

Trabalhando o Pensamento Computacional Através de uma Interface Tangível para Programação de Robôs

Title: The Use of Computational Thinking through Tangible Interfaces for Robot Programming

Bruno Gonzaga de Mattos Vogel
Universidade Estadual de Feira de Santana (UEFS)
Departamento de Ciências Exatas (DEXA)
ORCID: 0009-0007-3023-1554
bugvogel@gmail.com

Claudia Pinto Pereira
Universidade Estadual de Feira de Santana (UEFS)
Departamento de Ciências Exatas (DEXA)
ORCID: 0000-0002-2552-2089
claudiap@uefs.br

Resumo

As interfaces tangíveis de usuário (TUI - Tangible User Interface) são ferramentas que podem se encaixar em um ambiente comum de interação do usuário e propõem a representabilidade de objetos virtuais através de objetos físicos. Uma das intenções é tirar vantagem das habilidades que existem na interação háptica para atingir um determinado objetivo, inclusive o de ensino e aprendizagem. Este trabalho propõe a construção de uma interface tangível para ser utilizada na programação de robôs, fazendo uso de conceitos como pensamento computacional (PC) e a taxonomia de Bloom como bases teóricas do ensino e do aprendizado. O usuário interage com a interface através de blocos, possuindo, cada um deles, uma função específica (frente, direita, esquerda, trás, loop e if). Com os blocos devidamente posicionados em uma plataforma, em sua ordem de execução, o robô pode executar a movimentação planejada. A validação da interface foi realizada com 10 estudantes do ensino fundamental I, com idades de oito a nove anos. No resultado foi possível observar uma boa avaliação para a usabilidade, além da aplicação dos conceitos da taxonomia de bloom. Quantitativamente, não foi possível observar uma diferença sigficativa das médias dos testes sobre o PC (pré e pós-testes), apesar das medidas estatísticas descritivas demonstrarem uma maior concentração de acertos nos testes após o experimento. Qualitativamente, os experimentos evidenciaram o uso de conceitos do PC como abstração, generalização, análise lógica, avaliação e o próprio conceito de algoritmo.

Palavras-Chave: Tecnologias de Comunicação e Informação; Interface Tangível; Programação de Robôs; Pensamento Computacional; Ensino e aprendizagem de algoritmos.

Abstract

Tangible User Interfaces (TUI) are tools that can fit into a common user interaction environment and propose the representability of virtual objects through physical objects. One of the intentions is to take advantage of the skills that exist in haptic interaction to achieve a certain objective, including teaching and learning. This article proposes the construction of a tangible interface to be used in robot programming, using concepts such as computational thinking (CT) and Bloom's taxonomy as theoretical bases for teaching and learning. The user interacts with the interface through blocks, each of which has a specific function (front, right, left, back, loop and if). With the blocks properly positioned on a platform, in their execution order, the robot can execute the planned movement. The interface validation was performed with 10 elementary school students, aged eight to nine years. In the result, it was possible to observe a good evaluation for usability, in addition to the application of the concepts of bloom taxonomy. Quantitatively, it was not possible to observe a significant difference in the averages of the tests on the CT (pre and post tests), despite the descriptive statistical measures showing a higher concentration of correct answers in the tests after the experiment. Qualitatively, the experiments showed the use of CT concepts such as abstraction, generalization, logical analysis, evaluation and the very concept of algorithm.

Keywords: Communication and Information Technologies; Tangible Interface; Robot Programming; Computational Thinking; Teaching and learning algorithms.

1 Introdução

No cenário atual, algo que desperta o interesse de muitos estudiosos é a utilização de recursos didáticos que apresentem novas soluções de ensino e aprendizagem (Ferri & dos Santos Rosa, 2016). No ensino de algoritmos e programação, as tecnologias da informação e comunicação (TICs) possuem em seu acervo, por exemplo, o uso de jogos digitais didáticos, sites, aplicativos, robótica, plataformas digitais, e Interfaces Tangíveis (TUIs – *Tangible User Interfaces*), como formas de introduzir o ensino de lógica ou, até mesmo, de outras áreas do conhecimento. Essa abrangência pode justificar o alcance de um público cada vez maior ao pensamento computacional (PC) (Viana & Raabe, 2018). Na Inglaterra e no Japão, por exemplo, o uso de algoritmos é apresentado como aprendizagem básica em suas matrizes curriculares (Berry, 2013).

A inserção do PC por meio do ensino de algoritmos pode despertar a atenção dos alunos no processo educacional durante as aulas, além de incorporar o uso da lógica para a resolução de problemas do cotidiano (Ferri & dos Santos Rosa, 2016). Molina et al. (2016) afirmam, em seu estudo, que o ensino de algoritmos também tem o objetivo de tornar o aluno mais produtivo para o mercado de trabalho, devido ao envolvimento multidisciplinar que caracteriza o uso da lógica.

Blikstein (2008) aponta que esta influência no mercado de trabalho é fruto de uma época de transição no mundo científico no qual o PC está transformando profundamente a academia e a indústria. A maioria das carreiras do século XXI requer o entendimento do PC, e as mais diversas áreas de atuação profissional necessitam de um conhecimento computacional mínimo para se tornarem competitivas no mercado (Seehorn et al., 2011). Blikstein (2008) inclusive apresenta, em sua pesquisa, que os alunos aprendem mais facilmente a 'ciência computacional' em detrimento da ciência tradicional, devido a uma série de fatores cognitivos, epistemológicos e motivacionais. A ideia, neste caso, é introduzir, de maneira preliminar na educação básica do aluno, o conhecimento de estruturas lógicas que os permitam desenvolver habilidades como raciocínio lógico, decomposição de uma solução e abstração de um algoritmo.

De forma a unir estas características positivas da construção do pensamento lógico com as vantagens de manipulação física, as TUIs, para programação, tornam-se inovadoras, na medida em que permitem o uso da computação em favor da educação sem necessariamente requerer o uso de computadores (Falcão & Gomes, 2007). Segundo Panaggio et al. (2019), as TUIs são uma maneira potencialmente facilitada de introduzir às crianças a programação e de promover um engajamento longo e forte com capacidade de envolvê-las durante atividades de aprendizado (Sylla, Branco, Coutinho, & Coquet, 2012), além de evocar sentimentos de felicidade, alegria e motivação (Posada, Hayashi, & Baranauskas, 2014).

Entendendo que as TUIs, pela sua característica de serem objetos físicos e palpáveis, podem favorecer a materialização de conceitos mais abstratos relacionados ao PC, sobretudo para crianças que estão em uma fase de desenvolvimento mais concreto, este trabalho apresenta uma interface tangível, na qual um usuário, sem conhecimento de qualquer linguagem de programação, terá a capacidade de criar padrões de movimentos para um robô e tomar decisões com base em determinada condição. Para tanto, busca-se responder as seguintes questões iniciais de pesquisa relacionadas respectivamente à usabilidade do artefato e ao potencial uso no processo de ensino aprendizagem do PC: (a) Como é avaliada a TUI em relação à usabilidade?; (b) Qual o potencial uso da TUI criada no processo de ensino e aprendizagem dos conceitos preliminares do PC?

Esta interface é composta por uma plataforma, que deverá ser preenchida por blocos que indicam uma determinada ação. Os blocos são de tipos diferentes, sendo eles: blocos de direção (frente, trás, direita e esquerda), blocos condicionais que indicam a condição de um objeto à frente do robô (sensor de distância) e um bloco de *loop*, que constrói um laço de repetição em torno de um agrupamento de blocos. Com isso, o usuário pode traçar diferentes objetivos de trajetórias a serem atingidas, que poderão variar de acordo com a criatividade e a composição do ambiente (obstáculos físicos). Por exemplo, o usuário pode fazer uso da interface para traçar uma rota em que o robô encontre a saída de um labirinto, ou para que chegue a um determinado ponto, passando por um caminho estreito.

Além do próprio desenvolvimento da ferramenta, objetiva-se também avaliá-la em relação ao desenvolvimento do PC dos participantes da pesquisa (pré e pós-testes), como percebem a sua usabilidade, e utilizar a taxonomia de Bloom (Armstrong, 2016) como instrumento para avaliar o ganho cognitivo a partir das atividades realizadas através do uso desta TUI.

Este artigo está organizado em cinco seções, além desta introdução. Na seção do quadro teórico, são apresentadas a fundamentação teórica a respeito das interfaces tangíveis, PC e taxonomia de Bloom, além dos trabalhos correlatos. Os materiais e métodos utilizados estão na terceira seção; na quarta seção, é detalhada a solução da interface tangível proposta, com os resultados e discussões na quinta seção e, por fim, as considerações finais.

2 Quadro Teórico

Esta seção se subdivide em quatro tópicos. São eles: interfaces tangíveis; pensamento computacional; taxonomia de Bloom e trabalhos correlatos.

2.1 Interfaces tangíveis

As investidas para fusão do mundo físico e digital começaram a ser solidificadas na primeira década deste milênio (Shaer & Jacob, 2009). Segundo Penaggio et al. (2019), a primeira iniciativa a cunhar o nome de interface tangível de usuário foi a de Suzuki e Kata (1995), na produção do AlgoBlock. Neste trabalho, os pesquisadores conseguiram desenvolver uma linguagem de programação baseada em blocos físicos, através dos quais a interface interagia com a movimentação de um submarino dentro de um labirinto.

Diferente de uma interface gráfica de usuário (GUI), ao invés de fazer com que os *pixels* se agrupem e deem forma a uma tela, as TUIs usam formas físicas que se encaixam no ambiente físico do usuário, tirando vantagem das habilidades que existem da interação háptica¹ (Ishii, 2008) e que, por isso, podem ser intuitivas e fáceis de aprender (Shaer & Jacob, 2009). Estas formas físicas podem ser objetos especificamente criados ou comuns do cotidiano e servirão de entrada para o processamento do sistema. A manipulação destes objetos consiste no fornecimento de informação ao sistema que, conseqüentemente, pode gerar uma saída diferente. A Figura 1 traz o funcionamento de uma TUI, que pode apresentar como entrada alguns tipos de elementos físicos, tais como uma luva, uma lâmpada ou uma garrafa. Em seguida, a informação coletada por meio

¹Que se refere ao tato; tátil

dos objetos é processada pelo sistema, gerando uma saída, que pode ser apresentada ao usuário por meio de um monitor, um sinal sonoro ou, até mesmo, um sistema robótico.

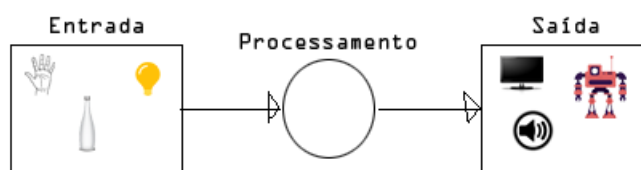


Figura 1: Diagrama geral de um sistema que faz uso de uma TUI. Fonte: Adaptada de (Nunes et al., 2011)

Mesmo com seus benefícios, as interfaces tangíveis podem apresentar uma dificuldade maior de *design* e construção em comparação com interfaces tradicionais. Alguns exemplos de desafios existentes na construção das TUIs são: a falta de dispositivos de entrada e saída padrão na construção de uma interação virtual-física e o cruzamento de informações dos limites disciplinares. O desafio da interação virtual-física está na atividade de criação de metáforas, símbolos ou objetos que dão forma física a informação virtual, e, assim, torna-se importante ao desenvolvedor identificar quais informações são melhor implementadas no meio físico ou virtual. Já em relação aos dispositivos de entrada e saída de uma TUI, é constatado que não existe um padrão na forma de utilização dos diversos dispositivos captadores de interações, tais como RFID (*Radio Frequency Identification* - Identificação por Rádio Frequência), sensores e microcontroladores. Uma diversidade de tecnologias ou instrumentos são utilizados para criar uma saída física e, em muitos casos, uma mesma TUI é prototipada várias vezes utilizando tecnologias diferentes.

Uma forma de segmentar alguns aspectos relacionados a uma TUI pode ser feita através das definições mencionadas por Fishkin (2004). Em seu estudo, é definida uma classificação utilizando dois parâmetros: a metáfora e a personificação. Enquanto a metáfora busca criar uma relação entre um objeto do cotidiano do usuário para utilizar as características e pontencialidade de utilização a seu favor, a personificação visa estudar a distância entre a entrada e a saída geradas, sendo que cada um desses possuem subtipos relacionados.

As metáforas podem ser de nome, de verbo e completa. Na metáfora de nome, o objeto de entrada utilizado se assemelha ao seu objeto virtual correspondente, porém provoca uma ação diferente; o símbolo de interrogação, por exemplo, é utilizado no final de uma sentença para indicar uma pergunta, contudo, em uma TUI composta com blocos que representam comandos a um robô, a interrogação pode representar uma estrutura condicional de linguagem de programação, exercendo assim uma ação diferente do que possui na realidade. Na de verbo, embora a aparência do objeto físico seja distinta da do objeto virtual, a ação gerada é a mesma. Por exemplo, a simbologia do bloco de 'andar para frente' é normalmente uma seta semelhante a um letreiro de direção, mas o seu correspondente virtual é um código-fonte mais complexo e distinto. Na metáfora completa, os objetos (virtual e físico) possuem uma forte relação semântica, como é o caso de um alto-falante ou um *buzzer*, que representam um dispositivo de saída de áudio e efetivamente exercem esta funcionalidade. Por fim, em alguns casos, pode ocorrer a ausência de metáfora em uma interface tangível, devido ao fato de os objetos físicos utilizados na interface em nada aparentarem os objetos virtuais correspondentes.

As personificações podem ser identificadas como completa, próxima, de ambiente, e dis-

tante. A personificação completa é quando a interface da entrada é exatamente a mesma da saída e seus efeitos são percebidos neste mesmo objeto. É possível vislumbrar esta personificação imaginando uma TUI em forma de mapa, que se utiliza do tato para que regiões sejam identificadas. Quando o objetivo é atingido, a resposta é emitida pelo próprio mapa, seja por som, por luz ou qualquer outro sinal, dessa forma, o sistema de resposta estaria embutido na própria TUI. A personificação próxima, como seu próprio nome já diz, trata-se de quando a interface tangível se encontra próxima a interface de saída, porém separados, como é o caso de um robô que é acionado por um conjunto de blocos que representam comandos. Neste caso, a resposta é visualizada pelo usuário por meio da movimentação do robô, que está próximo à interface de entrada. A personificação de ambiente ocorre quando a saída gerada pela interface de entrada gera efeitos no ambiente em que o usuário está inserido, como áudios que saem de caixas de sons espalhadas em uma sala, ou um conjunto de luzes que se acende ou modifica a sua cor como padrão de resposta. E finalmente, a personificação distante ocorre quando a interface de entrada se encontra distante da interface de saída do sistema em questão.

2.2 Pensamento computacional

O PC é uma habilidade que pode ser usada desde simples atividades da rotina do dia-a-dia ou, até mesmo, no planejamento de sistemas ou atividades complexas e abrangentes. Não necessariamente estas tarefas têm como alvo problemas computacionais. A ideia do PC não é programar nem fazer humanos pensarem como máquinas, mas sim a de criar uma forma dos humanos resolverem problemas (Wing, 2006). Esta habilidade engloba as capacidades de: decomposição do problema em pequenos pedaços, resolvendo cada um deles por vez; capacidade de abstração do problema em vários níveis, escolhendo uma boa forma de representação do mesmo; identificação de padrões e generalizações necessárias; e ter um pensamento voltado a avaliações de uma solução adotada, verificando os diferentes cenários possíveis (Csizmadia et al., 2015).

Em um projeto descrito por Barr e Stephenson (2011), é estudada a inserção do PC no currículo escolar nos ensinos primário e secundário. Foram reunidos 26 líderes de pensamento em um comitê para chegar a um consenso do que significaria o PC em uma matriz curricular. Durante esta reunião, o comitê destacou que, com o desenvolvimento do PC, os estudantes tornam-se engajados em utilizar novas ferramentas para resolver problemas e confortáveis com o processo de tentativa e erro existente na procura de uma solução. O grupo previu que estudantes com o PC poderiam seguir, por um modo mais fluído, na resolução de problemas, entendendo que um problema pode ser resolvido de várias formas diferentes.

O PC pode ser introduzido na educação através da utilização de um *hardware* ou *software*, porém não necessariamente a tecnologia é essencial para o fomento desta habilidade em sala de aula. Atividades desplugadas também podem ser utilizadas para isto (Mannila et al., 2014). A interdisciplinaridade também está envolvida com o fomento da habilidade do PC. Na matemática, um aluno, utilizando um enigma comum como "pense em um número e eu adivinharei", pode construir uma equação linear através de um algoritmo (Mannila et al., 2014).

A robótica na educação também pode fomentar o PC, não só através do uso de tecnologias, mas também através da computação desplugada. Dois colegas de classe podem assumir os papéis de programador e robô respectivamente, o primeiro emitindo ordens que, em conjunto, formam um algoritmo a ser executado pelo outro colega (na função de robô). Com esta estratégia, vários

objetivos podem ser traçados no intuito da construção de uma lógica para superar um obstáculo. Na disponibilidade de tecnologia e robôs reais, as variantes geradas pela existência de sensores, por exemplo, podem permitir uma variação saudável das soluções (Mannila et al., 2014).

Na área de jogos, a estratégia que alguns destes jogos propõem pode ser uma valiosa ferramenta na perspectiva educacional do PC. *Hay Day* ou *The Sims* são alguns exemplos (Mannila et al., 2014). Outros, como o CT *Pluzze* e o *Lightbot*² utilizam a programação em blocos de um robô, em escala digital, e tem como objetivo passar por alguns obstáculos e atingir um ponto específico.

2.3 Taxonomia de Bloom

A taxonomia de Bloom (TB) trata de delinear níveis hierárquicos de aprendizagem removido. Estes níveis cognitivos podem ser divididos em seis, e originalmente eles foram denominados como: conhecimento (nível da base), compreensão, aplicação, análise, síntese e avaliação (nível do topo) (Bloom, Krathwohl, & Masia, 1984). Em 2001, alguns destes níveis foram renomeados por meio de uma revisão publicada em meio científico (Armstrong, 2016), passando agora para os níveis cognitivos conhecidos como: recordar (nível da base), entender, aplicar, analisar, avaliar e criar (nível do topo). Estes níveis cognitivos associados à taxonomia são representados por meio de uma pirâmide, por ordem de complexidade, na qual a base representa o primeiro nível, menos complexo, e a ponta o último nível mais complexo. A etapa de recordar é aquela na qual há a busca e a reconexão com conhecimentos prévios. Em seguida, na etapa de entender, o estudante deve interpretar, inferir, comparar ou classificar. No momento de aplicar os conhecimentos adquiridos, o estudante deve implementar ou executar. Segundo Scott (2003) e Thompsom et al. (2008), o que caracteriza esta etapa é o aluno aplicar em um problema familiar um processo previamente conhecido, porém sem necessariamente conhecer previamente os dados do problema. Na etapa de analisar, serão realizadas atribuições e organizações. No penúltimo nível, em avaliar, devem ser realizadas verificações e críticas, e, por fim, a última etapa (criar) está atrelada a um planejamento ou uma produção (Armstrong, 2016). Thompsom et al. (2008) afirmam que, nesta etapa, o aluno é livre para juntar elementos que o levem a uma solução coerente e funcional do problema, como, por exemplo, inventar um algoritmo ou combinar algoritmos conhecidos de uma forma ainda não utilizada pelo aluno.

Alguns trabalhos já foram publicados na área de ensino de algoritmos utilizando a TB. Araújo et al. (2013) utilizaram a TB no ensino de raciocínio lógico e programação através da ferramenta *Scratch*. Os estudantes foram, por exemplo, provocados a recordar exemplos do seu próprio dia-a-dia, como trocar uma lâmpada, ir a escola ou fazer uma ligação de celular, introduzindo o conceito de algoritmos. Com isso, foi possível fazer com que o estudante recordasse e reconhecesse informações já adquiridas e guardadas na memória, como indicado no primeiro nível da TB. Em outro trabalho, desenvolvido por Jesus e Raabe (2009), um questionário voltado para o público-alvo de alunos iniciantes no aprendizado de algoritmos foi desenvolvido no sentido de identificar se alguns níveis da TB tinham sido atingidos no processo. Usando como base a TB revisada, o questionário apresentado continha dez questões e cada uma delas tinha o objetivo de identificar um determinado nível cognitivo da pirâmide.

²Disponíveis, respectivamente, em encurtador.com.br/xE156 e em loja virtual de aplicativos (*play store*).

2.4 Trabalhos correlatos

Nesta seção, são descritos alguns trabalhos que, apesar de apresentarem formas de uso distintas, se utilizam do conceito da aplicabilidade das interfaces tangíveis de usuário (TUIs). Também serão apontadas semelhanças e diferenças entre a ferramenta apresentada neste artigo e os estudos utilizados como quadro teórico.

O P-CUBE (Kakehashi, Motoyoshi, Koyanagi, Ohshima, & Kawakami, 2013) utiliza blocos para construção de um algoritmo que permite a programação de um robô, voltado para pessoas com deficiência visual. Os blocos são diferenciados de acordo com o relevo da superfície de cada face do cubo, indicando a direção para a qual o robô se movimentará. Além disso, em cada face de cada cubo existe uma indicação de um número em Braille, que determina a velocidade daquele movimento. Primeiro, no projeto P-CUBE, a plataforma dos blocos deve ser conectada ao computador para que os dados gerados sejam gravados em um *microSD*, e assim, em seguida seja embutido no robô, que executará o algoritmo.

Além do P-CUBE, citado anteriormente, existem outras três soluções semelhantes à proposta deste trabalho. Uma delas é o Kodockly, que é apresentada no artigo de Mohamed et al. (2021) e duas são soluções de mercado, sendo elas o Plobot³ e o Mi-go⁴. A base de funcionamento dos três robôs é semelhante, fazendo uso de blocos ou cartões, que dispostos em uma devida ordem enviam ao robô um algoritmo a ser seguido. Dentre estes produtos, o Plobot é o que possui a maior variedade de funcionalidades como *listen*, que realiza o comando seguinte caso ouça um bater de palmas; *feel* que executa o próximo comando, caso o robô seja inclinado ou movido fora do eixo; *dance* para simular uma dança e um canto; *pause* para pausar a execução do robô em determinado momento, dentre outros. O Mi-go e o Plobot possuem adesivos de pequenos mapas que podem ser colocados no ambiente para a realização de atividades, enquanto que o Kodockly realiza sua movimentação em uma plataforma específica. Um diferencial do Mi-go é que a funcionalidade de rotação e o movimento do robô pode ser realizado de forma mais precisa, com ângulos de curva e distâncias de locomoção menores, ademais, no decorrer da execução do código o usuário pode visualizar qual parte exatamente está sendo executada (o bloco fica verde). Embora também possua funcionalidades semelhantes, para o Kodockly foram realizadas validações quanto a usabilidade e quanto ao ganho de aprendizagem, apresentando resultados positivos em relação ao potencial de gerar o que o autor chama de ganho de aprendizagem.

O CardBot (Barros, Torres, Burlamaqui, & Natal, 2014), assim como o P-CUBE, também é uma solução voltada para que pessoas com deficiência visual construam algoritmos para a programação de robôs através da programação tátil. O seu método consiste na utilização de um aplicativo que realiza leituras de *QR Codes* correspondentes a ações do robô. Essas leituras são enviadas via *bluetooth* e cada um desses cartões possui um formato distinto para que a pessoa com deficiência visual possa discriminá-lo utilizando o tato.

Outros resultados podem ser obtidos no uso das TUIs na educação. O Mapa tátil desenvolvido por Júnior et al. (2018) representa os estados da região sudeste do Brasil. Neste mapa, cada estado é devidamente distinguido por meio de uma elevação específica para cada um deles (uso do tato). No mapa tátil, o usuário é direcionado a um novo passo a cada vez que soluciona o seu respectivo objetivo, funcionando como uma nova fase que o usuário deve superar para seguir para

³Disponível em: <https://www.kickstarter.com/projects/1365116576/plobot-a-robot-coding-companion>

⁴Disponível em: <https://migobot.com/pt/kit-mi-go>

a próxima. Reproduções sonoras explicativas são ativadas a cada passo para o qual o estudante é direcionado, tendo cada um dos passos seus próprios objetivos.

Assim como esses trabalhos que apresentam o tato como uma das formas de interação do estudante com deficiência visual, no trabalho de Lozano et al. (2018), o foco girava em torno necessariamente do estímulo deste sentido sensorial. O objetivo foi o de produzir uma caixa com alguns compartimentos contendo objetos diferentes. Os tipos de objetos variavam no sentido de identificar formas, texturas, números, dentre outros aspectos. A identificação do sistema era feita por meio de um aplicativo leitor de uma *tag* NFC (*Near Field Communication* - Comunicação por campo de proximidade). Cada item na caixa, em seu compartimento, tinha uma *tag* que identificava o seu tipo, e assim, quando o estudante recebia uma orientação sonora com descrição do objetivo de identificação, ele realizava uma busca pelo objeto por meio do sentido tátil, apresentando-o quando o encontrava.

O Topobo é outra solução que embarca, em objetos comuns, o uso da tecnologia (Farr, Yuill, & Raffle, 2010). Neste trabalho, foi feito o exame dos impactos do uso de TUIs sob o interesse e a motivação de brincar das crianças com Transtorno do Espectro Autista (TEA). O Topobo funciona semelhante à utilização de um LEGO®, que permite a conexão de vários blocos, e somente quando conectados um com o outro podem exercer funções de movimento. Dessa forma, a criança pode conectar pernas, braços e torso para caracterizar um avatar humanóide ou não, e realizar movimentos. Diferente do LEGO®, em que também existe a montagem, no Topobo o avatar pode ser construído sem comprometer a força da estrutura e fazê-lo desmoronar. Os resultados deste trabalho apresentaram uma ferramenta que demonstra um exercício de maior colaboração social entre os participantes, se comparado a uma ferramenta de montagem semelhante como o LEGO®.

Semelhante ao Mi-go, a TUI deste artigo apresenta uma comunicação sem fio (*bluetooth*), distinguindo-se do P-CUBE, que exige um dispositivo de armazenamento (*microSD*) para transmitir os dados ao robô, ou do Kodocly e do Plobot, ferramentas em que o robô realiza diretamente a leitura dos comandos. Além disso, também não exige o uso de aplicativos, o que limitaria o seu acesso apenas a crianças de posse de um *smartphone*, como no trabalho do CardBot. O Mapa tátil apresenta restrições relacionadas ao número de etapas a serem alcançadas, as quais engessam o seu uso somente aos modos de seus criadores, sem permitir liberdade de adaptação e construção, conforme interesses dos usuários. Na TUI de blocos proposta neste trabalho, é possível organizar o ambiente de diferentes maneiras, permitindo liberdade de ambientação e criatividade do usuário. O uso da tecnologia NFC, sinal sonoro e do tato do trabalho de Lozano et al. (2018) e do Topobo contribuíram como inspiração para a implementação da interface deste trabalho, como elementos de interação.

Adicionalmente, com exceção do Kodocly que foi submetido a uma avaliação do ganho de aprendizagem, uma avaliação quantitativa sobre a aprendizagem dos conceitos do pensamento computacional é um diferencial encontrado neste trabalho em relação aos demais apresentados, utilizando-se de um instrumento específico para este fim. Além desta avaliação, foi realizada uma avaliação qualitativa através da observação dos comportamentos e falas dos participantes, e associação dos resultados das atividades realizadas com os níveis cognitivos da taxonomia de Bloom.

3 Métodos

Este é um projeto que envolveu pesquisa e desenvolvimento de um artefato tecnológico para o ensino e a aprendizagem do PC através de uma interface tangível, que conduziu uma avaliação experimental para a sua validação. Para tanto, a atividade inicial de todo o processo, seja da solução computacional ou das etapas de pesquisa, foi a **revisão bibliográfica**, necessária para o levantamento do estado da arte, de soluções semelhantes (trabalhos relacionados) e para a aproximação com a temática (Webster & Watson, 2002).

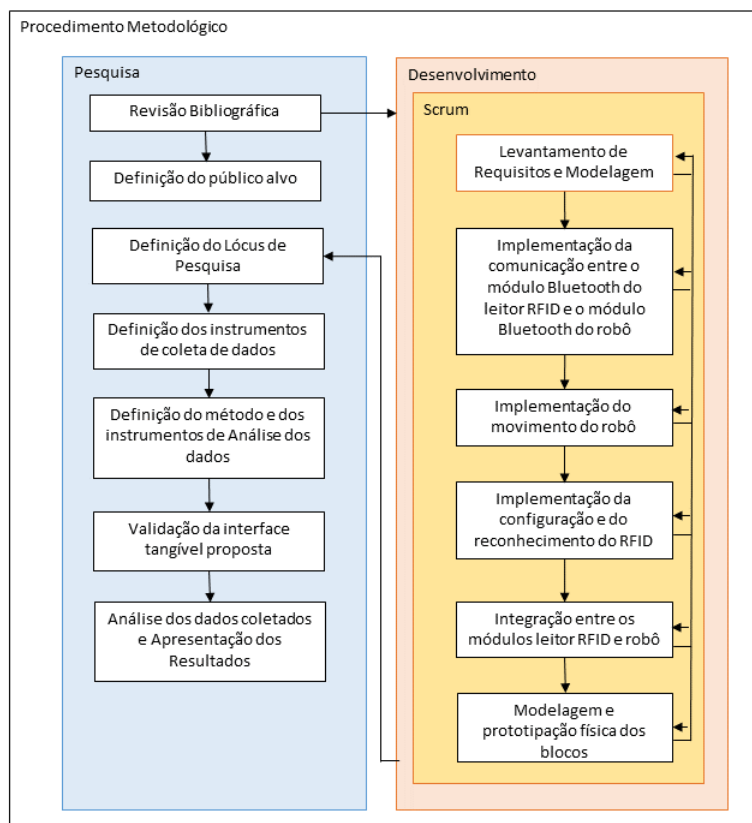


Figura 2: Etapas de desenvolvimento do projeto. Fonte: Própria, 2021

3.1 Etapa de desenvolvimento

Conforme demonstra a Figura 2, após a **revisão bibliográfica**, uma das etapas seguintes consistiu no **levantamento de requisitos e modelagem da interface tangível**. Nesta etapa, a interface tangível foi considerada como uma subdivisão entre dois componentes: a base de leitura RFID e o robô. Como requisitos, foram identificados: o estabelecimento de uma conexão *bluetooth*; a configuração e o reconhecimento das *tags* RFID, cada uma delas com a identificação de um comando de execução do algoritmos (i.e. frente, trás, direita, esquerda, *loop*, *if* e parada); programação de movimentação do robô; programação de verificação de obstáculos e utilização do bloco de comando '*if*' pelo usuário; a integração da lógica de envio e recebimento de um caractere entre a base de leitura do RFID e do robô respectivamente; e a modelagem e o protótipo físico dos blocos

de comando.

Na fase de modelagem, foi construído o diagrama de atividade (apresentado na Seção de Implementação da Solução). Esse diagrama tem o objetivo de definir o funcionamento lógico das atividades realizadas entre usuário, base de leitura RFID e o robô.

Na etapa de **Implementação da comunicação entre o módulo *bluetooth* do leitor RFID e o módulo *bluetooth* do robô**, foi realizado o monitoramento do recebimento dos dados via *bluetooth*. Nesta etapa, foi possível experimentar o envio e o recebimento de diferentes caracteres; a forma de recebimento e envio e a tipagem da informação de envio e recebimento (*bytes*).

Na etapa de **implementação do movimento do robô**, a programação inicial foi a de fazer o robô girar para a direita. Desta forma foi possível verificar qual o sentido da rotação dos motores seria necessário para realizar o movimento e o quanto o robô se desloca a depender do tempo que o motor é acionado.

Por padrão, apesar de possuir ID único previamente gravado de fábrica, o RFID também possui a capacidade de armazenar valores personalizados, que precisam ser devidamente gravados. Na etapa de **implementação da configuração e do reconhecimento do RFID**, portanto, foram gravados os valores correspondentes aos blocos frente, trás, direita, esquerda, *if*, *loop* e *stop*. Na etapa de **integração entre os módulos do leitor RFID e do robô** restou a implementação da lógica de reconhecimento destes caracteres para o armazenamento dos comandos. No processo de **modelagem e prototipação física dos blocos**, foi inserido, em cada um deles, uma *tag* RFID e a sua construção foi feita utilizando o acrílico. Entretanto, os primeiros testes com blocos foram realizados com a utilização de papelão (protótipo de baixa fidelidade) para a verificação prévia do funcionamento do leitor de rádio frequência utilizado.

Todas as etapas de desenvolvimento apresentam uma característica retroativa no decorrer da implementação ou de possível retrocesso, como é indicada pelas setas na região lateral do quadro de implementação da Figura 2. Isto foi característico do processo de desenvolvimento por efeito do surgimento de erros ou da necessidade de planejamento antecipado de etapas posteriores.

O Scrum foi escolhido como a metodologia de desenvolvimento pelas características atrativas para este projeto, tais como agilidade no desenvolvimento, contato antecipado com o mínimo produto viável do projeto e a facilidade de adaptação às condições de desenvolvimento da equipe (dos Santos Soares, 2004). A agilidade do *Scrum* permite a visualização do projeto em subdivisões e a implementação é realizada em pequenos pedaços. Esses pedaços são denominados de *sprints* e a listagem deles é conhecida como *product backlog*. No diagrama da Figura 2, uma etapa de desenvolvimento pode significar a existência de várias *sprints*, ou seja, pode ser dividido em vários outros pedaços, além de que o processo retroativo indicado pelas setas laterais implica na possibilidade de inclusão em uma *sprint* de alguma atividade de uma etapa anterior ou posterior. Por consequência desta divisão, o mínimo produto viável pode ser alcançado por meio da entrega das *sprints*, permitindo a realização de testes prévios.

3.2 Etapa de pesquisa

Neste subtópico as etapas de pesquisa foram segregadas em Participantes (3.2.1), com especificações do participantes, suas devidas características e do lócus de pesquisa; Instrumentos (3.2.2) com a apresentação dos questionários e suas respectivas adaptações; e por fim, Procedimentos de

coleta e de análise de dados (3.2.3) apresentando *softwares* e procedimentos realizados na coleta e análise dos dados.

3.2.1 Participantes

Esta pesquisa foi realizada no município de Serrinha-BA em um colégio da rede privada, e por conveniência, foi definida uma amostra com 10 alunos do terceiro ano do ensino fundamental I, com idades entre oito e nove anos, sendo cinco meninos e cinco meninas. Todo o processo de pesquisa ocorreu em ambiente escolar, em salas de aula disponibilizadas pela instituição de ensino. A pesquisa seguiu o rigor ético necessário para a aplicação de pesquisas envolvendo seres humanos, com a devida autorização da instituição e do Comitê de Ética em Pesquisa (CEP), com o parecer de número 5.897.868 e CAAE 56881922.3.0000.0053 do projeto guarda-chuva do qual faz parte. Foram realizados dois encontros com os participantes em dias consecutivos, sendo que em cada um deles houve uma carga horária de aproximadamente uma hora. Todos os participantes e seus responsáveis realizaram a devida autorização e consentimento através de Termo de Assentimento Livre e Esclarecido (TALE) e Termo de Consentimento Livre e Esclarecido (TCLE), respectivamente assinados.

A **revisão bibliográfica** foi etapa necessária para a pesquisa a respeito de trabalhos correlatos, os quais serviram como base para a etapa de **definição do público-alvo**. A **definição do lócus de pesquisa** se deu pela proximidade da instituição de ensino e pela pronta disponibilização do horário escolar para a realização dos experimentos, tanto pela instituição quanto pela professora da turma.

3.2.2 Instrumentos

Na etapa de **definição dos instrumentos de coleta de dados**, foram consideradas duas características que seriam avaliadas quantitativamente: a usabilidade e o PC. A usabilidade foi avaliada com base no paradigma SUS (*System Usability Scale* - Sistema de Escala de Usabilidade). O SUS surgiu no ano de 1996 com o objetivo de obter uma mensuração rápida a partir de uma avaliação de usabilidade coerente com as necessidades dos sistemas computacionais (Lewis, 2018).

Existe uma lista padrão de questões criadas por John Brooke (Lewis, 2018) (Brooke et al., 1996), contendo um total de 10 afirmações relacionadas à usabilidade. Entretanto, neste trabalho, sete das 10 questões foram utilizadas (Quadro 1) e três delas, Q5, Q6 e Q7, removidas por não apresentarem relação com o projeto proposto ou por se mostrarem muito complexas para o entendimento do público-alvo (crianças de oito a nove anos). Para cada uma das afirmações da lista proposta no Quadro 1, foi utilizada a escala *likert* (Discordo Totalmente, Discordo Parcialmente, Indiferente, Concordo Parcialmente e Concordo Totalmente) adaptada visualmente para o público infantil.

Para a avaliação do PC, foi utilizado o questionário publicado por Román-Gonzalez, Pérez-González e Jiménez-Fernández (2015) e traduzido por Brackmann (2017), por apresentar uma abordagem quantitativa e por se tratar de um instrumento com evidências favoráveis de validade e de fidedignidade. Houve dois momentos de uso deste instrumento, um anterior ao experimento (pré-teste), que fez uso das questões um, dois, três, seis e nove, e outro posterior (pós-teste), que fez uso das mesmas cinco questões citadas anteriormente e da questão onze, questão do exemplo

- Q1- Eu gostaria de usar esse sistema frequentemente
- Q2- Achei o sistema desnecessariamente complexo
- Q3- Eu penso que o sistema é fácil de usar
- Q4- Eu precisaria de suporte técnico para usar esse sistema
- Q5- Eu acho que as várias funções do sistema estavam bem integradas
- Q6- Eu acho que o sistema apresentou muita inconsistência
- Q7- Eu imagino que a maioria das pessoas poderiam aprender a usar este sistema muito rápido
- Q8- Achei o sistema estranho de usar
- Q9- Me senti confiante usando este sistema
- Q10- Tive que aprender muita coisa antes de começar a usar este sistema

Quadro 1: Questionário de usabilidade. Fonte: (Lewis, 2018)

II (Adaptada sem a resposta) e questão cinco, dispostas nesta respectiva ordem, tendo as três novas questões um nível de complexidade superior às anteriores. A seleção de questões e a respectiva redução do instrumento aconteceram para garantir a aplicação somente de questões que não apresentassem um nível de complexidade muito elevado e para não ultrapassar o tempo disponibilizado pela escola. A Figura 3 demonstra uma das questões utilizadas no processo; as demais questões citadas podem ser acessadas em Brackmann (2017).

Pergunta 6 - Quantas vezes a sequência abaixo deve ser repetida para levar o "Pac-Man" e o fantasma pelo caminho indicado?

Quantas vezes a sequência abaixo deve ser repetida para levar o "Pac-Man" e o fantasma pelo caminho indicado?

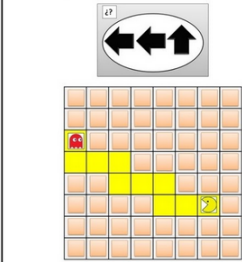
| | |
|--|---|
| <p>Quantas vezes a sequência abaixo deve ser repetida para levar o "Pac-Man" até o fantasma pelo caminho indicado?</p>  | <p>Alternativa A × 2</p> <hr/> <p>Alternativa B × 1</p> <hr/> <p>Alternativa C × 4</p> <hr/> <p>Alternativa D × 3</p> |
|--|---|

Figura 3: Questão 6 utilizada no questionário sobre pensamento computacional. Fonte: (Román-Gonzalez et al., 2015) e (Brackmann, 2017)

A intenção de trabalharem com as mesmas cinco questões no pré e no pós-testes foi de verificar se houve evolução nas respostas apresentadas, e portanto ganho no desenvolvimento do PC (um erro em uma questão poderia ser um acerto posterior, ou uma questão incompleta poderia se transformar em uma resposta completa, por exemplo). Já em relação às três novas questões adicionadas ao pós-teste, a intenção foi de que os participantes pudessem lidar com situações inusitadas e diferentes das anteriores, elaborando novas respostas e avançando no nível de complexidade

cognitivo. Por fim, nesta etapa também foram acrescentadas questões para o levantamento de perfil dos voluntários: sexo, idade, definição do grau de afinidade com a tecnologia, tempo diário de acesso à internet e contato com a robótica em algum outro momento anterior.

3.2.3 Procedimento de coleta e análise de dados

A etapa de **definição dos métodos e dos instrumentos de análise dos dados** implicou na utilização do *software* SPSS (*Statistical Package for the Social Sciences*). Esta análise foi realizada a partir da quantidade de acertos obtidos por cada participante da pesquisa no instrumento de avaliação do PC adaptado (somente cinco questões no pré-teste e oito questões no pós-teste). Já os dados coletados no instrumento de usabilidade foram avaliados de acordo com a perspectiva e a afirmação da questão, visto que a discordância ou a concordância poderia representar um aspecto positivo da interface tangível. A análise qualitativa foi realizada por meio da observação do comportamento das crianças ao longo do experimento, tais como falas, expressões e outras reações espontâneas dos participantes.

O processo de **validação da interface tangível proposta** foi feito por meio de um roteiro contendo seis passos. No primeiro momento, foi realizada uma **explicação breve** sobre o objetivo do experimento, que era realizar algumas atividades que estimulassem o PC, além de avaliar os resultados destas atividades e a usabilidade da ferramenta. A segunda etapa foi a **explicação do conceito de algoritmo e a sua relação com a programação e a robótica** (Figura 4 A), além da exemplificação de algoritmos, como a troca de uma lâmpada, por exemplo. Logo após a explicação, foram aplicadas as questões de levantamento de perfil assim como as questões para a **avaliação prévia do PC** (pré-teste) dos participantes.



Figura 4: Explicação inicial sobre algoritmos (Figura A). Atividade de computação desplugada (Figura B). Fonte: Própria, 2021

Na etapa seguinte deste roteiro, foi realizada uma **atividade de computação desplugada** (Figura 4B). Nesta etapa, um dos participantes assumiu o papel de robô enquanto outros montaram um algoritmo utilizando os blocos de acrílico. Após a montagem física do algoritmo, o participante, que se passava por um robô, deveria executar os movimentos com o intuito de realizar os passos definidos no algoritmo, inclusive o desvio de algum obstáculo. Na quinta etapa, **utilizou-se a interface tangível por completo** (partes física e lógica), com o robô e os blocos de acrílico para resolver um problema proposto. Por fim, na última etapa, foram aplicados instrumentos de **avaliação do PC posterior às atividades realizadas e a validação da usabilidade da interface tangível**.

Após a coleta de dados, foi realizada a **análise dos dados coletados e a apresentação dos resultados**, momento no qual os dados obtidos foram testados e comparados por meio de tabelas, gráficos e instrumentos estatísticos com o objetivo de observar os possíveis impactos do uso da interface tangível e das atividades realizadas no PC dos participantes, assim como a usabilidade da ferramenta.

4 Implementação da Solução

Nesta seção é apresentado o funcionamento da ferramenta, demonstrando os blocos construídos e a descrição do fluxo do diagrama de atividades. Também será descrita a etapa de construção e prototipagem dos blocos de comandos, apresentando a sua modelagem e evolução.

4.1 Funcionamento da interface tangível

Na utilização desta interface, o usuário dispõe de uma composição de vinte e dois blocos que podem ser combinados entre si para solucionar problemas que envolvam a transição de um robô em meio a uma determinada disposição do obstáculo previamente construído ou até um determinado ponto demarcado. O uso de vinte e dois blocos foi limitado devido à quantidade de *tags* RFID disponíveis no acervo, entretanto, o usuário pode recombina novos blocos para montar uma continuação do algoritmo já lido e armazenado pelo leitor, caso não tenha utilizado o bloco de finalização anteriormente. A imagem de cada tipo de bloco pode ser visualizada na Figura 5.

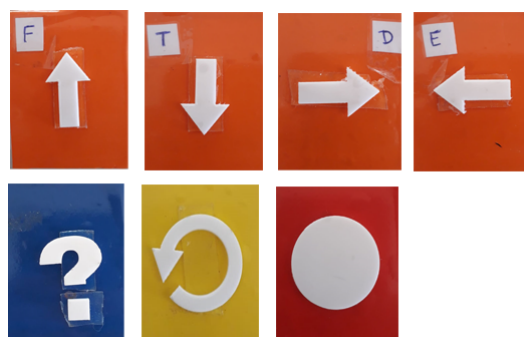


Figura 5: Blocos criados para serem utilizados na criação de um algoritmo. Fonte: Própria, 2021

Para cada tipo de bloco existe uma cor associada à sua função. Para os blocos de direção, todos são caracterizados pela cor laranja seguido da letra inicial da direção e da seta. O bloco condicional é reconhecido pela cor azul e pelo símbolo de interrogação, fazendo uma alusão ao processo de pergunta e resposta e ao operador ternário existente na maioria das linguagens de programação. O símbolo do bloco amarelo (*loop*) é o de uma seta fazendo um círculo, indicando um processo que se repete, sempre voltando ao início. Por fim, o bloco vermelho com um círculo no meio simboliza o fim do algoritmo.

O processo geral para o funcionamento da interface tangível compreende-se em ligar o **robô** e a **base de leitura RFID** dos blocos. Logo em seguida, o módulo *bluetooth* do robô automaticamente se conecta com o módulo *bluetooth* da base de leitura RFID. A partir daí, o usuário pode,

observando a posição em que o robô está, tentar montar pela primeira vez um algoritmo que o possibilite cumprir o desafio. Após a composição dos blocos na estrutura de acrílico, basta que o usuário passe a fila de blocos sob o leitor RFID. Caso o bloco de finalização esteja presente neste algoritmo, assim que for realizada a leitura deste bloco, o robô inicia o seu movimento (Figura 6). Para cada bloco, a Base de leitura RFID lê o caractere da *tag* RFID dentro do bloco e aciona um sinal sonoro para que o usuário entenda que aquele bloco já foi lido; em seguida, o caractere é enviado para o robô através do sinal *bluetooth*.

O robô possui dois estados de funcionamento: o estado de escuta (para o recebimento de um caractere) e o estado de movimentação. No estado de escuta, o robô pode receber caracteres a serem armazenados no *array* padrão de comandos ou no *array* de comandos condicionais. O *array* padrão de comandos é o algoritmo que o robô seguirá inicialmente, contudo, caso verifique a existência de um obstáculo deverá executar os comandos armazenados no *array* de comandos condicionais.

Por *default* o robô inicia suas atividades no estado de escuta, recebendo os caracteres a serem armazenados no *array* padrão de comandos. Caso o robô receba os caracteres 'f' (frente), 'b' (trás), 'l' (esquerda), 'r' (direita), 'o' (loop), armazena o comando no *array* padrão de comandos e o processo de leitura, envio e recebimento do bloco transcrito anteriormente se repete (Figura 6). Neste ponto, caso receba o caractere 'i' (simbolizando o bloco 'if', bloco azul da Figura 5), o robô também repete todo o processo de leitura, envio e recebimento, porém, antes disso, entra em estado de recebimento dos caracteres destinados ao *array* de comandos condicionais. Neste caso, portanto, os próximos caracteres recebidos pelo robô seriam armazenados no *array* de comandos condicionais (e.g. os comandos/caracteres 'f', 'b', 'l', 'r', 'o') e novamente o processo de leitura, envio e recebimento se repete. Para que o robô saia do estado de recebimento de comandos destinados ao *array* de comandos condicionais o robô deverá, neste ponto, receber o caractere 'i' novamente. Por fim, para finalizar o processo de armazenamento de comandos, o robô deve receber o caractere '\$' e armazená-lo no *array* padrão de comandos.

Após o armazenamento do '\$', o robô segue para o estado de movimentação. Todo caractere a ser enviado pelo usuário durante este estado será perdido, por isso o bloco de finalização (bloco vermelho da Figura 5) deve ser sempre o último bloco a ser utilizado. No estado de execução, o primeiro passo é executar os comandos armazenados no *array* padrão de comandos, na ordem recebida (Figura 6) e, em seguida, verificar se existe um obstáculo a sua frente. Se o robô não estiver executando o *array* de comandos condicionais e não encontrar obstáculo algum, deve executar o próximo comando. Se, por ventura, neste ponto, o robô encontrar um obstáculo e não foi armazenado qualquer comando condicional durante o estado de escuta (i.e. o usuário não fez uso do bloco 'if'), o robô encerra a movimentação para evitar que o robô colida com o obstáculo; caso contrário, o robô inicia a execução dos comandos armazenados no *array* de comandos condicionais, na ordem recebida.

Cada vez que executar um comando do *array* de comandos condicionais, o robô identifica se é o fim deste *array*. Em caso afirmativo, o robô volta ao fluxo de execução dos comandos do *array* padrão; em caso negativo, verifica a existência de outro obstáculo à sua frente. Como na atividade de verificação da existência de um obstáculo o robô está executando o *array* de comandos condicionais, caso identifique outro obstáculo a sua frente encerra a movimentação de imediato.

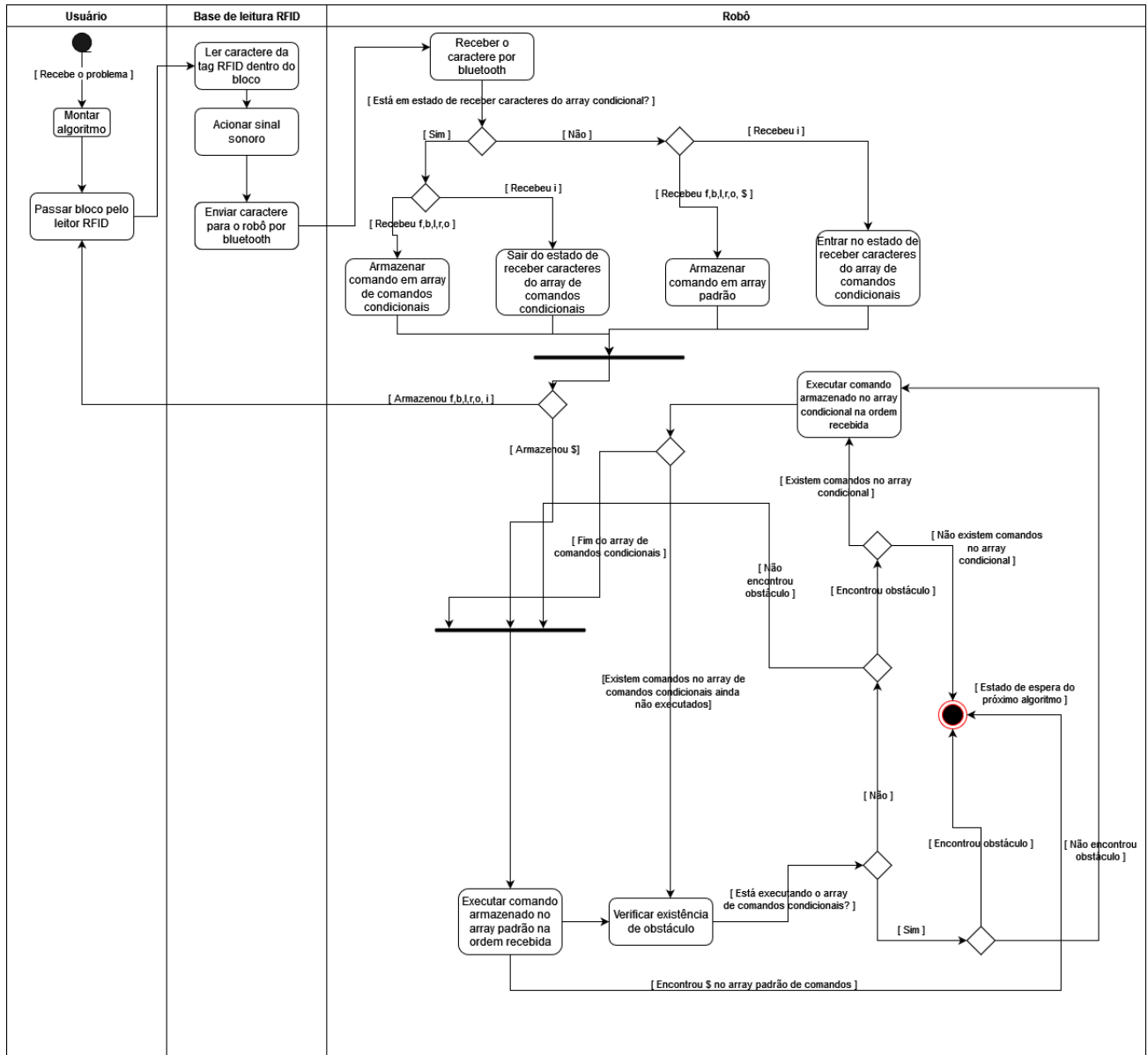


Figura 6: Diagrama de atividades do funcionamento da ferramenta. Fonte: Própria, 2021

Quando o robô encerra a movimentação, chegando ao ponto final do diagrama de atividades da Figura 6, entra em estado de espera do próximo algoritmo. Isto significa que o robô pode receber um novo algoritmo após encerrar a movimentação, reiniciando todo o fluxo de atividades.

O processo para resolver os problemas propostos passa também pelo exercício de tentativa e erro. É possível que o problema não seja resolvido em uma única e exclusiva tentativa, e, portanto, a ideia é permitir que o usuário consiga inferir o senso de distância que o robô pode percorrer em cada passo, e assim evoluir o seu algoritmo progressivamente.

A interface conta com um mecanismo de interrupção que pode ser utilizado em qualquer algoritmo, contudo, é indicado para os casos de *loop* nos quais o algoritmo pode acabar não realizando os movimentos esperados, ou nos casos em que o sensor de distância não identifique

algum obstáculo, com a possibilidade de colidir com outros objetos. O mecanismo consiste no uso de um potenciômetro. Quando o usuário gira o potenciômetro para a direita, o sinal lógico 'alto' é enviado ao robô, forçando a sua parada, e quando o usuário gira o potenciômetro para a esquerda, o sinal lógico 'baixo' é enviado ao robô, reiniciando o ciclo de atividades do robô. Na Figura 7 (A), é possível visualizar este mecanismo (cinza) fixado na face superior da base de leitura RFID. Esta imagem traz também o leitor RFID (azul) e os suportes laterais ao leitor (transparentes), que são utilizados para que a fila de blocos (Figura 8 B) seja corretamente translocada sob o leitor, e evite erros durante a leitura das tags.

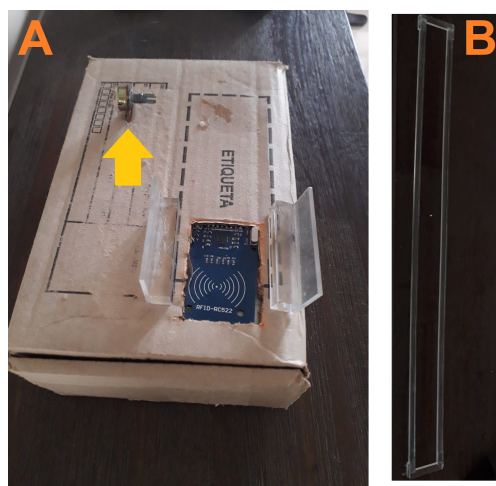


Figura 7: Interface do leitor dos blocos a ser utilizada (Figura A). Suporte de acrílico para o enfileiramento dos blocos (Figura B). Fonte: Própria, 2021

Uma estrutura de acrílico (Figura 7 B), com 94,1 centímetros de comprimento e 7,2 centímetros de altura, foi confeccionada para armazenar vinte blocos enfileirados. O usuário irá dispor os blocos na ordem em que os passos deverão ser executados pelo robô, exceto nos casos em que o usuário fizer uso dos blocos condicionais 'if' que pode estar em qualquer localização do suporte antes do bloco de parada.

Um exemplo de uma situação problema que pode ser solucionada com a utilização desta interface tangível é apresentado na Figura 8 (A). No desafio em questão, o robô se encontra a 40 centímetros de distância de um obstáculo que possui 44,5 centímetros de comprimento e 7 centímetros de largura. O objetivo final (ponto em que o robô deve estar ao final do algoritmo) está a 20 centímetros de distância do obstáculo. Considerando que o robô, ao se movimentar para frente ou para trás, move-se aproximadamente em 20 centímetros e gira para a direita e para a esquerda em um ângulo aproximado de 90 graus, foi realizada a montagem de um algoritmo que solucionasse o desafio (Figura 8 B).

A posição inicial do robô é de 40 centímetros de distância do objeto e encontra-se virado em direção ao mesmo, portanto, no primeiro passo o robô deve ir para frente, locomovendo-se em direção ao objeto a uma distância de aproximadamente 20 centímetros. Andar somente 20 centímetros para frente garante que o robô não chegue a alcançar o objeto, que o sensor de distância não identifique o obstáculo e que não seja necessária a execução do bloco condicional Implementado na Figura 8 (B). Em seguida, o robô deve virar 90 graus à direita e andar a uma distância de aproximadamente 40 centímetros (dois passos para frente); no quinto passo, ele deve

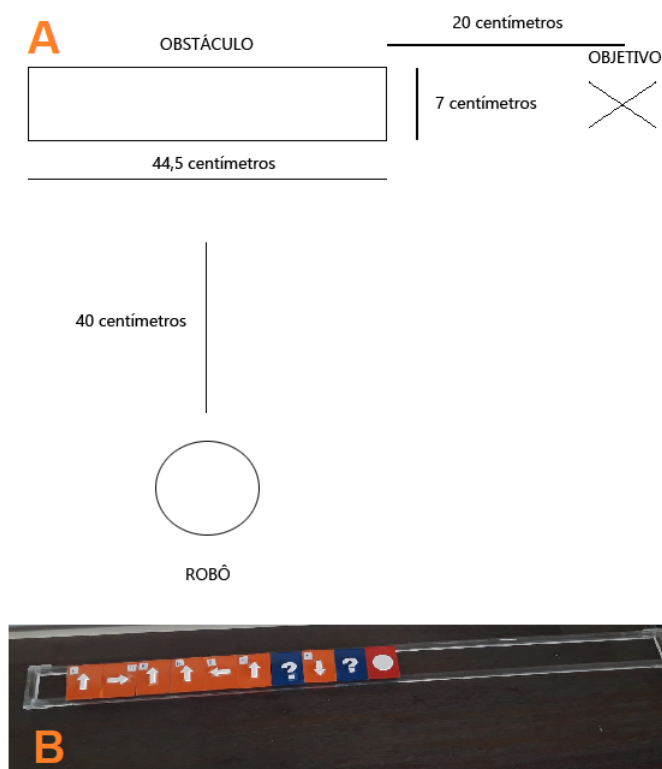


Figura 8: Problema exemplo para contornar um obstáculo utilizando a interface tangível (Figura A). Algoritmo de solução do problema apresentado (Figura B). Fonte: Própria, 2021

virar 90 graus para a esquerda, posicionando-se em direção ao objetivo. Neste ponto, o robô encontra-se a uma distância aproximada de 20 centímetros do objetivo; portanto, deve se mover 20 centímetros para frente. Como na execução do algoritmo implementado o robô não irá identificar o obstáculo presente na Figura 8 (A), o bloco condicional presente na Figura 8 (B) será acionado somente se encontrar qualquer outro obstáculo durante o percurso traçado. Neste exemplo de algoritmo, caso encontre um obstáculo, o robô irá se mover para trás.

O sensor de distância utilizado possui uma amplitude de atuação de aproximadamente 1 centímetro e reconhece um objeto a aproximadamente 6 centímetros de distância. A largura completa da frente do robô é de 23 centímetros, sendo que o sensor está posicionado no centro da frente do robô (Figura 9 A). Isto implica em ocorrências nas quais o robô pode colidir com um objeto logo a frente pelo fato de o sensor não estar devidamente alinhado com o objeto. Na Figura 9 (A), é apresentado um esquema hipotético no qual o robô não identificaria a existência de um obstáculo e a sua roda esquerda colidiria.

Em virtude desta limitação, foi confeccionado um obstáculo feito sob medida. O obstáculo é feito de madeira e consiste em uma parede com 35,5 centímetros de altura e 44,5 centímetros de comprimento e 7 centímetros de largura, na tentativa de evitar este tipo de ocorrência. Além disso, no processo de validação da ferramenta realizado pelo pesquisador, a posição inicial do robô foi determinada previamente, como mais uma forma de evitar este tipo de ocorrência. A Figura 9 (B) apresenta o obstáculo desenvolvido, utilizado no problema no qual o objetivo era encontrar este obstáculo e retornar o robô ao ponto de partida.

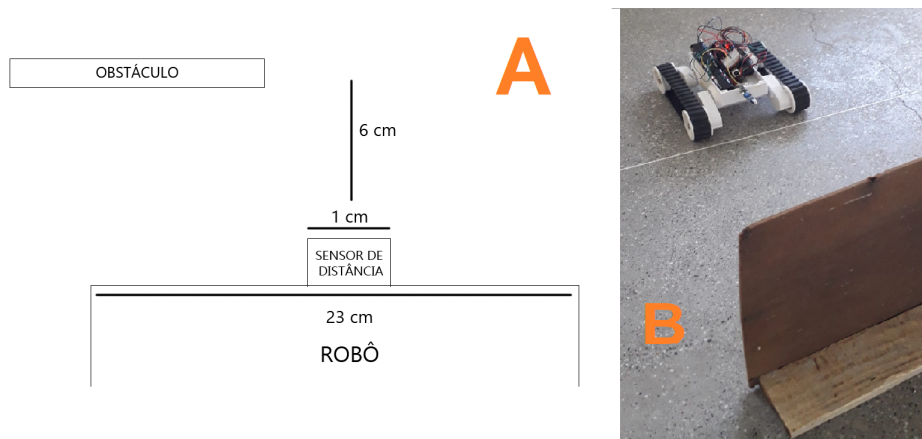


Figura 9: Problema de identificação de objeto pelo sensor de distância (Figura A). O robô diante de um obstáculo (Figura B). Fonte: Própria, 2021

4.2 Construção e prototipagem dos blocos de comandos

No esboço inicial dos blocos, foram utilizadas as dimensões de comprimento, largura e altura de 4 centímetros (Figura 10 A), com uma abertura em uma das faces de 0,5 centímetros, enquanto que a tag RFID apresentava um comprimento de 3 centímetros e uma altura de 4 centímetros. No esboço também é apresentada a face superior do cubo, que possui um símbolo e o nome do símbolo em questão.

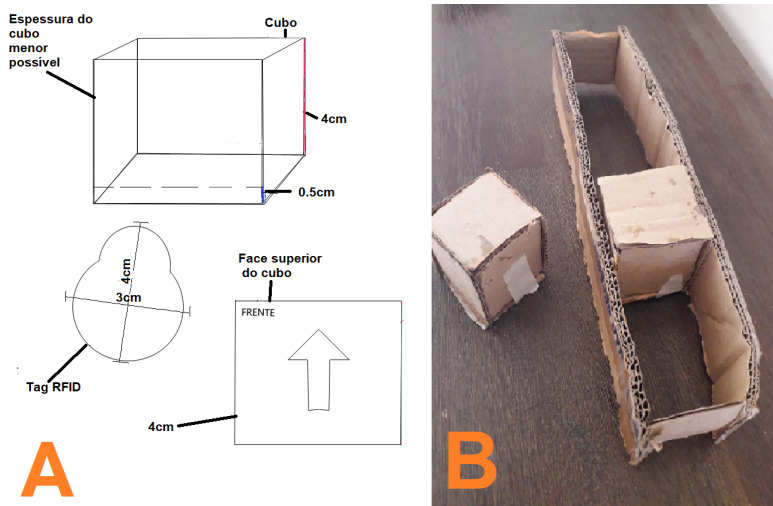


Figura 10: Esboço inicial com especificações do bloco, da tag RFID e da face superior do bloco (Figura A). Protótipo de blocos e suporte de enfileiramento dos blocos de papelão (Figura B). Fonte: Própria, 2021

O resultado da prototipação de baixa fidelidade utilizando o material de papelão é apresentado na Figura 10 (B), que traz imagens dos blocos e do suporte utilizado para o enfileiramento dos mesmos. Este primeiro processo de prototipação foi necessário para validar a capacidade de leitura do leitor RFID mediante um enfileiramento de blocos. As tags foram presas na parte inferior do cubo para que pudessem ser lidas.

Nos testes realizados com os protótipos de acrílico, os blocos apresentaram a capacidade de permitir a permeabilidade do sinal emitido pela *tag*, que deveria ser captado pelo leitor RFID, além de ser um material não flexível, diferente do papelão que, por exemplo, poderia ser facilmente amassado, cortado ou danificado. Existiram dois protótipos de cubo confeccionados, o primeiro deles (Figura 12 (A)) com dimensões de comprimento, largura e altura de 4 centímetros, com uma abertura lateral de 0,5 centímetros para a entrada da *tag*. Neste primeiro protótipo, foi testada a permeabilidade do sinal de rádio frequência pelo material, o que funcionou adequadamente. Entretanto, as dimensões do cubo se mostraram inadequadas para um encaixe harmonioso da *tag* dentro do bloco, além de o bloco não se manter fixo no suporte construído para o enfileiramento, implicando em uma remodelagem.

A Figura 11 apresenta o modelo final de blocos desenvolvido, no qual foram inseridas extensões laterais na face superior dos blocos, para que pudessem se sustentar no suporte de enfileiramento, além das próprias dimensões do bloco que também foram reajustadas. O resultado da confecção dos dois protótipos é apresentado na Figura 12, sendo a Figura A a primeira versão, e a Figura B a segunda versão.

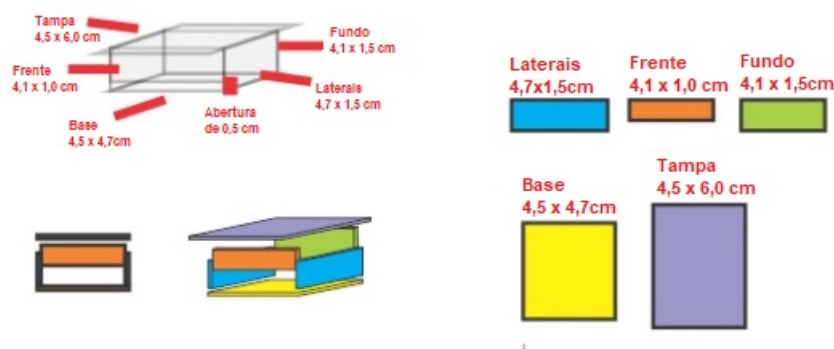


Figura 11: Modelagem final dos blocos de acrílico e suas especificações. Fonte: Própria, 2021

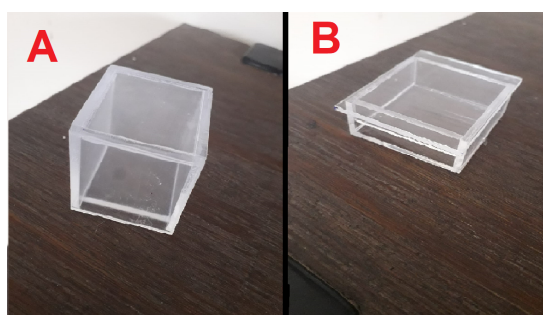


Figura 12: Blocos utilizados como protótipo do projeto. Figura A, primeira versão e Figura B, a segunda versão. Fonte: Própria, 2021

No total, foram construídos 5 blocos frente; 5 blocos trás; 4 blocos direita; 4 blocos esquerda; 2 blocos *if*; 2 blocos *loop*; 1 bloco *stop*; os símbolos para cada bloco; um suporte de acrílico para o enfileiramento dos blocos e um suporte de acrílico para sustentar a fila de blocos acima do leitor RFID. Caso todos os 22 blocos de comandos (exceto o bloco de parada) sejam

utilizados pelo usuário e ele deseje um algoritmo maior, basta que, logo após a leitura deste algoritmo (todos os blocos) pelo leitor RFID, o usuário monte um algoritmo de continuação, agora utilizando o bloco de parada no final, para garantir o início da movimentação do robô.

5 Resultados e discussões

Nesta seção serão descritos os resultados obtidos por meio de experimento de validação realizado com os participantes. A validação envolveu aspectos de usabilidade da interface, assim como a verificação do desenvolvimento do PC dos estudantes.

5.1 Validação e discussão dos resultados quantitativos

Os participantes da pesquisa não possuíam conhecimentos anteriores relacionados aos conceitos de algoritmo e programação. Ainda assim, 60% dos participantes declararam ter vivenciado algum tipo de contato com a robótica e todos da turma participante declararam sua simpatia com o convívio e o uso de tecnologias, sendo que 60% possuíam um alto grau de simpatia e 40% um grau menor. Todos da turma faziam uso de internet diariamente: 40% por pelo menos três horas por dia; 30%, por cinco horas ou mais; 20%, durante uma hora e 10% durante duas horas por dia. Dos 10 alunos participantes da pesquisa, três tinham oito anos de idade, e sete, nove anos.

No pré-teste do instrumento do PC, foi encontrada uma média de 2,10 acertos por participante em cinco questões, com um desvio padrão de 1,59 e mediana de 2,00; enquanto que no pós-teste foi atingida uma média de 3,00 acertos por participante com um desvio padrão de 1,24, respondendo as mesmas cinco questões. O *boxplot* da Figura 13 demonstra a distribuição dos dados dos pré e pós-testes.

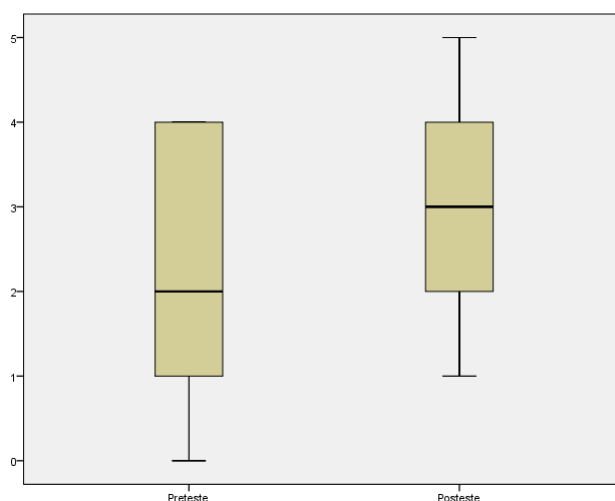


Figura 13: *Boxplot* da distribuição dos dados a respeito da avaliação do pensamento computacional. Fonte: Própria, 2021

Para a comparação entre as médias obtidas no pré-teste e no pós-teste (entre as cinco questões), foi utilizado o teste t de *student*. Contudo, antes desta medida, foi realizado o teste de

Kolmogorov-Smirnov para verificar a normalidade dos dados. Tanto no pré-teste quanto no pós-teste, foi obtido o p-valor de 0,200 indicando que as amostras seguem uma distribuição normal. Caso o p-valor encontrado fosse inferior a 0,05, a distribuição das amostras estaria diferindo significativamente de uma distribuição normal.

Na realização do teste t de *student*, obteve-se uma diferença média entre as duas medidas (média do pré-teste e média do pós-teste) de 0,900, com um desvio padrão de 1,44 e p-valor de 0,081, com um grau de confiança de 95%, o que indica que não houve uma diferença significativa entre as médias obtidas no pré-teste e no pós-teste. Para que as médias comparadas na realização do teste t de *student* fossem consideradas distintas, seria preciso que o p-valor encontrado fosse menor do que 0,05. No sentido de confirmar a diferença entre as médias, calculou-se o tamanho de efeito entre o pré e o pós-teste, ratificando a proximidade e a pequena diferença significativa. O valor de 0,1262, menor que 0,2⁵, significa um pequeno tamanho de efeito, representando que a chance do efeito acontecer do pré para o pós teste é pequena. Ainda assim, é possível perceber uma menor dispersão nos dados do pós-teste, o que significa maior concentração de acertos, além do aumento do número de acertos (Figura 13). É importante lembrar que, segundo Lipsey et al. (2012), em estudos de intervenção em educação, é raro haver tamanho de efeito acima dos 0,30, pela própria característica dessa área de intervenção.

Analisando o resultado individual dos participantes da pesquisa, comparando o pré-teste com o pós-teste (Figura 14), pode-se verificar que a maior parte dos participantes apresentou aumento na quantidade de acertos, neste caso, seis participantes, contudo, dois deles regrediram no número de acertos durante o pós-teste e dois mantiveram-se com a mesma quantidade de acertos. Adicionalmente, é possível verificar na figura a peculiaridade dos participantes seis e sete, os quais não acertaram questão alguma no pré-teste (quantidade de acertos zero), e conseguiram obter acertos no pós-teste.

Na avaliação da usabilidade da interface, as questões Q2, Q4 e Q8 (vide Quadro 1) constataram que 70%, 40% e 90% dos participantes, respectivamente, discordaram das afirmações "Achei o sistema desnecessariamente complexo", "Eu precisaria de suporte técnico para usar esse sistema", "Achei o sistema estranho de usar", o que significa positividade na avaliação destas características e a baixa complexidade de uso da interface, totalizando cerca de 67% de aprovação nas três questões. Em contrapartida, 30%, 50% e 10% dos participantes concordaram com as afirmações apresentadas, respectivamente, representando uma média de 30% de reprovação. A Q4 foi a única questão das três em que houve um percentual de 10% de indiferença (50% aprovação + 40% reprovação + 10% indiferença), representando aproximadamente 3% do total. A distribuição das respostas pode ser visualizada na Figura 15.

Na avaliação das perguntas Q1, Q3 e Q9 (vide Quadro 1), que analisam, respectivamente, o interesse de reutilizá-la em outros momentos, a facilidade de uso da ferramenta, e a confiança do participante no seu uso, observou-se que 80%, 70%, e 100% dos participantes concordaram, respectivamente, com as afirmações das questões citadas, totalizando 83% de respostas positivas; enquanto que 10%, 20%, e 0% discordaram, representando 10% de respostas negativas. Na pergunta Q10, é possível constatar que a maioria dos participantes (80%) considerou como grande a extensão dos conteúdos apresentados em sala a respeito de conceitos de algoritmos e programa-

⁵Valores superiores ou iguais a 0,8 representam tamanho de efeito grande; entre 0,8 a 0,2 são considerados médios e, inferiores a 0,2, pequenos (Cohen, 1988)

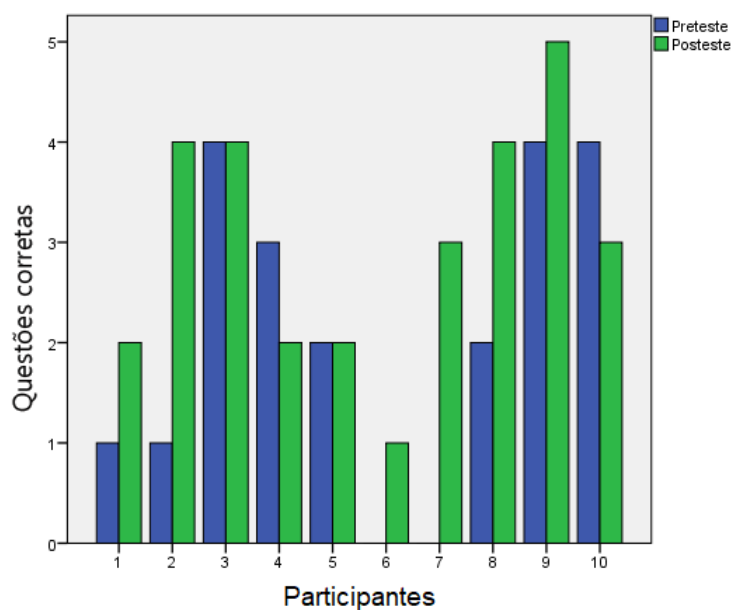


Figura 14: Quantidade de acertos por estudante no pré-teste e no pós-teste. Fonte: Própria, 2021

ção, conteúdos com os quais nunca tinham tido contato anterior, o que demonstra a necessidade de pesquisas futuras adotarem estratégias para dividir os conteúdos em um número maior de dias, com conteúdos mais espaçados. A Figura 16 apresenta o resultado das questões Q1, Q3, Q9 e Q10.

5.2 Validação e discussão dos resultados qualitativos

Nesta seção, serão apresentados os experimentos realizados, com a perspectiva de elucidar as falas dos participantes, seus interesses, dificuldades e facilidades de uso, além dos *feedbacks* gestuais e verbais dos estudantes ao longo de todo o processo. Também será possível perceber a relação das atividades com os níveis cognitivos de aprendizagem propostos pela Taxonomia revisada de Bloom e os conceitos do PC, tratados na Fundamentação Teórica deste trabalho. Para tanto, utilizar-se-á o roteiro das atividades executado durante a validação como sequência para este descritivo: (1) explicação breve sobre o objetivo do experimento; (2) explicação do conceito de algoritmo e a sua relação com a programação e a robótica; (3) avaliação prévia do PC; (4) atividade de computação desplugada; (5) utilização da interface tangível e (6) avaliação do PC posterior às atividades realizadas e a validação da usabilidade da interface tangível.

A participação na pesquisa foi de consenso unânime dos estudantes presentes em sala de aula. No momento inicial da explicação do objetivo do experimento, os participantes se mostraram dispostos a participar das atividades de computação desplugada, dos questionários de validação e do uso da interface tangível.

Após a primeira etapa, deu-se início à explicação do conceito de algoritmo e sua relação com a programação e a robótica. Um paralelo com situações do dia-a-dia foi feito de forma a conduzir os participantes a reflexões da abrangência robótica e computacional em alguns setores

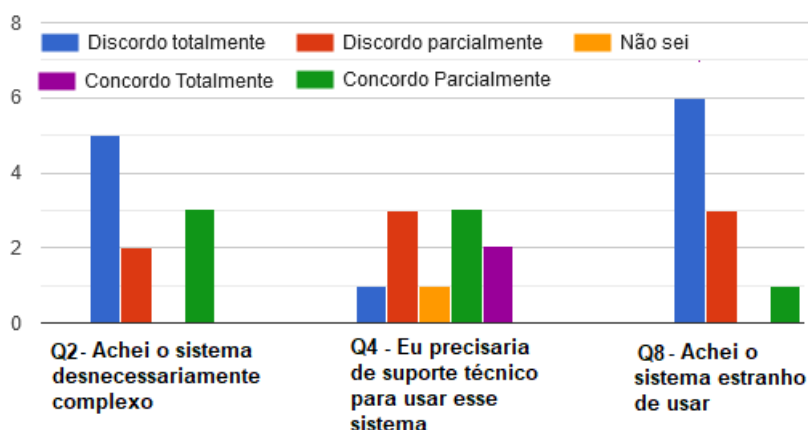


Figura 15: Resultado do questionário de usabilidade das questões Q2, Q4 e Q8. Fonte: Própria, 2021

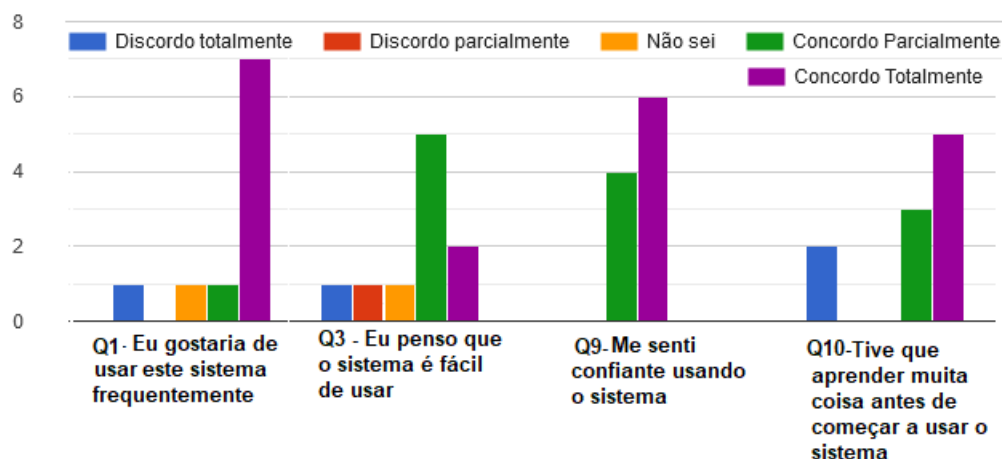


Figura 16: Resultado do questionário de usabilidade das questões Q1, Q3, Q9 e Q10. Fonte: Própria, 2021

da sociedade como nas missões espaciais, nas fábricas e, até mesmo, em algumas lojas. Os estudantes ligeiramente passaram a sugerir novos exemplos de utilização da robótica de acordo com a sua própria visão de mundo, como brinquedos e jogos de celulares.

Neste mesmo momento, foi introduzido o conceito de algoritmo, característico do PC. A princípio, o nome gerou estranheza na sala de aula, pois nenhum voluntário possuía qualquer ideia do que poderia ser um algoritmo. Didaticamente, uma receita de bolo foi utilizada para exemplificar o funcionamento de um algoritmo e, em seguida, uma atividade rápida para montar um algoritmo de trocar uma lâmpada foi realizada com a turma. Além de introduzir o conceito de **algoritmo**, foram trabalhados outros conceitos do PC, tais como a **abstração**, considerando um robô hipotético que trocava a lâmpada, e o de **generalização**, garantindo que qualquer robô com aquele algoritmo fosse capaz de atingir o objetivo proposto (trocar lâmpada).

Nesta etapa, também foram apresentados os blocos que seriam utilizados para a programação do robô. Os comandos que se mostraram com maior dificuldade de associação pelos estudantes foram os blocos de comando *if* e *loop*. Para esta situação, foi demonstrado que é possível fazer uma pergunta ao robô para a qual o mesmo pode responder 'sim' ou 'não' (condicional), assim como pode-se pedir ao robô que repita uma ação várias vezes.

No processo de **avaliação prévia do PC**, ficou evidente a discrepância de tempo na resolução das questões, alguns resolveram mais rapidamente que outros.

Na **atividade de computação desplugada**, os estudantes foram distribuídos em dois grupos, um de meninos e outro de meninas, para que nestes grupos pudessem resolver o problema proposto. A reação imediata dos participantes foi a de se disponibilizar prontamente para atuar como o robô (escolha aleatória e espontânea), indicando o engajamento promovido pelo estado emocional dos alunos através do envolvimento com a robótica educacional. Para que houvesse oportunidade de que todos participassem da atividade, cada membro da equipe deveria escolher um bloco para ser colocado no suporte de acrílico. Esta condição pode ter sido um dos motivos desencadeadores da ocorrência de algoritmos que se iniciavam, em sua maioria, com comandos corretos e finalizavam com algumas incoerências. Em contrapartida, esta estratégia permitiu a construção de uma solução colaborativa, com ideias e sugestões de todos os participantes, e maior integração entre eles.

No **processo de utilização da interface tangível**, os estudantes foram submetidos a três desafios, que seriam também realizados pelos dois grupos. Estes grupos manipularam a interface separadamente, para que pudessem construir soluções próprias, sem a influência da solução anterior do outro grupo.

O primeiro desafio foi utilizado para que os participantes pudessem verificar a distância percorrida pelo robô, utilizando somente um único comando de "ir para frente". Como as etapas já descritas tinham acontecido no dia anterior, antes de iniciar o processo de construção do algoritmo, foram apresentados novamente aos estudantes os blocos de comandos, questionando-lhes qual a função de cada um deles (Figura 17 A). Eles conseguiram relembrar as discussões prévias e associar cada comando à sua função, como a fala de um deles que representa o bloco *if*: "este é o bloco para fazer uma pergunta ao robô". Os blocos de movimentação foram aqueles sem dificuldade alguma de rememorar e o bloco de *loop* foi muito confundido com um comando de virar o robô. Acredita-se que esta associação pode ter ocorrido devido à indicação do círculo na face superior do bloco. Ainda neste momento, foi realizada uma movimentação exemplo com o robô, para que os estudantes criassem uma noção da distância que o robô era capaz de se movimentar com os comandos de direção.

Com base na pirâmide da taxonomia revisada de Bloom, neste momento foram observados traços de um processo de **recordação** e **entendimento**, demonstrados pelas reações de inferir, comparar e classificar os blocos de comando (entender), e as de rememorar conceitos aprendidos (recordar).

Na imagem do primeiro desafio (Figura 18 A), o objetivo era alcançar o ponto marcado com uma fita vermelha. Como o robô foi posto a 20 centímetros do ponto demarcado, os participantes tiveram que montar um algoritmo apenas com um único passo de "ir para frente".

O segundo desafio (Figura 18 B) consistiu na locomoção do robô também para frente, po-

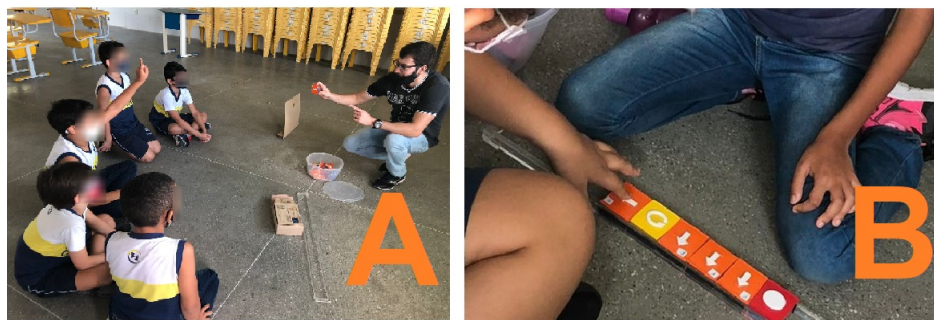


Figura 17: Processo de recordação e entendimento realizado momento antes da atividade de uso do robô (Figura A). Solução muito distante do esperado (Figura B). Fonte: Própria, 2021

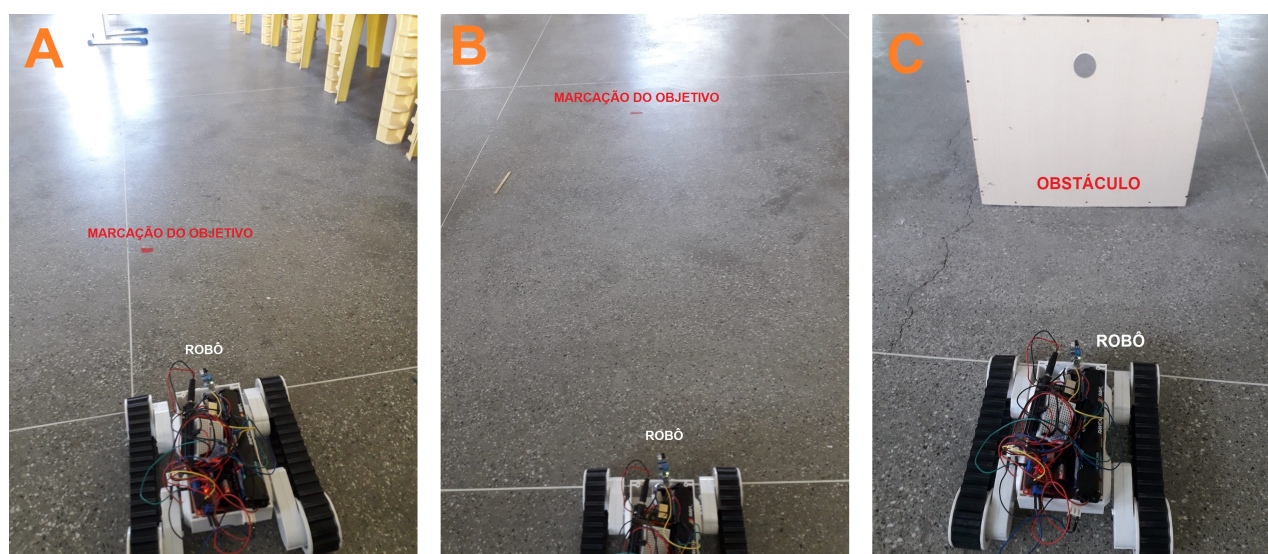


Figura 18: Desafio 1 realizado com os voluntários (Figura A). Desafio 2 realizado com os voluntários (Figura B). Desafio 3 realizado com os voluntários (Figura C). Fonte: Própria, 2021

rém com uma distância aproximada de 60 centímetros (três comandos para frente). O comportamento observado foi justamente a **aplicação** da locomoção previamente conhecida (desafio 1), caracterizando-se como o processo de **aplicar** da pirâmide da taxonomia revisada de Bloom. Além disso, com a variação da distância do segundo para o primeiro desafio, houve também a recordação do passo previamente utilizado.

Durante o processo de construção dos algoritmos, houve inicialmente pouca intervenção do pesquisador com a solução proposta pelos estudantes. Os dois grupos discutiram bastante a solução, e alguns alunos se mostraram mais seguros no uso dos blocos do que outros e, com isso, com maior grau de intervenção no grupo. A intervenção e a mediação do pesquisador aconteceram posterior à elaboração do algoritmo pelo grupo, para que eles pudessem experimentar os níveis cognitivos de **analisar** e **avaliar** presentes na pirâmide revisada da taxonomia de Bloom, e, com isso, desenvolver um pensamento crítico a respeito da solução. Esta mediação posterior foi bastante útil, sobretudo nas soluções que não levavam o robô ao ponto desejado, para que pudessem experimentar soluções alternativas (novas tentativas). Nos momentos nos quais a solução era

muito distante do esperado, como no algoritmo da Figura 17 (B), no qual esperava-se que o robô se locomovesse três passos para frente (desafio da Figura 18 B), algumas dicas mais diretas foram apresentadas aos participantes, para evitar uma frustração muito grande.

Não foi realizado um comparativo direto entre os grupos de meninas e meninos no decorrer das observações. Entretanto, foi possível perceber que o tempo de resolução dos problemas foi muito semelhante entre eles.

No terceiro desafio, o robô deveria ser programado para, ao identificar um obstáculo, recuar novamente para a posição inicial, fazendo com que o estudante fizesse uso do bloco de comando 'if' (Figura 18 C). Ainda que este desafio envolvesse um grau maior de complexidade, os participantes foram imbuídos de um maior grau de liberdade para tomar suas decisões na construção do algoritmo, na perspectiva de que atingissem o nível cognitivo **criar** da Taxonomia Revisada de Bloom. Neste momento, os estudantes recordaram da função do bloco condicional 'if', realizando comentários como: "temos que perguntar ao robô se existe um obstáculo". Também foi observado um grau maior de planejamento nesta etapa, visto a ocorrência das experiências adquiridas nos desafios anteriores.

Examinando o comportamento dos participantes no decorrer do processo de resolução dos três desafios, os conceitos do PC como **abstração**; **algoritmo**; **generalização**; **análise lógica** e **avaliação** foram constatados pela observação de alguns comportamentos.

A capacidade de isolar o conceito de um determinado bloco de comando, levando-o para outras situações do cotidiano como demonstram as falas anteriores, reforça a hipótese de que os participantes atingiram um determinado nível de **abstração**. Além da explicação anterior do conceito de **algoritmo**, no processo de resolução dos desafios, os participantes tiveram a capacidade de definir na prática o que o robô deveria fazer (passos algorítmicos). Em relação ao conceito de **decomposição**, o algoritmo de solução da Figura 18 B é uma composição de três passos do algoritmo de solução da Figura 18 A, representando a possibilidade de o estudante subdividir um percurso maior (algoritmo mais complexo) em percursos menores (algoritmos menores intermediários), ainda que de maneira intuitiva.

Foi também observado que a padronização das cores de acordo com as funções específicas de cada bloco de comando foi impactante para o processo de **generalização**, expressando capacidades de resolução de outros tipos de desafios ou de criação de diferentes algoritmos para posicionar o robô no lugar demarcado. O conceito da **análise lógica** foi atribuído, além da própria construção de todo o algoritmo, à correta utilização do bloco condicional 'if' que gera sempre um valor lógico (verdadeiro ou falso) que direciona o comportamento do robô. Neste sentido, os participantes foram capazes de identificar se o robô havia executado os comandos dentro do bloco 'if'. Por fim, o conceito de **avaliação** das soluções foi estimulado no decorrer dos diferentes desafios, por meio de dicas pontuais, ou, até mesmo, através do processo de tentativa e erro, permitindo-lhes desenvolver a capacidade de verificar blocos mal posicionados ou que levavam a uma posição divergente da posição do objetivo final.

Na etapa de pós-tesde de validação do PC, de forma semelhante ao pré-teste, observou-se discrepância de tempo entre os participantes para a resposta às questões. Um paralelo pode ser feito com os estudos de Vieira et al. (2015), os quais identificaram a variação de níveis de conhecimento entre estudantes, porém provenientes de uma distinta formação educacional, o que não ocorre necessariamente neste caso, contudo, ainda assim, é um fator observado durante as

atividades de compreensão e construção das soluções algorítmicas.

Para as três novas questões inseridas no pós-teste, utilizadas para verificar como os participantes lidariam com problemas inusitados e mais complexos que os anteriores, os resultados mostraram que dos 10 participantes, 50% acertaram a primeira questão, 30% a segunda e 40% a terceira.

Pode-se inferir, pelos resultados, que a complexidade e a característica de serem questões diferentes das anteriores tenham impactado na quantidade menor de acertos. Além disso, observou-se que questões que envolviam a identificação de um erro no algoritmo também apresentaram maior taxa de erro, isso foi constatado tanto em questões de identificação de erro no pré-teste quanto no pós-teste. Uma possível estratégia com a utilização desta TUI é a proposição de algoritmos/exercícios de movimentação do robô com alguns erros, para que as crianças percebam passos que resultem em soluções incompletas ou equivocadas, simulando um processo de depuração (*debug*).

Apesar da dificuldade na resolução de algumas das questões do instrumento do PC, o experimento e os achados demonstraram que a interface pode contribuir de forma lúdica e tangível para a apreensão dos conhecimentos lógicos computacionais. Ainda em momentos posteriores à experimentação, o processo surtiu impactos positivos no ambiente escolar com fomento, pelos participantes e funcionários da escola, da inclusão da disciplina de robótica educacional no currículo da instituição.

6 Considerações finais

Considerando a capacidade e a abrangência do pensamento computacional, nas mais diversas áreas do conhecimento e os benefícios futuros de sua utilização para os alunos, a dinâmica da utilização de uma interface tangível para a construção de algoritmos na programação de robôs pode ser inserida com evidentes aspectos práticos, tais como utilização de sinal sonoro, do tato, de conexão sem fio, e também de *software* imbutido. Outras interfaces tangíveis também podem se utilizar de aspectos voltados à acessibilidade e ao uso de aplicativos de dispositivos móveis; nos trabalhos correlatos foi observado que as características podem variar de acordo com o objetivo da interface tangível.

Uma interface tangível abstrai o procedimento de construção lógica de um algoritmo ou resolução de um problema, fazendo com que esta capacidade seja exercitada por meio de sua utilização. No contexto dos perfis dos cursos relacionados à área de computação, a utilização de uma interface tangível com estes mesmos aspectos apresentados na educação básica pode prover habilidades preliminares, necessárias à introdução de conceitos de algoritmos e linguagens de programação em cursos superiores.

Um aspecto evidente durante a realização deste trabalho foi o entusiasmo dos participantes durante a pesquisa no uso da robótica e no aprendizado dos conceitos de algoritmo e programação. Apesar da validação quantitativa não apresentar uma diferença significativa nas médias de acertos (pré-teste e pós-teste), a interface tangível desenvolvida ainda mostra-se capaz de trabalhar os conceitos do pensamento computacional e níveis cognitivos da taxonomia de Bloom. O mecanismo de funcionamento e de utilização da interface também apresentou aspectos positivos segundo a

avaliação feita pelos participantes, porém deve-se considerar a necessidade de uma introdução mais alongada (em carga horária) dos conceitos de algoritmo e programação e do manuseio da interface tangível para que os usuários possam interagir previamente com o artefato.

No processo de avaliação qualitativa, também foram evidenciados os diferentes níveis de conhecimento entre os participantes. Os alunos com tempos de resolução das questões no pré e no pós-teste discrepantes do tempo médio geral da turma também apresentaram necessidade maior de auxílio para o entendimento das questões de pensamento computacional. Pensar, portanto, na inserção da robótica e do pensamento computacional nas escolas brasileiras, no ensino fundamental, pode ser uma alternativa para o desenvolvimento desta habilidade.

Este estudo apresentou, portanto, a possibilidade de utilização da robótica e do uso de algoritmos como forma de estímulo de aprendizagem para esta área, assim como a potencialidade do uso das interfaces tangíveis através da robótica, no contexto do currículo escolar brasileiro, para o desenvolvimento do pensamento computacional. Vislumbra-se também a utilização das interfaces tangíveis para o ensino de outras áreas do conhecimento básico, tais como matemática, geometria e outras, em função de sua capacidade de modulação.

6.1 Limitações externas e internas à pesquisa e ao desenvolvimento

Limitações externas como a quantidade da amostra devido ao período de pandemia (redução de número de alunos no ambiente escolar), para evitar o contágio pelo vírus do COVID-19, impediu uma análise do impacto com uma quantidade maior de participantes. Houve também limitação no tempo disponibilizado pela instituição de ensino para a aplicação dos experimentos, exigindo adaptações/redução na quantidade de questões do instrumento de avaliação do pensamento computacional.

Limitações internas, como problemas inesperados no uso de um novo bloco condicional (*else*) também impediram que esta funcionalidade fosse avaliada na realização deste trabalho. A retirada de três questões do instrumento original de avaliação da usabilidade (Q5, Q6 e Q7 vide Quadro 1) também se constitui como uma limitação interna, na medida em que não foi possível a atribuição de uma nota classificatória do nível da usabilidade da interface segundo os padrões apresentados pelo instrumento SUS.

6.2 Trabalhos Futuros

Um possível trabalho futuro para esta pesquisa é a realização de testes com uma amostra maior de participantes. Acompanhando o crescimento da amostra, espera-se um aumento proporcional do tempo para a realização das validações, assim como uma maior variedade de desafios com o robô e consequente aprofundamento dos conceitos algorítmicos.

Alguns ajustes na ferramenta também precisariam ser realizados, tais como a substituição do compartimento de papelão onde está localizado o leitor RFID (Figura 7 A) por um material de acrílico; a organização e a diminuição dos componentes localizados no *chassi* do robô; a organização dos fios necessários às conexões, além de novos comandos como a condicional composta (*else*), *loops* condicionais, funções, dentre outros.

A avaliação da interface na perspectiva docente também é um possível trabalho futuro. Esse

incremento permitirá, inclusive, ampliar as discussões a respeito do pensamento computacional, relacionando-a com a proposição da BNCC (Brasil, 2022).

Referências

- Armstrong, P. (2016). Bloom's taxonomy. *Vanderbilt University Center for Teaching*. [[GS Search](#)]
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community? *Acm Inroads*, 2(1), 48–54. [[GS Search](#)]
- Barros, R. P., Torres, V. P., Burlamaqui, A. M. F., & Natal, R. (2014). Cardbot: Tecnologias assistivas para imersão de deficientes visuais na robótica educacional. In *Workshop de robótica educacional* (Vol. 5, pp. 11–16). [[GS Search](#)]
- Berry, M. (2013). *Computing in the national curriculum: a guide for primary teacher*. newnorth print, ltd. bedford. [[GS Search](#)]
- Blikstein, P. (2008). O pensamento computacional e a reinvenção do computador na educação. [[link completo](#)]
- Bloom, B. S., Krathwohl, D. R., & Masia, B. B. (1984). Bloom taxonomy of educational objectives. In *Allyn and bacon*. Pearson Education. [[GS Search](#)]
- Brackmann, C. P. (2017). Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica. [[GS Search](#)]
- Brasil (2022). Normas sobre computação na educação básica – complemento à base nacional comum curricular (bncc). Ministério da Educação. Conselho Nacional de Educação. Parecer Homologado. Despacho do Ministro, publicado no D.O.U. de 03/10/2022, Seção 1, Pág. 55, 2022c. [[link completo](#)]
- Brooke, J., et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4–7. [[GS Search](#)]
- Cohen, J. (1988). *Statistical power analysis for the behavioural sciences* (2^{eu}). Hillsdale, NJ: Lawrence Erlbaum Associates. [[GS Search](#)]
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking-a guide for teachers*. [[GS Search](#)]
- de Araújo, A. L. S. O., Scaico, P. D., de Paiva, L. F., de Moraes Rabêlo, H., de Luna Santos, L., Pessoa, F. I. R., ... dos Santos Costa, L. (2013). Aplicação da taxonomia de bloom no ensino de programação com scratch. In *Anais do workshop de informática na escola* (Vol. 1, p. 31). [[GS Search](#)]
- de Jesus, E. A., & Raabe, A. L. A. (2009). Interpretações da taxonomia de bloom no contexto da programação introdutória. In *Brazilian symposium on computers in education (simpósio brasileiro de informática na educação-sbie)* (Vol. 1). [[GS Search](#)]
- dos Santos Soares, M. (2004). Metodologias ágeis extreme programming e scrum para o desenvolvimento de software. *Revista Eletrônica de Sistemas de Informação*, 3(1). [[GS Search](#)]
- Falcão, T. P., & Gomes, A. S. (2007). Interfaces tangíveis para a educação. In *Brazilian sym-*

- posium on computers in education (simpósio brasileiro de informática na educação-sbie)* (Vol. 1, pp. 579–589). [[GS Search](#)]
- Farr, W., Yuill, N., & Raffle, H. (2010). Social benefits of a tangible user interface for children with autistic spectrum conditions. *Autism*, 14(3), 237–252. [[GS Search](#)]
- Ferri, J., & dos Santos Rosa, S. (2016). Como o ensino de programação de computadores pode contribuir com a construção de conhecimento na educação básica uma revisão sistemática da literatura. *RENOTE-Revista Novas Tecnologias na Educação*, 14(2). [[GS Search](#)]
- Fishkin, K. P. (2004). A taxonomy for and analysis of tangible interfaces. *Personal and Ubiquitous computing*, 8(5), 347–358. [[GS Search](#)]
- Ishii, H. (2008). The tangible user interface and its evolution. *Communications of the ACM*, 51(6), 32–36. [[GS Search](#)]
- Júnior, E. d. S. S., Pinto, S. C. C., & Braz, R. M. M. (2018). A computação embarcada, a plataforma arduino e a internet das coisas como tecnologia assistiva na construção de mapas táteis para os alunos com deficiência visual no processo de ensino e aprendizagem. In *Anais dos workshops do congresso brasileiro de informática na educação* (Vol. 7, p. 53). [[GS Search](#)]
- Takehashi, S., Motoyoshi, T., Koyanagi, K., Ohshima, T., & Kawakami, H. (2013). P-cube: Block type programming tool for visual impairments. In *2013 conference on technologies and applications of artificial intelligence* (pp. 294–299). [[GS Search](#)]
- Lewis, J. R. (2018). The system usability scale: past, present, and future. *International Journal of Human-Computer Interaction*, 34(7), 577–590. [[GS Search](#)]
- Lipsey, M. W., Puzio, K., Yun, C., Hebert, M. A., Steinka-Fry, K., Cole, M. W., . . . Busick, M. D. (2012). Translating the statistical representation of the effects of education interventions into more readily interpretable forms. *National Center for Special Education Research*. [[GS Search](#)]
- Lozano, M. D., Penichet, V. M., Leporini, B., & Fernando, A. (2018). Tangible user interfaces to ease the learning process of visually-impaired children. In *Proceedings of the 32nd international bcs human computer interaction conference 32* (pp. 1–5). [[GS Search](#)]
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in k-9 education. In *Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference* (pp. 1–29). [[GS Search](#)]
- Mohamed, K., Dorgham, Y., & Sharaf, N. (2021). Kodockly: Using a tangible robotic kit for teaching programming. In *Csedu (1)* (pp. 137–147). [[link completo](#)]
- Molina-García, J. C., Rodríguez-Elías, O. M., Glasserman-Morales, L. D., & Rodríguez-Pérez, J. M. (2016). Designing a strategy of programming learning for kids through the use of the "micompu. mx" federal program: A pilot study. In *2016 4th international conference in software engineering research and innovation (conisoft)* (pp. 104–109). [[GS Search](#)]
- Nunes, A. L., Radicchi, A. O., & Botega, L. C. (2011). Interfaces tangíveis: Conceitos, arquiteturas, ferramentas e aplicações. *Realidade Virtual e Aumentada: aplicações e tendências*, 26–44. [[GS Search](#)]
- Panaggio, B. Z., Carbajal, M. L., & Baranauskas, M. C. C. (2019). Programação tangível no mundo físico: Taprec+ sphero. *Revista Brasileira de Informática na Educação*, 27(03), 32–51. [[GS Search](#)]
- Posada, J. E. G., Hayashi, E. C., & Baranauskas, M. C. C. (2014). On feelings of comfort, motiva-

- tion and joy that gui and tui evoke. In *International conference of design, user experience, and usability* (pp. 273–284). [[GS Search](#)]
- Román-Gonzalez, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2015). Test de pensamiento computacional: diseño y psicometría general. In *Iii congreso internacional sobre aprendizaje, innovación y competitividad (cinaic 2015)* (pp. 1–6). [[GS Search](#)]
- Scott, T. (2003). Bloom’s taxonomy applied to testing in computer science classes. *Journal of Computing Sciences in Colleges*, 19(1), 267–274. [[GS Search](#)]
- Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O’Grady-Cunniff, D., ... Verno, A. (2011). *Costa k–12 computer science standards: Revised 2011*. ACM. [[GS Search](#)]
- Shaer, O., & Jacob, R. J. (2009). A specification paradigm for the design and implementation of tangible user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 16(4), 1–39. [[GS Search](#)]
- Suzuki, H., & Kata, H. (1995). Interaction-level support for collaborative learning: Algorblock—an open programming language. [[GS Search](#)]
- Sylla, C., Branco, P., Coutinho, C., & Coquet, E. (2012). Tuis vs. guis: comparing the learning potential with preschoolers. *Personal and Ubiquitous Computing*, 16(4), 421–432. [[GS Search](#)]
- Thompson, E., Luxton-Reilly, A., Whalley, J. L., Hu, M., & Robbins, P. (2008). Bloom’s taxonomy for cs assessment. In *Proceedings of the tenth conference on australasian computing education-volume 78* (pp. 155–161). [[GS Search](#)]
- Viana, C., & Raabe, A. (2018). Interface de programação tangível para produção de algoritmos sonoros. In *Anais dos workshops do congresso brasileiro de informática na educação* (Vol. 7, p. 125). [[GS Search](#)]
- Vieira, C. E. C., de Lima Junior, J. A. T., & de Paula Vieira, P. (2015). Dificuldades no processo de aprendizagem de algoritmos: uma análise dos resultados na disciplina de al1 do curso de sistemas de informação da faeterj—campus paracambi. *Cadernos UniFOA*, 10(27), 5–15. [[GS Search](#)]
- Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly*, xiii–xxiii. [[GS Search](#)]
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. [[GS Search](#)]