

Chapter

2

Computação Física Inclusiva com Micro:bit

Mariana Arfux Pereira Cavalcante de Castro, Bruna C. R. Cunha

Abstract

Assistive Technology (AT) is fundamental to inclusion, necessitating the development of new solutions. While physical AT systems offer benefits, options such as Arduino present barriers for beginners. In this context, the Micro:bit and the MakeCode platform serve as accessible tools for inclusive methodologies. This chapter presents a theoretical-practical approach grounded in Human-Computer Interaction (HCI) focused on accessibility, Universal Design for Learning (UDL), and the board's ecosystem. Furthermore, it provides a guide for creating prototypes with multimodal feedback, demonstrating ways to overcome accessibility barriers in educational settings and interface development. Notably, projects designed for autistic students are presented, fostering their active role in creating their own AT resources.

Resumo

A Tecnologia Assistiva (TA) é fundamental na inclusão, exigindo o desenvolvimento de novas soluções. Embora sistemas físicos de TA tragam benefícios, opções como o Arduino impõem barreiras a iniciantes. Nesse contexto, o Micro:bit e a plataforma MakeCode são ferramentas acessíveis para metodologias inclusivas. Este capítulo apresenta uma abordagem teórico-prática fundamentada em Interação Humano-Computador (IHC) voltada à acessibilidade, Design Universal para a Aprendizagem (DUA) e no ecossistema da placa. Ademais, traz um guia de criação de protótipos com feedbacks multimodais, demonstrando opções para superar barreiras de acessibilidade no contexto educacional e no desenvolvimento de interfaces. Em especial, apresentam-se projetos para estudantes autistas, promovendo o protagonismo na criação de seus próprios recursos de TA.

2.1. Introdução à Computação Física e Acessibilidade Digital

A democratização do acesso às tecnologias digitais tem sido pauta de políticas públicas, produções científicas e estratégias de inclusão social [1, 2, 3]. Nesse cenário, o Pensamento Computacional (PC) e o letramento digital consolidam-se como competências

fundamentais para o pleno exercício da cidadania na sociedade contemporânea [2]. Essa relevância é impulsionada por marcos regulatórios e diretrizes nacionais, como as atualizações da Base Nacional Comum Curricular - Computação (BNCC - Computação) (2022) e a instituição da Política Nacional de Educação Digital (PNED, Lei nº 14.533, que estabelecem a inclusão tecnológica como um direito estruturante) [1].

Entretanto, para auxiliar na compreensão dos conceitos abstratos sobre desenvolvimento de sistemas, tornam-se indispensáveis abordagens que mediem a interação entre o humano e o computador. A Computação Física surge nesse cenário como um paradigma que conecta o mundo digital (*software*) ao mundo físico (*hardware*) [7]. Quando aliada aos princípios da Acessibilidade Digital e às diretrizes da Lei Brasileira de Inclusão da Pessoa com Deficiência (LBI) (Estatuto da Pessoa com Deficiência, Lei nº 13.146/2015) [10], essa abordagem é uma possibilidade de transformar objetos concretos em interfaces tangíveis, permitindo a criação de tecnologias e produtos acessíveis.

Essa integração ampara-se no arcabouço normativo de acessibilidade, a exemplo do Cap. III, Art. 8º das Leis nº 10.048/2000 e nº 10.098/2000 [11], que fundamenta as barreiras na comunicação e na informação. Ao incorporar conceitos como desenho universal e o desenvolvimento de ajudas técnicas (Tecnologias Assistivas), garante-se a autonomia e a participação equitativa de pessoas com diferentes perfis sensoriais, motores ou cognitivos no manuseio de sistemas computacionais.

No âmbito de projeto e prototipagem, o ambiente do Micro:bit pode ser um exemplo de como ferramentas de prototipagem rápidas podem incorporar esses preceitos de *Design Inclusivo*. A plataforma disponibiliza recursos projetados para garantir a acessibilidade de usuários cegos ou com baixa visão, tais como diagramas táteis detalhados do hardware e descrições técnicas completas em texto para leitura de tela, além de possibilidades de alto contraste, leitura de tela e mudança de idiomas na plataforma de programação *MakeCode*. Tais estratégias demonstram como a integração entre hardware acessível, multimodal e boas práticas de Interação Humano-Computador (IHC) viabilizam a superação de barreiras comunicacionais, expandindo o desenvolvimento de interfaces e ecossistemas tecnológicos inclusivos.

2.1.1. Interfaces Tangíveis e Computação Física na Inclusão

As Interfaces Tangíveis de Usuário (TUI - *Tangible User Interfaces*) permitem que a manipulação de dados digitais ocorram através da interação direta com objetos físicos. Para estudantes com deficiência visual, neurodivergências ou limitações motoras, o suporte tátil e multissensorial proporcionado pelas TUIs oferecem um canal intuitivo de *feedback*, reduzindo a carga cognitiva e tornando conceitos abstratos de programação mais concretos e palpáveis [5].

A Computação Física atua, portanto, como uma ponte para a criação de Recursos de Tecnologia Assistiva (RTA). Segundo a legislação brasileira (Lei nº 13.146/2015) [10], a TA compreende todo o arsenal de recursos, equipamentos, dispositivos e serviços que objetivam promover a funcionalidade, a autonomia e a inclusão de pessoas com deficiência. Nesse sentido, a prototipagem eletrônica acessível possibilita que os próprios alunos, seus educadores e pesquisadores, construam soluções customizadas para superar barreiras de acessibilidade do cotidiano escolar [4].

2.1.2. O Panorama do Ensino de Tecnologia

No âmbito do ensino do Pensamento Computacional na educação básica, a plataforma Scratch é um ambiente muito utilizado devido a sua gratuidade e experiência visual. Considerando soluções de hardware, o Arduino é outra ferramenta muito adotada, neste caso, em diferentes níveis de ensino.

- **Scratch:** Plataforma de programação baseada em blocos visuais desenvolvida pelo MIT ¹. Suas principais contribuições incluem a eliminação do erro sintático típico das linguagens baseadas em texto e uma interface altamente lúdica. No entanto, em sua aplicação pura, o Scratch permanece restrito à tela do computador. Ademais, o formato exclusivo em blocos, ao mesmo tempo que simplifica conceitos de forma visual, inclui barreiras de interação para pessoas cegas ou com baixa visão.
- **Arduino:** Placa de prototipagem eletrônica de código aberto (*open-source*) focada na criação de circuitos customizáveis ². O Arduino democratizou a robótica educacional ao permitir o uso de uma vasta gama de sensores e atuadores externos. Contudo, seu uso apresenta desafios significativos de acessibilidade: a necessidade de montagem de circuitos complexos em matrizes de contato (*protoboards*), o manuseio de componentes minúsculos e o suporte nativo predominante a linguagens baseadas em texto (C/C++) impõem barreiras elevadas para iniciantes, estudantes com limitações motoras finas ou deficiência visual.

2.1.3. Justificativa e Vantagens do Ecossistema Micro:bit

Diante das limitações apresentadas pelo uso isolado do Scratch ou da complexidade do Arduino, o ecossistema Micro:bit posiciona-se como uma opção de Computação Física com sensores acoplados que facilitam a interação e introdução dos conceitos abstratos computacionais [12]. O Micro:bit integra o melhor de ambos os mundos: a simplicidade e acessibilidade das linguagens visuais com a tangibilidade dos dispositivos físicos, apresentando vantagens estratégicas:

- **Hardware:** Diferente do Arduino, que exige a fiação de periféricos externos para tarefas básicas, o Micro:bit V2 possui atuadores (matriz de LEDs, alto-falante) e sensores (acelerômetro, bússola, luz, temperatura, microfone e toque) integrados na própria placa. Isso elimina a dependência inicial de coordenação motora fina para montagem de circuitos em *protoboard*.
- **Conectividade e Feedback Multimodal Nativo:** A presença simultânea de saídas visuais (LEDs) e auditivas (alto-falante nativo) possibilita a entrega imediata de *feedbacks* multissensoriais, contornando limitações sensoriais específicas de cada estudante.
- **Ecossistema de Programação Híbrido:** A plataforma *MakeCode* oferece suporte nativo à transição entre blocos visuais, *JavaScript* e *Python*, incorporando desde o início diretrizes globais de acessibilidade para leitores de tela e navegação via teclado. Além de oferecer um aumento gradual sobre linguagens de programação.

¹<https://scratch.mit.edu>

²<https://www.arduino.cc>

2.1.4. Mapeamento de Barreiras e o *Design Universal para a Aprendizagem* (DUA)

Os ambientes tradicionais de desenvolvimento de software frequentemente falham ao desconsiderar a diversidade humana, impondo barreiras sensoriais (falta de leitores de tela ou alto contraste), motoras (dependência exclusiva do mouse) e cognitivas (sintaxe complexa e *feedback* abstrato). Para superar essas limitações, este trabalho fundamenta-se nos princípios do *Design Universal para a Aprendizagem* (DUA), uma abordagem pedagógica voltada ao desenvolvimento de ambientes de ensino acessíveis a todos os estudantes [13]. O DUA estrutura-se em três pilares fundamentais, os quais encontram no Micro:bit uma aplicação direta:

1. **Múltiplos Meios de Engajamento (O “Porquê” da Aprendizagem):** O Micro:bit permite a criação de projetos contextualizados e significativos, mantendo o interesse dos alunos através da manipulação de um objeto físico e reativo.
2. **Múltiplos Meios de Representação (O “O quê” da Aprendizagem):** A informação pode ser apresentada de formas variadas: texto na tela, ícones na matriz de LEDs, tons sonoros ou respostas tácteis, garantindo que estudantes com limitações sensoriais processem o conteúdo de forma equivalente.
3. **Múltiplos Meios de Ação e Expressão (O “Como” da Aprendizagem):** Os estudantes podem demonstrar o aprendizado programando via blocos, texto ou interagindo fisicamente com a placa (chacoalhando, tocando no logotipo, emitindo sons), respeitando os diferentes estilos e capacidades motoras.

Dessa forma, a integração da computação física pautada pelo DUA e suportada pelo ecossistema Micro:bit não apenas mitiga as barreiras dos ambientes tradicionais, mas consolida a tecnologia como um meio de emancipação e inclusão social.

2.2. O Ambiente do Micro:bit sob a Ótica da Inclusão

Nessa seção será apresentada a análise técnica da placa Micro:bit V2, destacando seus sensores e atuadores nativos como canais de *feedback* multimodal.

O desenvolvimento de práticas pedagógicas inclusivas no ensino de computação e robótica educacional exige ferramentas que contemplem a diversidade dos estudantes. Nesse cenário, a placa Micro:bit V2 é um recurso, que promove a acessibilidade física, sensorial e cognitiva por meio da integração entre hardware e o ambiente de programação.

2.2.1. Análise Técnica da Placa

A arquitetura do Micro:bit foi projetada para atuar como uma plataforma de interação multissensorial. A combinação de seus sensores nativos e atuadores integrados permite que a aprendizagem ocorra por diferentes canais perceptivos (visual, auditivo e tátil/proprioceptivo), o que é fundamental para estudantes com deficiências sensoriais ou neurodivergências.

Além disso, o Micro:bit em sua segunda versão (V2), possui adicionais, em relação à sua versão um, de sensores possibilitando mais interações com a placa³, como o

³<https://microbit.org/accessibility/full-text-tour/>

Logotipo Sensível ao Toque, LED do microfone, o próprio microfone, alto-falante, LED vermelho de energia, LED amarelo do USB e o botão de economia de bateria, todos esses elementos, foram adicionados na placa V2 que é o modelo utilizado e explicitado no capítulo. A Figura 2.1 ilustra a interface frontal do Micro:bit V2, seus componentes e respectivas descrições.

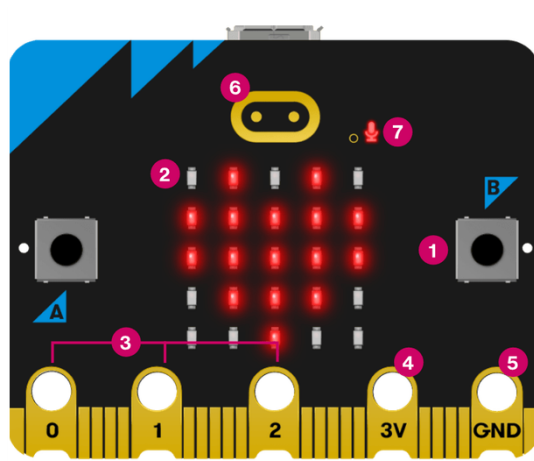


Figure 2.1. Frente da placa Micro:bit V2.

Componentes da parte frontal:

1. **Botões de Entrada (A e B):** Botões físicos de fácil acionamento que oferecem *feedback* tátil mecânico.
2. **Matriz de LEDs (5x5) e Sensor de Luz:** Atua como um *display* visual básico para matrizes, ícones, letras e animações, servindo simultaneamente para mensurar níveis de luminosidade como um sensor de luz.
3. **Pinos GPIO (General Purpose Input/Output):** Pinos de entrada e saída de fácil acesso para expansão de circuitos.
4. **Pino de Alimentação (3V):** Fornece tensão para componentes externos.
5. **Pino de Terra (GND):** Ponto de fechamento de circuito.
6. **Logotipo Sensível ao Toque:** Botão sensível ao toque, expandindo as opções de interação tátil.
7. **Microfone Integrado e LED Indicador:** Canal de entrada acústica com LED indicatório para *feedback* visual sobre a captação de som.

Complementarmente, os componentes localizados na parte de trás da placa (Figura 2.2) estão descritos abaixo.

1. **Antena de Rádio e Bluetooth:** Comunicação sem fio para redes de placas e conectividade móvel.

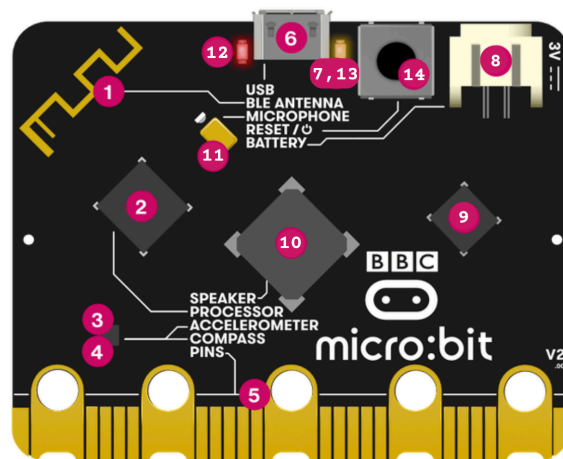


Figure 2.2. Trás da placa Micro:bit V2.

2. **Processador e Sensor de Temperatura:** Unidade central de processamento com termômetro integrado.
3. **Bússola:** Detecção de campos magnéticos e orientação espacial.
4. **Acelerômetro:** Medição de movimento, inclinação e aceleração em três eixos.
5. **Pinos de Conexão Traseiros:** Acesso simplificado à interface de expansão.
6. **Porta Micro USB:** Interface para programação e alimentação de energia.
7. **LED Amarelo de Transferência:** *Feedback* visual durante o carregamento do código.
8. **Conector da Bateria:** Permite o uso autônomo e portátil da placa.
9. **Interface USB:** Gerencia a comunicação entre o computador e o processador principal.
10. **Alto-falante Integrado:** Emissor de áudio nativo para *feedback* sonoro (músicas, tons e alarmes).
11. **Microfone (Face Posterior):** Captador de áudio e pressão sonora ambiental.
12. **LED Vermelho de Alimentação:** Indicação visual de placa energizada.
13. **LED Amarelo da Interface USB:** Indicador do status da conexão USB.
14. **Botão Multifunção de Energia e Reset:** Permite colocar a placa em modo de baixo consumo ou reiniciar o programa.

2.2.2. A Plataforma *MakeCode*

A plataforma de programação utilizada é o *MakeCode*, que possui algumas características que fornecem suporte e acessibilidade ao usuário. Como por exemplo:

- **Suporte a Leitores de Tela:** Estruturação de código e blocos compatíveis com softwares de leitura de tela, permitindo que estudantes com deficiência visual naveguem pelos blocos de código e instruções.
- **Modo de Alto Contraste:** Opção de interface gráfica com paletas ajustadas para usuários com baixa visão ou daltonismo.
- **Flexibilidade na Programação:** Permite a transição gradual e adaptável do nível de abstração. O ambiente possibilita transitar sem perdas entre a linguagem baseada em blocos visuais, *JavaScript* e *Python*, adequando-se ao nível de desenvolvimento e às preferências de interação de cada aluno.
- **Materiais Complementares:** o *MakeCode* possui guias de aplicação, diagramas táteis⁴ e tutoriais que possuem uma linguagem de fácil compreensão para qualquer público-alvo.

2.2.3. Introdução ao *MakeCode*

Para viabilizar a criação de protótipos inclusivos, a plataforma Web *MakeCode*⁵ oferece um ambiente amigável e acessível de programação baseada em blocos e texto. A seguir, detalham-se os passos fundamentais para utilização do ambiente:

2.2.3.1. Criando e Nomeando um Novo Projeto

A plataforma *MakeCode* disponibiliza uma interface para a construção, simulação e transferência de códigos para a placa Micro:bit. Para fundamentar a aplicação das atividades propostas neste capítulo, esta subseção apresenta o fluxo inicial de navegação no ambiente, detalhando os procedimentos para criação de novos projetos.

Na página inicial do *MakeCode* (Figura 2.3), deve-se selecionar a opção "Novo Projeto". Em seguida, uma janela de configuração é exibida (Figura 2.4), solicitando a inserção do nome da atividade (por exemplo, *Teste*). Após clicar em "Criar", a área de trabalho do projeto é carregada. O nome atribuído ao código permanece visível e pode ser editado a qualquer momento no campo de texto localizado na barra inferior da tela, conforme ilustrado na Figura 2.5.

2.2.3.2. Visão Geral da Interface

Após a criação do projeto, o *MakeCode* carrega a sua área de trabalho principal, conforme ilustrado na Figura 2.6. O ambiente é estruturado de forma intuitiva e dividida em três

⁴<https://microbit.org/teach/classroom-resources/microbit-tactile-diagram/>

⁵<https://makecode.microbit.org/>

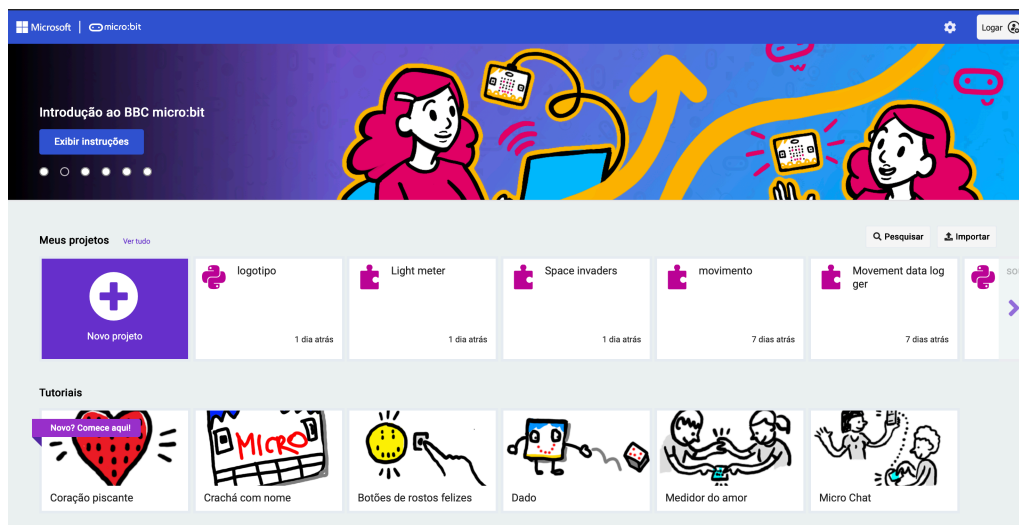


Figure 2.3. Imagem da Página Inicial da Plataforma



Figure 2.4. Imagem da Janela de Criação da Programação

regiões funcionais suplementares, projetadas para facilitar a visualização e o desenvolvimento dos programas:

- **Simulador (1):** Permite testar o comportamento do código em tempo real diretamente na tela, oferecendo retornos visuais e sonoros (como a execução de sons, matriz de LEDs e interações com os botões) antes de gravar a lógica na placa física.
- **Caixa de Ferramentas / Categorias (2):** Organiza os blocos de programação por funções essenciais (como *Básico*, *Entrada*, *Música*, *LEDs*, *Lógica* e *Variáveis*). Cada categoria conta com codificação por cores distintas, o que facilita a identificação rápida e o reconhecimento visual dos comandos.

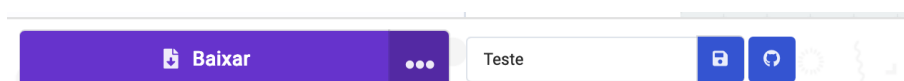


Figure 2.5. Local para Alterar o Nome do Projeto.

- **Área de Trabalho (3):** Espaço central de edição onde os blocos são arrastados, encaixados e estruturados sequencialmente para construir a lógica do programa.

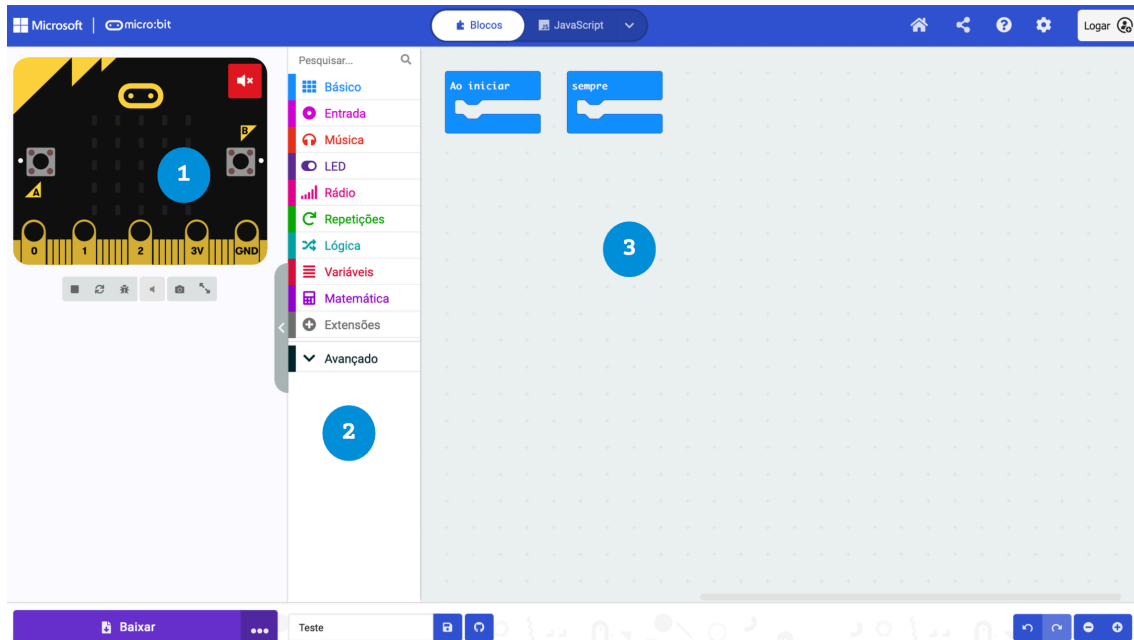


Figure 2.6. Divisão da Interface do *MakeCode*.

2.2.3.3. Transferindo o Código para a Placa Micro:bit (Via Cabo USB):

Após a lógica do programa ser construída e testada no simulador, a etapa seguinte consiste em embarcar o algoritmo no *hardware* físico. A gravação na placa Micro:bit V2 é realizada de forma simplificada via conexão USB, podendo ocorrer por comunicação direta via navegador ou por transferência manual de arquivo, conforme os passos descritos a seguir:

- **Conexão Física:** Conecte a placa Micro:bit ao computador utilizando um cabo Micro-USB adequado para dados.
- **Pareamento Direto (WebUSB):** Clique no botão "Baixar" ou acesse o ícone de três pontos ao lado dele para parear o dispositivo diretamente com o navegador (via suporte WebUSB). Esse procedimento permite que o código seja enviado à placa de forma automática a cada atualização.
- **Transferência Manual (Arquivo .hex):** Caso o pareamento direto não esteja ativado, a plataforma gera um arquivo compilado com extensão *.hex*, salvo automaticamente no computador. Para gravar o programa, basta mover ou arrastar o arquivo *.hex* para dentro da unidade de disco nomeada MICROBIT, exibida no gerenciador de arquivos do sistema operacional.

- **Confirmação de Gravação:** Durante o processo de transferência, o LED amarelo indicador de interface (representado pelo número 7 da Figura 2.2) piscará rapidamente. Após a conclusão do carregamento, a aplicação iniciará a execução automaticamente no *hardware*.

2.2.3.4. Exportando e Salvando o Arquivo de Programação:

A preservação e o compartilhamento dos artefatos desenvolvidos são etapas essenciais para promover a reusabilidade de projetos e o trabalho colaborativo entre estudantes, educadores e pesquisadores. A plataforma *MakeCode* permite exportar e reimportar arquivos de código de maneira rápida e sem perda de dados, conforme o procedimento descrito a seguir:

- **Exportação do Projeto (*Backup*):** Para salvar uma cópia de segurança no armazenamento local ou compartilhar a aplicação com outros usuários, clique no ícone "Salvar" (representado pelo símbolo de um disco no rodapé da página). Esse comando realizará o download do arquivo compilado em formato *.hex*.
- **Reimportação e Continuidade:** O arquivo *.hex* gerado contém tanto o código executável do *hardware* quanto a estrutura editável de blocos. Para dar continuidade ao desenvolvimento em outro momento ou computador, basta selecionar a opção "Importar" na tela inicial do *MakeCode* e carregar o arquivo previamente salvo.

2.2.4. Propostas de Aplicações Práticas e Sensoriais

A existência de sensores embarcados na placa do Micro:bit V2 permite a construção de artefatos acessíveis e projetos contextualizados de forma simples. Abaixo são destacadas cinco explorações pedagógicas fundamentadas nos sensores da placa tendo o público-alvo, estudantes autistas:

1. **Termômetro Digital (Sensor de Temperatura):** Utiliza o sensor embutido no processador para medir a temperatura ambiente. Permite criar alarmes térmicos sonoros ou visuais para sinalizar variações térmicas de forma tátil e/ou auditiva.
2. **Monitor de Luminosidade (Sensor de Luz):** A matriz de LEDs inverte sua função para mensurar a luz incidente. Permite desenvolver sistemas que podem auxiliar nos avisos sobre sobrecarga sensorial por muita quantidade de luz.
3. **Sonômetro (Microfone Integrado):** Mede a amplitude do som ambiente. Pode ser aplicado na criação de monitores visuais de ruído em sala de aula, convertendo volume sonoro em padrões visuais na matriz de LEDs.
4. **Rastreador de Atividades (Acelerômetro):** Identifica gestos, agitações e inclinação. Utilizado na construção de pedômetros acessíveis, alertas de postura corporal ou interfaces controladas por pequenos movimentos corporais (para alunos com mobilidade reduzida).

5. **Botão Sensível ao Toque (Logotipo Dourado):** Funciona como um atuador capacitivo adicional. Permite acionamentos suaves por proximidade ou toque leve, sendo ideal para criar comutadores acessíveis para estudantes com limitações de força motora fina nos dedos.

2.3. Práticas de Programação Acessível e Design de Atividades Inclusivas

No contexto do Transtorno do Espectro Autista (TEA), as alterações no processamento sensorial, caracterizadas por hiper ou hiporresponsividade a estímulos do ambiente, representam um dos principais fatores desencadeadores de estresse, sobrecarga cognitiva e crises comportamentais no ambiente escolar [15]. Salas de aula tradicionais frequentemente apresentam oscilações imprevisíveis de ruído, luminosidade e temperatura, criando barreiras invisíveis para a permanência e a aprendizagem desses estudantes.

A Computação Física, operacionalizada pelo Micro:bit V2 junto ao MakeCode, atuam como uma ferramenta mediadora e preditiva. Ao converter variáveis ambientais intangíveis em indicadores visuais, sonoros ou táteis claros e padronizados, a placa oferece previsibilidade e concretude ao aluno autista. Dessa forma, a tecnologia deixa de ser apenas um objeto de estudo técnico e passa a funcionar como uma ponte de comunicação não verbal e autorregulação, permitindo que o estudante compreenda o ambiente, comunique seu desconforto de forma autônoma e antecipe intervenções antes do esgotamento sensorial [14].

Nas subseções a seguir, apresenta-se um guia prático “mão na massa” com cinco protótipos desenvolvidos no ambiente MakeCode, detalhando a construção, a lógica de programação e a fundamentação pedagógico-sensorial de cada atividade. Destaca-se que as atividades propostas visam exemplificar como o aluno pode ser colocado como protagonista na implementação de soluções de TA que possam apoiar suas necessidades e desafios causados pelo ambiente. Ideias similares podem ser propostas em outros cenários, por exemplo, ao criar jogos ou recursos de TA para pessoas com outras deficiências e incentivar que alunos proponham e implementem suas próprias soluções.

2.3.1. Protótipo 1: Termômetro Digital Mediador de Conforto Térmico

Fundamentação Sensorial e Pedagógica: Estudantes autistas com hiposensibilidade ou alteração na propriocepção térmica podem apresentar dificuldades em reconhecer e nomear precocemente o próprio desconforto associado ao calor ou ao frio excessivo. Essa dificuldade de autorrelato frequentemente se manifesta como irritabilidade ou desatenção repentina. O Termômetro Digital estabelece um indicador externo e objetivo: ao transformar a temperatura do ambiente em um alerta visual imediato, o protótipo valida a percepção do estudante perante a turma e o professor, justificando ações mitigadoras (como ligar um ventilador ou abrir a janela) antes que o estresse térmico gere uma crise.

Recursos Utilizados: Sensor de Temperatura do Processador e Matriz de LEDs (5x5).

Lógica de Programação:

1. No bloco de repetição contínua (*forever*), realiza-se a leitura do sensor de temperatura interno (em graus Celsius).

2. Condicional I: Se a temperatura for superior a um limiar confortável (ex.: $> 28^{\circ}\text{C}$), a matriz exibe o ícone de alerta ou um símbolo de desconforto, sinalizando a necessidade de readequação do espaço.
3. Condicional II: Caso a temperatura esteja adequada, exibe-se uma carinha feliz ou ícone suave, sinalizando estabilidade térmica.

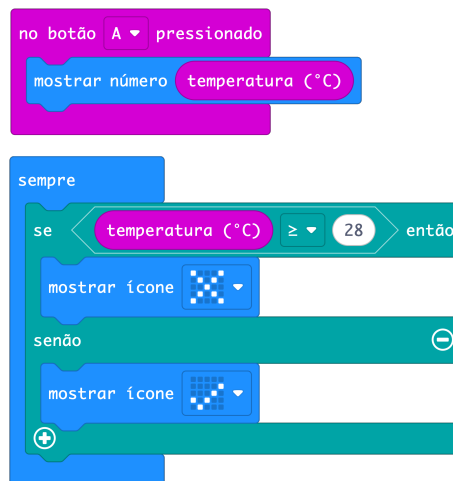


Figure 2.7. Programação em Blocos do Protótipo 1

```

1 input.onButtonPressed(Button.A, function () {
2   basic.showNumber(input.temperature())
3 })
4 basic.forever(function () {
5   if (input.temperature() >= 28) {
6     basic.showIcon(IconNames.No)
7   } else {
8     basic.showIcon(IconNames.Yes)
9   }
10 })

```

Figure 2.8. Programação em JavaScript do Protótipo 1

```

1 def on_button_pressed_a():
2   basic.show_number(input.temperature())
3 input.on_button_pressed(Button.A, on_button_pressed_a)
4
5 def on_forever():
6   if input.temperature() >= 28:
7     basic.show_icon(IconNames.NO)
8   else:
9     basic.show_icon(IconNames.YES)
10 basic.forever(on_forever)

```

Figure 2.9. Programação em Python do Protótipo 1

2.3.2. Protótipo 2: Monitor de Luminosidade e Suporte Visual à Rotina

Fundamentação Sensorial e Pedagógica: A hipersensibilidade visual a luzes fluorescentes ou variações bruscas de iluminação natural é uma queixa recorrente em indivíduos

autistas, podendo causar fadiga ocular, dor de cabeça e ansiedade. O Monitor de Luminosidade atua como um recurso não verbal de apoio à rotina: a placa monitora continuamente a luz da sala e alerta a turma quando os níveis de iluminação ultrapassam limites confortáveis, servindo também como um sinalizador visual suave para momentos de transição de atividades e relaxamento.

Recursos Utilizados: Sensor de Luz da Matriz de LEDs e Matriz de LEDs para exibição de padrões.

Lógica de Programação:

1. A matriz de LEDs inverte periodicamente sua função para ler o nível de luminosidade ambiental (escala de 0 a 255).
2. Se a luz medida exceder o limite confortável estabelecido (ex.: > 180), a placa exibe um padrão visual de aviso suave para que as luzes ou cortinas sejam ajustadas.
3. Para níveis ideais de iluminação, exibe-se um padrão de menor intensidade gráfica para não sobrecarregar visualmente o aluno.

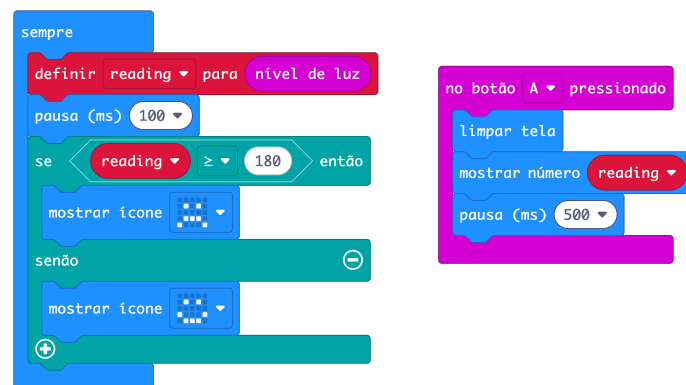


Figure 2.10. Programação em Blocos do Protótipo 2

```

1  let reading = 0
2  input.onButtonPressed(Button.A, function () {
3    basic.clearScreen()
4    basic.showNumber(reading)
5    basic.pause(500)
6  })
7  basic.forever(function () {
8    reading = input.lightLevel()
9    basic.pause(100)
10   if (reading >= 180) {
11     basic.showIcon(IconNames.Sad)
12   } else {
13     basic.showIcon(IconNames.Happy)
14   }
15 })

```

Figure 2.11. Programação em JavaScript do Protótipo 2

```

1  reading = 0
2
3  def on_button_pressed_a():
4      basic.clear_screen()
5      basic.show_number(reading)
6      basic.pause(500)
7  input.on_button_pressed(Button.A, on_button_pressed_a)
8
9  def on_forever():
10     global reading
11     reading = input.light_level()
12     basic.pause(100)
13     if reading >= 180:
14         basic.show_icon(IconNames.SAD)
15     else:
16         basic.show_icon(IconNames.HAPPY)
17     basic.forever(on_forever)

```

Figure 2.12. Programação em Python do Protótipo 2

2.3.3. Protótipo 3: Sonômetro de Sinalização Não Verbal de Ruído

Fundamentação Sensorial e Pedagógica: A sobrecarga auditiva é uma das causas mais frequentes de crises comportamentais no TEA. Sons agudos ou sobreposição de vozes no laboratório de informática podem desencadear respostas de fuga ou defesa. O Sonômetro transforma o nível sonoro do ambiente em um barramento gráfico visual na matriz de LEDs. Esse recurso transfere a responsabilidade da autorregulação para o grupo: em vez de o professor utilizar comandos verbais impositivos, que adicionam mais estímulo auditivo à sala, a proposta é a própria matriz indicar se o ruído está adequado, permitindo a autorregulação coletiva e protetiva.

Recursos Utilizados: Microfone Integrado e Matriz de LEDs.

Lógica de Programação:

1. O microfone capta continuamente o nível de som ambiente (escala de 0 a 255).
2. Utiliza-se a função nativa *plot bar graph* para mapear proporcionalmente o som na matriz de 5x5 colunas.
3. Se o som ultrapassar um valor crítico de sobrecarga (ex.: > 150), o Micro:bit exibe um alerta visual imediato.

2.3.4. Protótipo 4: Rastreador de Atividades

Fundamentação Sensorial e Pedagógica: Comportamentos motores repetitivos ou estereotipados (como o *flapping* de mãos ou o balanço do corpo) desempenham um papel legítimo de autorregulação emocional e sensorial no autismo. Em vez de reprimir ou punir tais movimentos, o Rastreador de Atividades utiliza o acelerômetro para validar e reorientar essa necessidade motora. O movimento físico do aluno é canalizado para uma interação computacional intencional e gamificada, fornecendo *feedback* visual e reforço positivo imediato à medida que o aluno atinge metas de movimento.

Recursos Utilizados: Acelerômetro Integrado (Detecção de Gestos e Aceleração) e Matriz de LEDs.

Lógica de Programação:

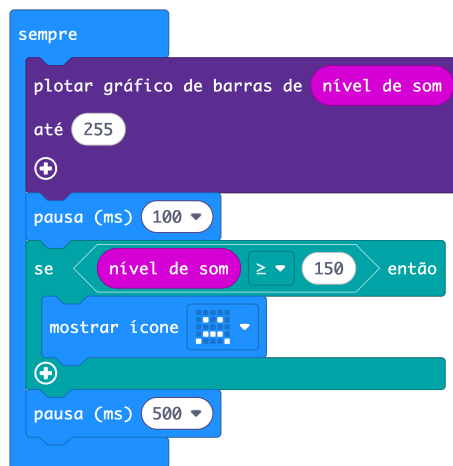


Figure 2.13. Programação em Blocos do Protótipo 3

```

1 basic.forever(function on_forever() {
2   led.plotBarGraph(input.soundLevel(), 255)
3   basic.pause(100)
4   if (input.soundLevel() >= 150) {
5     basic.showIcon(IconNames.Sad)
6   }
7
8   basic.pause(500)
9 })

```

Figure 2.14. Programação em JavaScript do Protótipo 3

1. Declara-se uma variável global de contagem de movimentos (*contador*).
2. Ao detectar o evento de movimento vigoroso do dispositivo (*on shake*), a variável é incrementada em 1 unidade.
3. A matriz exibe o número atualizado de repetições e celebra marcas atingidas com ícones visuais expansivos.

2.3.5. Protótipo 5: Botão de Autorregulação e Descompressão Tátil

Fundamentação Sensorial e Pedagógica: Em momentos de início de sobrecarga sensorial ou aumento de ansiedade, o estudante necessita de uma estratégia rápida de ancoragem (*grounding*). O Botão Sensível ao Toque (logotipo capacitivo dourado) atua como um comutador de acessibilidade: com um toque suave e sem a necessidade de aplicação de força mecânica, o aluno ativa um estímulo sonoro auditivo (como uma melodia suave)

```

1 def on_forever():
2   led.plot_bar_graph(input.sound_level(), 255)
3   basic.pause(100)
4   if input.sound_level() >= 150:
5     basic.show_icon(IconNames.SAD)
6   basic.pause(500)
7 basic.forever(on_forever)

```

Figure 2.15. Programação em Python do Protótipo 3

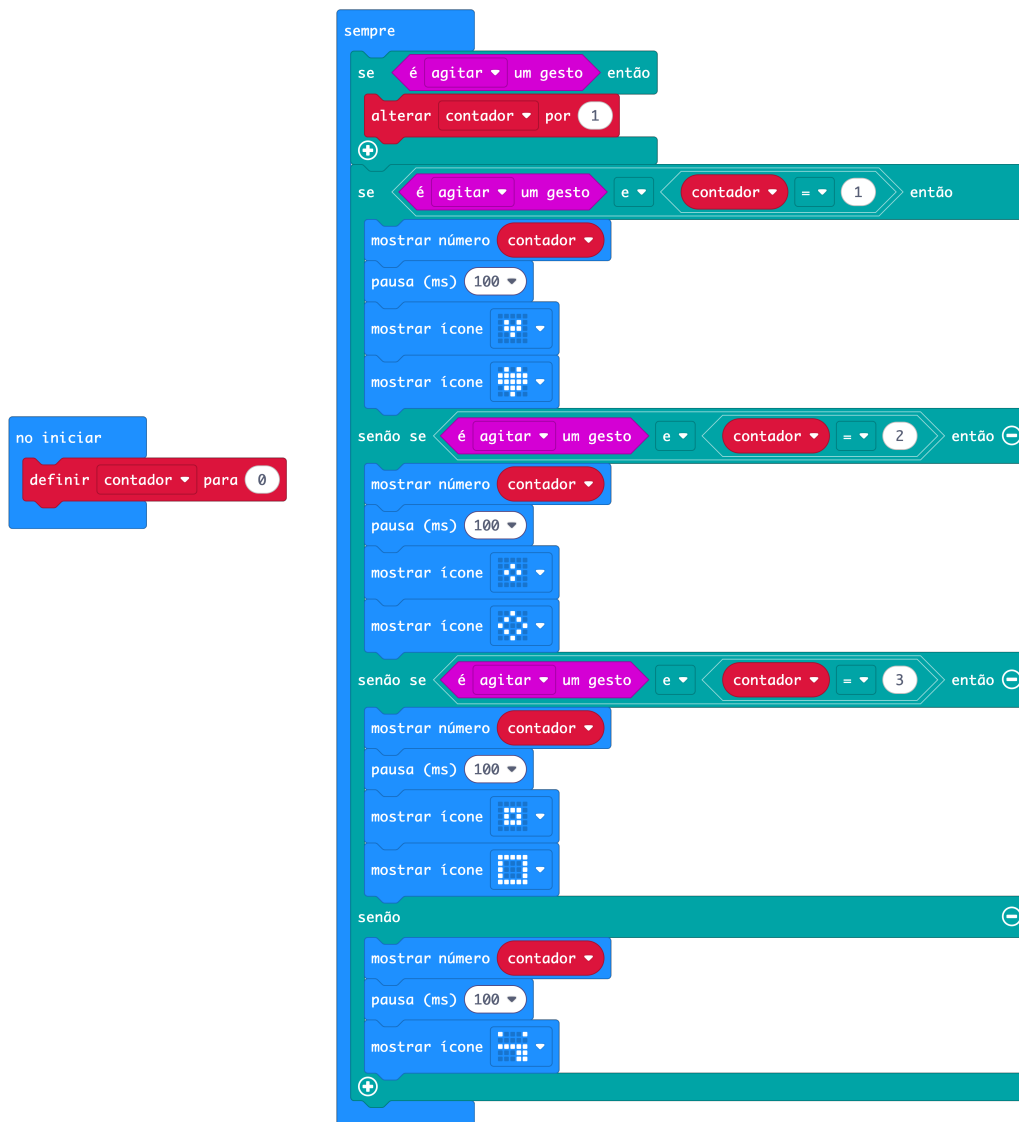


Figure 2.16. Programação em Blocos do Protótipo 4

ou um padrão de animação visual rítmica e previsível na placa. Esse recurso oferece ao próprio estudante o controle ativo sobre um estímulo aconchegante para retomada do foco.

Recursos Utilizados: Logotipo Sensível ao Toque Capacitivo e Alto-falante Integrado.

Lógica de Programação:

1. Configura-se o bloco de evento para o toque no logotipo capacitivo (*on logo pressed*).
2. Ao ser tocado, o Micro:bit reproduz um efeito sonoro suave e previsível via alto-falante nativo, acompanhado por um padrão gráfico desacelerado na matriz de LEDs.

```

1  let contador = 0
2  basic.forever(function () {
3      if (input.isGesture(Gesture.Shake)) {
4          contador += 1
5      }
6      if (input.isGesture(Gesture.Shake) && contador == 1) {
7          basic.showNumber(contador)
8          basic.pause(100)
9          basic.showIcon(IconNames.SmallHeart)
10         basic.showIcon(IconNames.Heart)
11     } else if (input.isGesture(Gesture.Shake) && contador == 2) {
12         basic.showNumber(contador)
13         basic.pause(100)
14         basic.showIcon(IconNames.SmallDiamond)
15         basic.showIcon(IconNames.Diamond)
16     } else if (input.isGesture(Gesture.Shake) && contador == 3) {
17         basic.showNumber(contador)
18         basic.pause(100)
19         basic.showIcon(IconNames.SmallSquare)
20         basic.showIcon(IconNames.Square)
21     } else {
22         basic.showNumber(contador)
23         basic.pause(100)
24         basic.showIcon(IconNames.Silly)
25     }
26 })

```

Figure 2.17. Programação em JavaScript do Protótipo 4

2.3.6. Orientações Metodológicas para Aplicação e Adaptabilidade em Sala de Aula

Para que a implementação dessas práticas de computação física com o Micro:bit seja efetiva em turmas inclusivas com estudantes autistas, recomendam-se as seguintes diretrizes metodológicas para professores e pesquisadores:

- **Previsibilidade e Antecipação:** Apresentar previamente o Micro:bit, os sons que ele pode emitir e as luzes que serão acesas, permitindo que o estudante explore o dispositivo em um ambiente calmo antes da atividade coletiva.
- **Ajuste Fino de Parâmetros de Sensibilidade:** Os limiares de acionamento (níveis de som, luz e sensibilidade do toque) devem ser calibrados individualmente em conjunto com o estudante, garantindo que o alerta reflita o limite real de conforto de cada aluno.
- **Uso do Suporte Físico e Cases de Proteção:** Utilizar capas de silicone ou suportes de proteção tátil na placa, que além de proteger o circuito contra quedas, suavizam o contato térmico e tátil para alunos com hipersensibilidade.
- **Gamificação Flexível e Sem Pressão Temporal:** Estruturar os desafios de programação sem impor prazos rígidos ou dinâmicas competitivas ruidosas, priorizando o progresso individual e o aprendizado cooperativo.
- **Construção de Espaços de Descompressão:** Permitir que os protótipos de autorregulação (como o botão de toque ou o monitor de ruído) fiquem disponíveis no ambiente para uso espontâneo do aluno sempre que sentir necessidade.

```

1  contador = 0
2
3  def on_forever():
4      global contador
5      if input.is_gesture(Gesture.SHAKE):
6          contador += 1
7      if input.is_gesture(Gesture.SHAKE) and contador == 1:
8          basic.show_number(contador)
9          basic.pause(100)
10         basic.show_icon(IconNames.SMALL_HEART)
11         basic.show_icon(IconNames.HEART)
12     elif input.is_gesture(Gesture.SHAKE) and contador == 2:
13         basic.show_number(contador)
14         basic.pause(100)
15         basic.show_icon(IconNames.SMALL_DIAMOND)
16         basic.show_icon(IconNames.DIAMOND)
17     elif input.is_gesture(Gesture.SHAKE) and contador == 3:
18         basic.show_number(contador)
19         basic.pause(100)
20         basic.show_icon(IconNames.SMALL_SQUARE)
21         basic.show_icon(IconNames.SQUARE)
22     else:
23         basic.show_number(contador)
24         basic.pause(100)
25         basic.show_icon(IconNames.SILLY)
26 basic.forever(on_forever)

```

Figure 2.18. Programação em Python do Protótipo 4



Figure 2.19. Programação em Blocos do Protótipo 5

2.4. Desafios de Avaliação

A avaliação do impacto pedagógico e da usabilidade de Recursos de Tecnologia Assistiva (RTA) desenvolvidos por meio da computação física apresenta complexidades particulares. Diferente do software educacional tradicional, cujas métricas de sucesso baseiam-se predominantemente no tempo de execução de tarefas ou no acerto de respostas em tela, a eficácia de um protótipo físico e acessível como o Micro:bit exige uma abordagem avaliativa holística. É necessário mensurar não apenas o ganho conceitual em computação, mas também o grau de autonomia, o conforto sensorial, a redução de barreiras de interação e a aceitação social da solução no ambiente escolar.

```

1 input.onLogoEvent(TouchButtonEvent.Pressed, function () {
2   music.play(music.builtinPlayableSoundEffect(soundExpression.happy), music.PlaybackMode.InBackground)
3 })
4 basic.forever(function () {
5   basic.showIcon(IconNames.SmallHeart)
6   basic.showIcon(IconNames.Heart)
7 })

```

Figure 2.20. Programação em JavaScript do Protótipo 5

```

1 def on_logo_pressed():
2     music.play(music.builtin_playable_sound_effect(soundExpression.happy),
3               music.PlaybackMode.IN_BACKGROUND)
4 input.on_logo_event(TouchButtonEvent.PRESSED, on_logo_pressed)
5
6 def on_forever():
7     basic.show_icon(IconNames.SMALL_HEART)
8     basic.show_icon(IconNames.HEART)
9 basic.forever(on_forever)

```

Figure 2.21. Programação em Python do Protótipo 5

2.4.1. Métricas e Métodos de Avaliação de Usabilidade e Acessibilidade

No campo da Interação Humano-Computador (IHC), a usabilidade de um dispositivo assistivo é definida pela extensão em que o recurso permite que usuários com diferentes perfis funcionais alcancem objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso real (ISO 9241-11). Quando aplicada a estudantes neurodivergentes ou com deficiência, a avaliação de usabilidade deve adaptar métodos consagrados para evitar sobrecarga estressante:

- **Adaptação da Escala de Usabilidade do Sistema (SUS - *System Usability Scale*):** A escala SUS [9] de 10 itens pode ser simplificada e traduzida para formatos visuais ou pictográficos (usando cartões com expressões faciais ou escalas de Likert visuais), permitindo que estudantes autistas ou com dificuldades de comunicação verbal expressem diretamente o grau de facilidade ou frustração ao manipular o protótipo.
- **Protocolo de Pense em Voz Alta Adaptado (*Think-Aloud*):** Em vez de exigir o relato verbal contínuo durante o uso da placa, prioriza-se a observação estruturada de comportamento não verbal, registrando hesitações, buscas por ajuda, sorrisos de engajamento ou sinais de desconforto tátil ao interagir com os sensores e botões do Micro:bit.
- **Análise de Carga Mental e Estresse Sensorial:** Utilização de métricas qualitativas e observacionais para identificar se a resposta do hardware (sons emitidos, brilho dos LEDs) reduz ou eleva a carga cognitiva e a ansiedade do aluno durante a atividade.
- **Manequim de Autoavaliação (*Self-Assessment Manikin - SAM*):** Utilização da ferramenta não verbal e pictográfica SAM [8] como uma alternativa acessível para avaliar as dimensões emocionais, como valência, alerta e dominância, na interação

com os protótipos. Por meio de figuras ilustrativas simplificadas, o protocolo permite que os participantes expressem suas respostas afetivas e o nível de conforto sem a necessidade de relatos verbais complexos.

2.4.2. Métricas de Impacto Pedagógico e Aprendizagem Inclusiva

A avaliação do impacto pedagógico fundamenta-se nos princípios do *Design Universal* para a Aprendizagem (DUA), mensurando em que medida a introdução da computação física tangível expandiu o acesso do estudante ao conhecimento de programação. As principais dimensões avaliativas incluem:

1. **Ganho de Autonomia e Participação Ativa:** Verificação da diminuição da necessidade de suporte verbal direto ou mediação constante do professor à medida que o aluno domina o manuseio e a lógica do protótipo.
2. **Apropriação do Pensamento Computacional:** Avaliação do domínio de conceitos chave (como algoritmos, variáveis, condicionais e eventos de entrada/saída) por meio da demonstração prática no objeto físico, e não estritamente por provas escritas ou testes formais.
3. **Eficácia na Mediação Sensorial e Autorregulação:** Nos protótipos voltados ao ambiente escolar (como o sonômetro e o monitor térmico), mede-se a capacidade da ferramenta em antecipar e mitigar desorganizações comportamentais, reduzindo a frequência de crises sensoriais durante as aulas de informática.

2.4.3. Desafios Metodológicos na Coleta de Dados em Ambientes Inclusivos

Pesquisadores e educadores enfrentam desafios éticos e práticos significativos ao avaliar soluções assistivas em sala de aula:

- **Heterogeneidade dos Perfis de Aprendizagem:** A extrema variabilidade de habilidades e necessidades dentro do espectro autista e de outras neurodivergências inviabiliza amostragens estatísticas padronizadas rigorosas. Métodos baseados em Estudo de Caso Único (*Single-Subject Research Design*) ou Pesquisa-Ação mostram-se mais adequados por permitirem comparar o desempenho e o comportamento do próprio aluno antes, durante e após a introdução da tecnologia assistiva.
- **Efeito Novidade versus Uso Continuado:** Um desafio recorrente é diferenciar o entusiasmo passageiro provocado pela luz e som da placa Micro:bit (efeito novidade) da real eficácia da ferramenta como recurso permanente de aprendizagem e autorregulação. Avaliações longitudinais são recomendadas para verificar a sustentabilidade do uso.
- **Triangulação de Fontes de Dados:** Para obter dados robustos e confiáveis sem expor o estudante a testes invasivos, recomenda-se a triangulação metodológica, combinando: (a) registros em diários de campo do pesquisador/professor; (b) observação videográfica estruturada do comportamento de interação; e (c) percepção dos mediadores (professores, tutores e familiares) sobre mudanças no engajamento e bem-estar do aluno.

2.5. Considerações Finais

A busca pela democratização do acesso às tecnologias digitais e pelo ensino inclusivo de programação exige o desenvolvimento constante de novas abordagens pedagógicas e metodológicas. Este capítulo apresentou a Computação Física Inclusiva apoiada pelo ecossistema Micro:bit V2 e pela plataforma *MakeCode* como uma alternativa viável, de baixo custo e integrativa para mitigar as barreiras sensoriais, motoras e cognitivas presentes nos ambientes tradicionais de ensino de computação.

Ao fundamentar as práticas nos preceitos da IHC voltada à acessibilidade, do *Design Universal* para a Aprendizagem (DUA) e no arcabouço normativo nacional (a exemplo da PNED, BNCC-Computação e LBI), o trabalho demonstrou como a tangibilidade e o *feedback* multimodal (visual, sonoro e tátil) podem transformar o aprendizado de conceitos abstratos em uma experiência concreta. Em particular, a proposição de protótipos funcionais voltados ao contexto do TEA traz possibilidades do uso dessas ferramentas como RTA não apenas para o aprendizado técnico, mas também para a mediação sensorial, previsibilidade e autorregulação dos estudantes.

Além do desenvolvimento de projetos, o capítulo discutiu as particularidades e os desafios associados à avaliação de usabilidade e impacto pedagógico em ambientes inclusivos. A adaptação de protocolos consagrados (como a simplificação pictográfica do SUS, a aplicação da escala não verbal SAM e a condução do *Think-Aloud* adaptado) tornam-se opções para capturar respostas emocionais e atitudinais de forma ética, respeitando a heterogeneidade dos perfis dos participantes e superando abordagens avaliativas estritamente quantitativas ou formais.

Apesar das vantagens mapeadas, reconhece-se que a adoção efetiva da computação física inclusiva nas escolas enfrenta desafios operacionais, como a necessidade de formação continuada de educadores, o investimento contínuo na infraestrutura dos laboratórios e a superação do "efeito novidade" por meio de práticas pedagógicas longitudinais e sustentáveis.

Como trabalhos futuros, incentiva-se a realização de estudos com pesquisa-ação para avaliar a usabilidade e a eficácia pedagógica dos protótipos propostos diretamente em salas de aula inclusivas. Ademais, sugere-se a ampliação do escopo de projetos para contemplar demandas de estudantes com outras especificidades funcionais (como deficiências motoras severas ou deficiência visual), estimulando o protagonismo dos próprios alunos na concepção, prototipagem e validação de suas próprias soluções de Tecnologia Assistiva.

References

- [1] BRASIL. (2023) Lei nº 14.533, de 11 de janeiro de 2023. Brasília, DF: Presidência da República.
- [2] BRASIL. (2022) Ministério da Educação. Computação na Educação Básica: Complemento à BNCC. Brasília, DF: MEC.
- [3] BRASIL. (2021) Ministério da Educação. Secretaria de Educação Básica. *Programa Nacional do Livro e do Material Didático (PNLD)*. Brasília, DF: FNDE/MEC.

Disponível em: <<https://www.gov.br/fnde/>>.

- [4] Humberto A. P. Zanetti e Marcos A. F. Borges. (2020) Ensino de Programação utilizando Computação Física: uma Revisão Sistemática da Literatura. *Comunicações em Informática*, v. 4, n. 1, p. 19–22. ISSN 2595-0622. DOI: 10.22478/ufpb.2595-0622.2020v4n1.52065.
- [5] Molly Follette Story, James L. Mueller e Ronald L. Mace. (1998) *The Universal Design File: Designing for People of All Ages and Abilities*. Edição revista. Center for Universal Design, North Carolina State University, Raleigh, NC. ERIC ED 460 554.
- [6] Angelaki, S., Triantafyllidis, G. A., Besenecker, U. (2022). Lighting in Kindergartens: Towards Innovative Design Concepts for Lighting Design in Kindergartens Based on Childrens Perception of Space. *Sustainability*, 14(4), 2370. <https://doi.org/10.3390/su14042370>.
- [7] Freitas, M. L. P. de, Pires, F., Pessoa, M. (2025). Uso de Computação Física para desenvolvimento de Pensamento Computacional: um estudo de caso. In *Anais do XXXI Workshop de Informática na Escola* (pp. 475487). Sociedade Brasileira de Computação (SBC). <https://doi.org/10.5753/wie.2025.13490>.
- [8] Bynion, T. M., Feldner, M. T. (2017). Self-assessment manikin. In V. Zeigler-Hill, T. K. Shackelford (Eds.), *Encyclopedia of Personality and Individual Differences* (pp. 1–3). Springer. https://doi.org/10.1007/978-3-319-28099-8_77-1
- [9] Brooke, J. SUS: A quick and dirty usability scale. In: Jordan, P. W.; Thomas, B.; Weerdmeester, B. A.; McClelland, I. L. (eds.). *Usability Evaluation in Industry*. London: Taylor and Francis, 1996. p. 189194.
- [10] BRASIL. (2021) Decreto nº 10.645, de 11 de março de 2021. Regulamenta o art. 75 da Lei nº 13.146, de 6 de julho de 2015, para instituir o Plano Nacional de Tecnologia Assistiva. *Diário Oficial da União*, Brasília, DF.
- [11] BRASIL. (2004) Decreto nº 5.296, de 2 de dezembro de 2004. Regulamenta as Leis nº 10.048, de 8 de novembro de 2000, que dá prioridade de atendimento às pessoas que especifica, e nº 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade. *Diário Oficial da União*, Brasília, DF.
- [12] Micro:bit Educational Foundation, “Micro:bit impact report and educational resources,” 2020. [Online]. Available: <https://microbit.org>
- [13] Zerbato, A. P., Mendes, E. G. (2018). Desenho universal para a aprendizagem como estratégia de inclusão escolar. *Educação Unisinos*, 22(2), 147–155.
- [14] Koegel, R., Kim, S., Koegel, L., & Schwartz, B. (2013). Improving Socialization for High School Students with ASD by Using Their Preferred Interests. *Journal of Autism and Developmental Disorders*, 43(9), 2121–2134. <https://doi.org/10.1007/s10803-013-1765-3>

- [15] AMERICAN PSYCHIATRIC ASSOCIATION. (2013) *Diagnostic and Statistical Manual of Mental Disorders (DSM-5)*. 5th ed. Washington, DC: American Psychiatric Association. <https://doi.org/10.1176/appi.books.9780890425596>