, na figura 9.

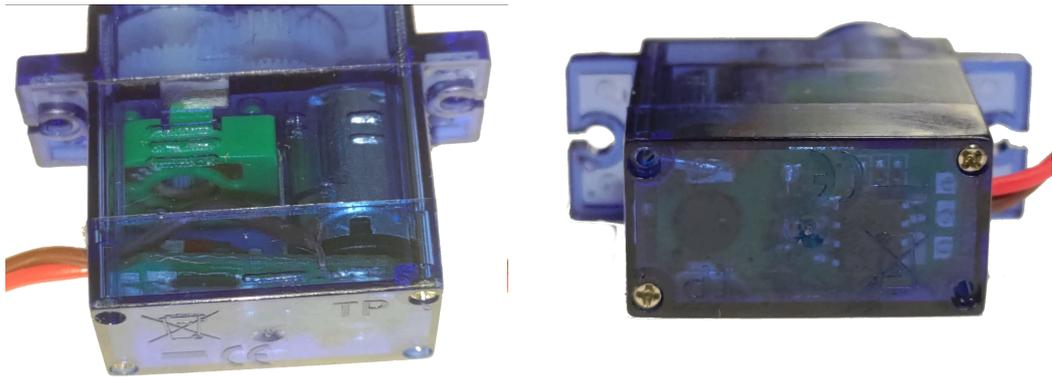


Figura 9. Motor após a modificação.

Fonte: Elaboração própria.

3.4 Desenvolvimento do Código Arduino

Esta fase envolve a criação de um código em C/Arduino para o funcionamento do robô, junto de vários cartões coloridos, e assim classificar os valores no firmware produzido para o protótipo.

Inicialmente foi explorado a capacidade do sensor de cores e os códigos de exemplo fornecidos pela Adafruit, onde então foi adaptado um programa básico para analisar o feedback fornecido pelo sensor de cores.

Logo após, foi feito um código para controle dos motores usando os mesmos conceitos e disponibilizado abertamente no site github com o nome do projeto, batizado de Indi-Duino, disponível em: github.com/Slayer-Nefarox/Indi-Duino.

O código desenvolvido é estruturado no padrão Arduino, incluindo as bibliotecas, que são códigos extras já montados por padrão, e necessários, com definições globais onde são definidos os pinos a serem utilizados, a instância do sensor (Adafruit_TCS34725) onde ele se torna acessível pelo arduino, campos Setup onde é declarado a intenção de entrada ou saída de informação nos pinos, e por fim Loop, onde é escrita a lógica de repetição do robô.

```
#include <Wire.h>
#include "Adafruit_TCS34725.h"

#define Mot_L_Frente 11
#define Mot_L_Tras 10
#define Mot_R_Frente 6
#define Mot_R_Tras 5

#define Tempo 1500 // tempo (em milissegundos) entre comandos

#define intervalo 15 // intervalo de tolerancia da faixa de cor

#define LedEnable 2 //pino que liga/desliga o led (caso use iluminação propria da placa)

Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_50MS,
TCS34725_GAIN_60X); // ganho possivel: X01, X04, X16, X60.
//variaveis globais
//Velocidades individuais para cada motor para balancear imparidade de hardware. Minimo
1, máximo até 63.
int speed_L = 59;
int speed_R = 63;

int speed_mult = 0;
int ultima_Cor = 0;

//indice de cores
int verde[3] = {60,110,80};
int vermelho[3] = {150, 55,55};
int azul[3] = {28,86,131};
int roxo[3] = {103,51,94};
// ...
```

Então, inicialmente são criadas duas funções: uma que escreve para o computador o que o sensor está vendo chamada “Debug”, em RGB, para caso de cadastro e monitoramento de cores, e outra que verifica se um valor de cor está dentro de uma faixa de tolerância definida, chamada “Faixa”. A faixa de tolerância é determinada pelo parâmetro `intervalo`, que ajusta a sensibilidade da comparação entre as cores lidas pelo sensor e as cores pré-definidas no código.

```
void debug(float red, float green, float blue)
{
  //debug valor do RGB
  Serial.print("R:\t"); Serial.print(int(red));
  Serial.print("\tG:\t"); Serial.print(int(green));
  Serial.print("\tB:\t"); Serial.print(int(blue));
  Serial.print("\n");
}
```

```
bool faixa(int var, int valor)
{
  if( var > valor-intervalo && var <= valor+intervalo )// se 'var' estiver dentro do intervalo
  levando em conta a margem de tolerância
  {
    return true;
  }
  else
  {
    return false;
  }
}
```

No início do programa, a função `setup` inicializa os pinos dos motores e do LED de modo a definir se haverá entrada (input) ou saída (output) de dados, além de verificar a presença do sensor de cor. Caso o sensor não seja detectado, o código entra em um modo de recuperação, indicando erro na comunicação com o sensor e pedindo a checagem das conexões e reinicialização. Se tudo ocorrer de maneira certa, a comunicação é iniciada para permitir a leitura de cores.

```
void setup()
{
  // Inicia a comunicação serial a 9600 bits por segundo.
```

```

// Isso serve para enviar mensagens para o computador e depurar o funcionamento.
Serial.begin(9600);

// Configura os pinos dos motores como saída, ou seja, eles irão enviar sinais elétricos.
pinMode(Mot_L_Frente, OUTPUT);
pinMode(Mot_L_Tras, OUTPUT);
pinMode(Mot_R_Frente, OUTPUT);
pinMode(Mot_R_Tras, OUTPUT);

// Garante que os motores comecem desligados.
// O valor 0 no analogWrite significa "sem potência" (PWM desligado).
analogWrite(Mot_L_Frente, 0);
analogWrite(Mot_L_Tras, 0);
analogWrite(Mot_R_Frente, 0);
analogWrite(Mot_R_Tras, 0);

// Inicializa o sensor de cores TCS34725.
// Se ele for encontrado, imprime uma mensagem no Serial.
if (tcs.begin())
{
  Serial.println("Found sensor");
}
else
{
  // Caso o sensor não seja detectado, o programa entra em um loop infinito,
  // imprimindo uma mensagem de erro continuamente.
  // Isso evita que o programa continue rodando sem o sensor.
  while (1)
  {
    Serial.println("No TCS34725 found ... check your connections and reboot");
  }
}

// Configura o pino do LED como saída, para controlar o LED de iluminação do sensor.
pinMode(LedEnable, OUTPUT);
}

```

A função `loop` é executada continuamente e realiza a detecção das cores no ambiente, interpretando os valores RGB capturados pelo sensor e realizando as ações correspondentes com as cores a serem detectadas.

A cadeia de ações foi estruturada com base em uma lógica de “se isso, faça aquilo” “senão, faça outro”, criando uma árvore de possibilidades associadas aos seguintes comportamentos:

- Cor verde: Acelera o robô, ajustando a velocidade dos motores.
- Cor azul: O robô vira à direita.
- Cor vermelha: O robô desacelera ou para.
- Cor roxa: O robô vira à esquerda.

Essas ações são realizadas por meio do comando `analogWrite()`, que ajusta a velocidade dos motores, permitindo controlar a intensidade da movimentação do robô.

```
void loop()
{
  // Ativa o LED integrado do sensor TCS34725, garantindo iluminação adequada
  // para a leitura precisa das cores no ambiente.
  digitalWrite(LedEnable, HIGH);

  // Define o tempo de atraso base para manobras (virar, acelerar, etc.).
  // Esse tempo pode ser recalculado caso a velocidade do robô seja alterada.
  float tempo_delay = Tempo;

  // Variáveis para armazenar as leituras de cor (valores RGB).
  float red, green, blue;

  // Captura os valores de vermelho, verde e azul detectados pelo sensor.
  tcs.getRGB(&red, &green, &blue);

  // -----
  // 1) AÇÃO PARA COR VERDE → "Seguir em frente / Acelerar"
  // -----
  if(faixa( red, verde[0] ) && faixa(green, verde[1]) && faixa( blue, verde[2] ) )
  {
    // Garante que o comando só seja executado quando a cor detectada for diferente da
    última.
    if(ultima_Cor != 1)
```

```

{
  Serial.print("Frente ou acelera."); // Mensagem para fins de depuração.

  // Aumenta gradualmente a velocidade do robô (até um limite de multiplicador).
  if(speed_mult<4)
  {
    speed_mult +=2;
    // Recalcula o tempo de curva proporcional à velocidade.
    tempo_delay = Tempo / ((2/3)*speed_mult);
  }

  ultima_Cor = 1; // Atualiza memória da última cor lida.
  delay(500); // Pausa para estabilizar a leitura do sensor.

  // Aciona ambos os motores no sentido de frente.
  analogWrite(Mot_L_Frente, speed_mult*speed_L);
  analogWrite(Mot_L_Tras, 0);
  analogWrite(Mot_R_Frente, speed_mult*speed_R);
  analogWrite(Mot_R_Tras, 0);
}
else
{
  // Caso a cor já tenha sido executada, evita repetir a ação.
  Serial.print("Frente ou acelera, Negado.");
}
}

// -----
// 2) AÇÃO PARA COR AZUL → "Virar à direita"
// -----
else if(faixa( red, azul[0] ) && faixa( green, azul[1] ) && faixa( blue, azul[2] ) )
{
  if(ultima_Cor != 2)
  {
    Serial.print("Virar a direita.");
    ultima_Cor = 2;
    delay(500);

    // Liga apenas o motor esquerdo para girar o robô à direita.
    analogWrite(Mot_L_Frente, speed_mult*speed_L);

```

```

analogWrite(Mot_L_Tras, 0);
analogWrite(Mot_R_Frente, 0);
analogWrite(Mot_R_Tras, 0);

// Mantém a manobra por um tempo proporcional à velocidade.
delay(tempo_delay);

// Retoma o movimento para frente com ambos os motores.
analogWrite(Mot_L_Frente, speed_mult*speed_L);
analogWrite(Mot_L_Tras, 0);
analogWrite(Mot_R_Frente, speed_mult*speed_R);
analogWrite(Mot_R_Tras, 0);
}
else
{
  Serial.print("Virar a direita, Negado.");
}
}

// -----
// 3) AÇÃO PARA COR VERMELHO → "Desacelerar ou Parar"
// -----
else if(faixa( red, vermelho[0] ) && faixa(green, vermelho[1]) && faixa( blue, vermelho[2] ) )
{
  if(ultima_Cor != 3)
  {
    Serial.print("Desacelera ou para.");

    // Reduz gradualmente a velocidade do robô, até parar totalmente.
    if(speed_mult>0)
    {
      speed_mult -=2;
      tempo_delay = Tempo / ((2/3)*speed_mult);
    }

    ultima_Cor = 3;
    delay(500);

    // Aplica a nova velocidade nos dois motores.
    analogWrite(Mot_L_Frente, speed_mult*speed_L);

```

```

    analogWrite(Mot_L_Tras, 0);
    analogWrite(Mot_R_Frente, speed_mult*speed_R);
    analogWrite(Mot_R_Tras, 0);
}
else
{
    Serial.print("Desacelera ou para, Negado.");
}

}

// -----
// 4) AÇÃO PARA COR ROXO → "Virar à esquerda"
// -----
else if(faixa( red, roxo[0] ) && faixa(green, roxo[1]) && faixa( blue, roxo[2] ) )
{
    if(ultima_Cor != 4)
    {
        Serial.print("Virar a esquerda.");
        delay(500);
        ultima_Cor = 4;

        // Liga apenas o motor direito para girar o robô à esquerda.
        analogWrite(Mot_L_Frente, 0);
        analogWrite(Mot_L_Tras, 0);
        analogWrite(Mot_R_Frente, speed_mult*speed_R);
        analogWrite(Mot_R_Tras, 0);

        delay(tempo_delay);

        // Retoma o movimento para frente.
        analogWrite(Mot_L_Frente, speed_mult*speed_L);
        analogWrite(Mot_L_Tras, 0);
        analogWrite(Mot_R_Frente, speed_mult*speed_R);
        analogWrite(Mot_R_Tras, 0);
    }
}
else
{
    Serial.print("Virar a esquerda, Negado.");
}
}

```

```

// -----
// CASO NENHUMA COR SEJA DETECTADA ENTRE AS CADASTRADAS
// -----
else
{
  Serial.print("Nenhuma cor detectada entre as cadastradas.");
  ultima_Cor = 0; // Reseta memória de cor para permitir futuras leituras.
}

// -----
// MODELO DE COMO ADICIONAR NOVAS CORES
// Este bloco comentado serve como referência para expandir
// o sistema com novos comportamentos associados a cores extras.
// -----
/*
else if(faixa( red, cor[0] ) && faixa(green, cor[1]) && faixa( blue, cor[2] ) // cor
{
  Serial.print("frente");
  analogWrite(Mot_L_Frente, 200);
  analogWrite(Mot_L_Tras, 0);
  analogWrite(Mot_R_Frente, 200);
  analogWrite(Mot_R_Tras, 0);
  delay(tempo_delay);
}
*/

// Função de depuração: imprime no Serial os valores RGB detectados,
// permitindo o ajuste fino da calibração de cores.
debug(red, green, blue);
}

```

3.5 Validação do Protótipo

Na última fase, o protótipo foi submetido a uma bateria de testes, verificando a integridade das conexões e a precisão na execução dos comandos de controle do veículo.

Durante os testes, observou-se que o robô demonstrou um alto grau de precisão ao interpretar os comandos visuais fornecidos por meio dos cartões coloridos. No entanto, verificou-se também uma relação direta entre o nível de carga da bateria e a exatidão do movimento do motor ao executar rotações angulares específicas. Esse fenômeno ocorre devido à variação da tensão disponível, que impacta a potência entregue ao motor, influenciando a fidelidade do deslocamento planejado. Como o cálculo do movimento rotacional é baseado em um tempo de atraso (delay) para executar giros à esquerda ou à direita, qualquer alteração na energia fornecida pode resultar em desvios da trajetória esperada, comprometendo a previsibilidade do sistema.



Figura 10. Testes de funcionamento realizados na etapa de validação
(fonte: geração própria)

Além disso, a etapa de reconhecimento de cores mostrou-se robusta e confiável, garantindo um mapeamento preciso das tonalidades utilizadas nos comandos. Para aumentar a tolerância a variações na iluminação ambiente, foi

adotado um modelo de detecção baseado em faixas de cor, permitindo uma margem de variação sem comprometer a interpretação correta dos comandos. Esse ajuste se revelou essencial para minimizar erros decorrentes de mudanças na intensidade da luz ou pequenas diferenças na pigmentação dos materiais empregados nos cartões.

Os experimentos foram conduzidos utilizando folhas de papel sulfite coloridas, previamente calibradas para otimizar a identificação e assimilação das funcionalidades programadas. A escolha desse material também se mostrou vantajosa devido à sua ampla disponibilidade e custo acessível, tornando-o uma alternativa viável para aplicações educacionais e experimentais.

4 Resultados

Como resultado da pesquisa e do desenvolvimento realizados, comprovou-se a viabilidade da replicação do robô educacional Sphero Indi utilizando uma placa controladora Arduino. Os testes demonstraram que o protótipo desenvolvido apresentou um funcionamento adequado, confirmando sua capacidade de desempenhar as funções essenciais do modelo original. Além disso, os experimentos indicaram que há margem para aprimoramentos, permitindo a adição de novas funcionalidades que podem tornar o dispositivo ainda mais versátil e completo para aplicações educacionais, evidenciando que a abordagem adotada abre caminho para inovações que ampliam sua utilidade no contexto do ensino de robótica.

Ambos os sistemas compartilham o mesmo princípio de funcionamento: a leitura de cores como meio de controle e navegação do robô. Assim como o Sphero Indi, o protótipo é capaz de interpretar comandos por meio de cartões coloridos posicionados no trajeto, o que o torna igualmente aplicável ao ensino de lógica e pensamento computacional para crianças.

Em termos de desempenho, o protótipo apresentou estabilidade na interpretação das cores e confiabilidade nas respostas, com um tempo de processamento adequado às dinâmicas de sala de aula. Embora o Sphero Indi possua um sistema fechado e otimizado, com sensores embutidos e resposta tão precisa quanto, os testes realizados com o protótipo indicam que ele é suficientemente robusto para uso educacional, cumprindo as exigências de um ambiente escolar.

Um dos principais diferenciais do protótipo em relação ao Sphero Indi está na sua flexibilidade e potencial de edição. Por ser baseado na plataforma aberta do Arduino, ele permite a adição de novas funcionalidades, e até cadastro de novas cores.

Do ponto de vista financeiro, o protótipo se destaca entre soluções de baixo custo, sendo que além de ser construído a partir de componentes acessíveis, cuja seleção foi feita com base em uma pesquisa de preços, tem fabricação fácil e barata do chassi por meio de impressão 3D.

Além disso, por conta da disponibilização do projeto de maneira pública e/ou *open source*⁴ faz com que a manutenção e reprodução do projeto seja possível abertamente sem necessidade de tarifas e feito por qualquer um com conhecimento mínimo de eletrônica, programação, e uso de impressoras 3D.

Os resultados obtidos foram satisfatórios dentro dos critérios estabelecidos para esta pesquisa, garantindo um nível de desempenho e precisão compatível com as exigências do ambiente escolar. O protótipo demonstrou estabilidade na interpretação das cores, confiabilidade na resposta aos comandos e um tempo de processamento adequado às dinâmicas de sala de aula.

⁴ Open source (ou código aberto) é um termo que se refere a softwares cujo código-fonte é disponibilizado publicamente

Para a construção do protótipo, foi essencial realizar uma pesquisa detalhada de preços e a aquisição dos componentes necessários, gerando assim uma tabela. Para efeito de comparação e melhor compreensão dos custos envolvidos, os valores de cada componente foram convertidos para a cotação do dólar americano vigente no momento da escrita, utilizando a taxa de câmbio de R\$5,7081 para US\$1, e colocado lado a lado.

| NOME DA PEÇA | QUANTIDADE | PREÇO UNITÁRIO | PREÇO TOTAL Real - Dólar |
|--|-------------------|----------------------------|-------------------------------------|
| Arduino Uno R3 | 1 | R\$42,38 | R\$42,38 -\$7,42 |
| Ponte H | 1 | R\$15,20 | R\$15,20 -\$2,66 |
| Micro Servo Motor Tower Pro 9g Sg90 para Arduino | 2 | R\$18,81 | R\$37,62 -\$6,59 |
| Sensor TCS | 1 | R\$35,90 | R\$35,90 -\$6,29 |
| Kit Cabos Jumper 20cm Fxf + Mxm + Mxf 120 Unidades Arduino | 1 | R\$49,9 | R\$49,9 - \$8,75 |
| Chassi e rodas impressos em 3d | 1 | 8,28 + 3,19 | R\$11,47 - \$2,01 |
| Total | 7 | R\$192,56 - \$33,72 | |

5 Trabalhos Futuros

Conforme discutido no tópico anterior, a metodologia adotada neste trabalho não apenas viabilizou a replicação do robô educacional, mas também abriu um leque de possibilidades para futuras otimizações, tanto no âmbito do software quanto na infraestrutura e no design do protótipo. A flexibilidade do sistema permite a incorporação de novos recursos, como sensores adicionais para ampliar a capacidade de interação com o ambiente e aprimoramentos estruturais que aumentem a robustez e a durabilidade do robô. Além disso, a arquitetura modular do código torna a programação mais acessível e intuitiva, facilitando sua aplicação e ampliação em contextos educacionais diversos.

Dentre as perspectivas para trabalhos futuros, destaca-se a oportunidade de melhorar o desempenho do sistema por meio da implementação processamento multithread, reduzindo latências e aumentando a eficiência computacional caso haja a substituição da atual placa controladora por um modelo mais avançado, capaz de oferecer maior poder de processamento e suporte para funcionalidades expandidas. Além disso, ajustes no hardware do robô podem ser feitos para regular a voltagem das rodas e garantir a mesma rotação caso o chassi seja alterado para um projeto mais robusto. Também é possível a geração de um material didático e planos de aula para a utilização do robô de forma a otimizar o uso. E por fim, a aplicação por meio de testes com usuários, para geração de dados de aprendizado e avaliação da aplicação prática e pedagógica do robô, obtendo visões e opiniões por parte dos professores. Uma das ideias é o envolvimento de professores na avaliação e na melhoria do protótipo para uso em sala de aula e possibilidade de aplicação transversais para ensino de conteúdos curriculares.

Referências Bibliográficas

OLIVEIRA, Prabhát K.; CARDOSO, Felipe; SOUSA, Marcos R.; DELGADO, Martha X. T. **Plataforma de baixo custo como alternativa para o Cubetto**. *Revista de Ciência da Computação*, na Universidade Estadual de Santa Cruz (UESC), 2020. Disponível em: <https://seer.ufrgs.br/index.php/reic/article/download/102747/59184/450344>. Acesso em: 05 out. 2024.

ESQUICHA-TEJADA, José et al (ANO). *Interactive Toy to Strengthen the Memory, Attention and Logic of Primary Education Students Using Sphero, Arduino and Neurosky MINDWAVE EEG*. pg 6, 7. Disponível em: <https://ceur-ws.org/Vol-2747/paper23.pdf> . Acesso em: 05 out. 2024.

BAKALA, Ewelina; GEROSA, Anaclara; HOURCADE, Juan Pablo; TEJERA, Gonzalo; PETERMAN, Kerry; TRINIDAD, Guillermo. *A Systematic Review of Technologies to Teach Control Structures in Preschool Education*. *Frontiers in Psychology*, v. 13, p. 1-19, set. 2022. Disponível em: <https://doi.org/10.3389/fpsyg.2022.911057>. Acesso em: 9 out. 2024.

D'ALESSANDRO, Giulia. *Kit robotico educativo per l'apprendimento del Coding nella scuola dell'infanzia*. 2022 <http://archspace.unicam.it:8080/xmlui/handle/1336/6451?show=full>. Acesso em: 9 out. 2024.

PAPERT, Seymour. *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, 1980. Disponível em: <https://mindstorms.media.mit.edu/>. Acesso em: 25 mar. 2025.

TEXAS ADVANCED OPTOELECTRONIC SOLUTIONS INC. *TCS3472 Color Light-to-Digital Converter with IR Filter*. Plano: TAOS, 2012. Disponível em: <https://www.taosinc.com>. Acesso em: 9 out. 2024.

VEIGA, Ernesto Fonseca; ARAÚJO, Wendi Müller; SILVEIRA JÚNIOR, Carlos Roberto da. Projeto de um robô de baixo custo para utilização como ferramenta de robótica educativa para escolas públicas. **Anais da Mostra Nacional de Robótica (MNR)**, Ensino Superior, Pós-graduação e Pesquisa, 2011. Disponível em: https://www.researchgate.net/profile/Carlos-Silveira-Junior-2/publication/307651847_PROJETO_DE_UM_ROBO_DE_BAI_XO_CUSTO_PARA_UTILIZACAO_COMO_FERRAMENTA_DE_ROBOTICA_EDUCATIVA_PARA_ESCOLAS_PUBLICAS/links/5d6d2786a6fdcc547d73195b/PROJETO-DE-UM-ROBO-DE-BAIXO-CUSTO-PARA-UTILIZACAO-COMO-FERRAMENTA-DE-ROBOTICA-EDUCATIVA-PARA-ESCOLAS-PUBLICAS.pdf . Acesso em: 03 set. 2025.