

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Interfaces Tangíveis para Educação Inclusiva: Adaptação de Metodologias de Programação em Blocos para Utilizadores com Deficiência Visual

Vitor H. Hokino

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientador
Prof. Dr. Jorge Henrique Cabral Fernandes

Brasília
2025



Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Interfaces Tangíveis para Educação Inclusiva: Adaptação de Metodologias de Programação em Blocos para Utilizadores com Deficiência Visual

Vitor H. Hokino

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Prof. Dr. Jorge Henrique Cabral Fernandes (Orientador)
CIC/UnB

Prof. Dr. Jonathan Rosa Moreira
Universidade de Brasília

Prof. Dr. Edison Ishikawa
Universidade de Brasília

Prof. Dr. Flavio Marzadro
Universidade de Brasília

Prof. Dr. Jorge Henrique Cabral Fernandes
Coordenador do Curso de Computação — Licenciatura

Brasília, 20 de agosto de 2025

Dedicatória

Dedico este trabalho acadêmico a todos os deficientes visuais (DVs) e profissionais envolvidos na educação de deficientes visuais (DVs), por este autor acreditar no poder do ensino do currículo da Computação para transformar as vidas dessas pessoas em um futuro próspero e repleto de novas oportunidades.

Dedico este trabalho acadêmico também a meu filho e ao designer de jogos eletrônicos Hideo Kojima, cujas paixões infinitas pela criação de jogos inspiram este autor no desenvolvimento da pesquisa científica aqui relatada.

De Sapiens a Ludens
Nós somos Homo Ludens (Aqueles que Jogam).

Desde o momento em que adentramos este mundo,
Instintivamente inventamos maneiras de ter "diversão"
e compartilhar nossas invenções com aqueles ao nosso redor.
Isso não é solicitado a nós,
tampouco precisamos de razões para criar.
Isso é simplesmente quem somos.

Encontramos uns aos outros e competimos entre nós mesmos,
Rimos e choramos juntos: tudo enquanto jogamos juntos.
Nossas experiências entrelaçam-nos e liberam-nos.
Para compartilhar nossas mais valiosas experiências,
criamos estórias, inventamos ferramentas, evoluimos a arte de jogar.
O jogo tem sido nosso aliado desde o berço da civilização.

"Jogar" não é um mero passatempo,
é a base primordial da imaginação e criação.
Verdade seja dita, Homo Ludens (Aqueles que Jogam) são
simultaneamente Homo Faber (Aqueles que Criam).

Ainda que a Terra fosse despojada de vida e reduzida a um terreno baldio,
nossa imaginação e desejo de criar perseverariam para além da sobrevivência,
providenciariam esperança de que talvez flores em um dia novamente floresceriam.
Através da invenção do jogar, nossa nova evolução aguarda.

KOJIMA PRODUCTIONS
Nós somos Homo Ludens. Nós somos Aqueles que Jogam.

Hideo Kojima

Agradecimentos

Gratidão deste autor a todos os envolvidos na realização deste trabalho.

Obrigado ao corpo docente, técnico e discente do Centro de Ensino Especial para Deficientes Visuais (CEEDV), por acolher a pesquisa, envolver-se com o desenvolvimento dela , inclusive destinando recursos institucionais próprios para tal.

Obrigado ao Prof. Flávio Marzadro por atuar como coorientador e co-pesquisador, o qual me apresentou ao Centro de Ensino Especial para Deficientes Visuais (CEEDV), prestou amplo apoio na realização da pesquisa de campo e na escrita da monografia.

Obrigado ao Prof. Jorge Henquique Cabral Fernandes por me orientar neste desafio de realizar o TCC e conseguir a formatura.

Obrigado ao Dr. Diogo Vieira Tibery pelo grande auxílio na escrita final deste trabalho.

Obrigado à minha mãe, Rosmery Tomoe Yamane, e meu pai, Milton Hidenobu Hokino, pelo apoio incondicional prestado à minha carreira acadêmica para torná-la possível.

Obrigado à Profa. Maria de Fátima Ramos Brandão, Profa. Lívia Santos Brizolla, ao Centro de Ensino Médio 01 do Paranoá e à Coordenação Regional de Ensino do Paranoá por possibilitarem a prática experimental de ensino em computação com deficientes visuais (DVs) que originou o trabalho desta tese.

Obrigado à *Google for Education* pela distribuição livre e gratuita do Blockly.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

O ensino de algoritmos e lógica de programação vem se consolidando em currículos escolares, contudo a predominância de interfaces visuais cria barreiras para a inclusão digital de estudantes com deficiência visual. Este trabalho investiga o uso da computação desplugada como fundamento teórico para trazer o ensino de algoritmos para alunos com deficiências visuais pelo uso de blocos manipuláveis como material didático inclusivo que representa instruções e programas tangíveis. Adotando a metodologia *Design Science Research*, a qual destina-se à criação de design de tecnologias para resolver problemas e desenvolver teorias sobre o comportamento humano dos usuários, a pesquisa partiu de atividades exploratórias no Centro de Ensino Médio 01 do Paranoá e no Centro de Educação Especial para Deficientes Visuais, ambas instituições localizadas no Distrito Federal, Brasil. As atividades conduzidas tiveram o objetivo de produzir observações que preencham o vazio de conhecimento técnico-científico sobre o problema para subsidiar o design de um artefato educacional. Os resultados obtidos evidenciam novas descobertas sobre a possibilidade do ensino de algoritmos para deficientes visuais, sobretudo pelo potencial da ludicidade para representar tangivelmente a execução de algoritmos programáveis em tempo real e colaborativamente por uma dinâmica de jogo cooperativo em dupla. Esta abordagem de computação desplugada para o ensino de algoritmos tem potencial para proporcionar, em uma versão de artefato mais refinada, um método pedagógico que oferece agência aos alunos para criarem seus próprios problemas e programas, e menores custos de produção e distribuição em comparação com abordagens de computação plugada dependentes de computadores e demais eletrônicos.

Palavras-chave: computação desplugada, deficiência visual, execução tangível de código, programação baseada em blocos

Abstract

The teaching of algorithms and programming logic has been increasingly consolidated in school curricula; however, the predominance of visual interfaces creates barriers to the digital inclusion of students with visual impairments. This work investigates the use of unplugged computing as a theoretical foundation to bring algorithm education to visually impaired students through the use of manipulable blocks as inclusive educational materials that represent tangible instructions and programs. Adopting the Design Science Research methodology, which aims to create technology designs to solve problems and develop theories about user behavior, the study began with exploratory activities at *Centro de Ensino Médio 01 do Paranoá* and the *Centro de Educação Especial para Deficientes Visuais*, both located in the Federal District of Brazil. The activities conducted sought to produce observations that fill the gap in technical and scientific knowledge on the topic to support the design of an educational artifact. The results reveal new insights into the feasibility of teaching algorithms to visually impaired students, especially considering the potential of playfulness to tangibly represent the execution of programmable algorithms in real time and collaboratively through a cooperative two-player game dynamic. This unplugged computing approach to teaching algorithms has the potential, in a more refined version of the artifact, to provide a pedagogical method that offers students agency to create their own problems and programs, along with lower production and distribution costs compared to plugged computing approaches that depend on computers and other electronic devices.

Keywords: unplugged computing, vision impairment, tangible code execution, block-based programming

Sumário

1 Problema e Contexto **1**

1.1 Aprendizagem de Programação por indivíduos portadores de deficiência visual 1

1.2 Relato do autor de experiência prévia de aprendizagem de programação com um aluno cego 2

1.3 Centro de Ensino Especial para Deficientes Visuais de Brasília 9

1.4 Definição do objeto de estudo 10

1.4.1 Computação Plugada 11

1.4.2 Computação Desplugada 11

1.5 Objetivos da pesquisa 12

1.5.1 Objetivo geral 12

1.5.2 Objetivos específicos 12

1.6 Estrutura da monografia 12

2 Metodologia **13**

2.1 Contexto e problema 13

2.2 Artefato 14

2.3 Pergunta de Pesquisa e Hipóteses de Trabalho 15

2.4 Conjecturas Teóricas 15

2.5 Modelo cíclico-iterativo e avaliações da Design Science Research 17

2.6 Avaliação 1: O artefato é válido? 18

2.7 Avaliação 2: O problema foi resolvido? 18

2.8 Avaliação 3: As conjecturas teóricas parecem válidas? 18

3 Conjecturas Teóricas (Sobre Blocos Manipuláveis) **19**

3.1 A execução tangível de um algoritmo em pseudocódigo por computação desplugada possibilita que DVs leigos sejam capazes de aprender a programar? 19

3.2 A ludicidade é capaz de dar significância aos conceitos digitais da computação a fim de promover a aprendizagem significativa de algoritmos em linguagem computacional por DVs? 21

4 Quadro Teórico **23**
4.1 Raciocínio Computacional e Computação Desplugada 23
4.2 Teoria Sociocultural de Lev Vygotsky 24
4.3 Homo Ludens 25

5 Estado da Técnica **28**
5.1 Aprendendo a programar de forma geral 28
5.1.1 Block-Based Programming 28
5.1.2 Blockly 28
5.1.3 Scratch 29
5.1.4 AlgoBlock 29
5.2 Aprendendo a programar com deficiência visual 30
5.2.1 Tangible Block Editor 30
5.2.2 Torino 31
5.2.3 Outros 32

6 Desenvolvimento do Artefato **33**
6.1 Envolvimento de stakeholders 33
6.2 Escrita Braille sobreposta à impressão em tinta 34
6.3 Labirinto em peças de *LEGO* 37
6.4 Artefatos com velcro 39
6.5 Artefatos com fixação em pinos 42
6.6 Artefatos com encaixe guiado por lacunas 45
6.7 Artefatos em impressão 3D 48

7 Avaliações **52**
7.1 Os artefatos desenvolvidos são válidos? 52
7.1.1 Validade didática da Programação Baseada em Blocos Tangíveis em LEGO para ensino de algoritmos à DVs leigos 52
7.1.2 Validade pedagógica da Programação Baseada em Blocos Tangíveis em LEGO como método de aculturamento em programação para DVs leigos 53
7.2 O problema foi resolvido? 54
7.3 As conjecturas teóricas foram desafiadas pela sua experiência? 55

8 Conclusões **56**
8.1 O trabalho cumpriu os objetivos? 56
8.2 Limitações do trabalho 56
8.3 Propostas futuras 56

Referências **59**

Lista de Figuras

1.1 Exercício que compreende parte da avaliação diagnóstica aplicada ao aluno E.J.. 3
1.2 Código em linguagem de programação Python3 que representa a estrutura de dados manipulada no exercício da Figura 1.2 pelo aluno E.J. e seu tutor durante a avaliação diagnóstica. 4
1.3 Fluxograma em impressão de tinta para enxergantes sobre ações e decisões a serem tomadas ao acordar. 5
1.4 Versão Braille do fluxograma em impressão de tinta da Figura 1.3. A escrita foi feita com lápis pelo autor para o aluno E.J. por falta de espaço para o uso de punção de escrita Braille. 6
1.5 Fluxograma com formas em Braille com algoritmo de multiplicação por somas repetidas de um número por ele mesmo. 7
1.6 Brinquedo sensorial Pop-It! usado em conjunto com o fluxograma da Figura 1.5 como tabela de variáveis e representar tangivelmente a execução do algoritmo de multiplicação. 8
1.7 Código Python de uma função que executa uma multiplicação de dois números de acordo com a lógica do algoritmo do fluxograma na Figura 1.5. . 8

2.1 Fluxograma da estrutura do DSR, contendo suas unidades de análise e as relações entre elas. Fonte: Página 10 de [1]. 17

6.1 Esboço manuscrito de labirintos e algoritmo do *Blockly Games*. 34
6.2 Visão superior do bloco avançar, montado por 3 peças de LEGO e uma tag de acetato e papel com escrita braille fixada por fita dupla face. 35
6.3 Usuário com baixa visão lendo o código tangível construído sobre a base fixadora com o canto externo do olho direito a poucos centímetros de distância da escrita negrito em fonte Verdana tamanho 18. 36
6.4 Labirinto de *LEGO* com peças de 1 unidade de largura. 37
6.5 Par de elementos do artefato lado a lado: labirinto e algoritmo tangível. . . 38

6.6 Uso do artefato por dupla em pair-programming: programador à esquerda e navegador à direita. 39
6.7 Primeira iteração do artefato, com blocos tangíveis em madeira MDF. . . . 40
6.8 As três variações em largura dos blocos MDF: 2 cm (superior); 1,5 (meio); 1 cm (inferior). 41
6.9 Tabuleiro de xadrez para cegos, adaptado com fixação das peças em pinos. 42
6.10 Pino duplo de LEGO e bloco com furações laterais. 43
6.11 Bloco tangível em LEGO com pinos duplos fixados em cada ponta na parte inferior. 43
6.12 Blocos em LEGO fixados por pinos duplos na base fixadora de blocos furados, formando três linhas de código tangível. 44
6.13 Design de encaixe vertical dos blocos guiado por lacunas. 46
6.14 Algoritmo construído por blocos de encaixe vertical guiado por lacunas, com indentação completa tangível. 47
6.15 Vista superior de amostra do artefato com encaixe deslizante, feito em impressora 3D e plástico PBS. 48
6.16 Vista lateral de amostra do artefato com encaixe deslizante, feito em impressora 3D e plástico PBS. 49
6.17 Vista de frente de amostra do artefato com encaixe de pinos, feito em impressora 3D e plástico PBS. 50
6.18 Vista lateral de amostra do artefato com encaixe de pinos, feito em impressora 3D e plástico PBS. 51

Lista de Abreviaturas e Siglas

CAP Centro de Apoio às Pessoas com Deficiência Visual.

CEEDV Centro de Ensino Especial para Deficientes Visuais.

DSR *Design Science Research*.

DVs deficientes visuais.

PBB Programação Baseada em Blocos.

UnB Universidade de Brasília.

Capítulo 1

Problema e Contexto

1.1 Aprendizagem de Programação por indivíduos portadores de deficiência visual

A inclusão digital de deficientes visuais (DVs) é um desafio cuja superação traz grandes barreiras à emancipação desse grupo na sociedade em busca de gerar indivíduos independentes e autônomos.

Limitações e ausências em recursos de acessibilidade digitais geram problemas de soluções inacessíveis, inclusive em ambientes escolares [2] [3] [4]. Isto é debilitante à inclusão digital de alunos DVs como política pública, considerando a necessidade de educação em computação em ambientes de aprendizagem de programação de sistemas digitais e computadores nos currículos escolares [5].

Os computadores digitais em suas diversas formas (celulares, *desktops*, *notebooks*, *SmartTVs*, eletrodomésticos inteligentes etc.) estão constantemente evoluindo com funcionalidades e interfaces humano-computador com designs majoritariamente não (ou pouco) inclusivos. Este autor compreende que o desenvolvimento independente e inovador de acessibilidade digital aos DVs é inerentemente muito mais lento por consistir predominantemente em iniciativas científicas e voluntárias, a despeito da existência de normas de acessibilidade vigentes que norteiam e incentivam o desenvolvimento de acessibilidade, como a Lei Brasileira de Inclusão da Pessoa com Deficiência [6].

Consequentemente, no atual estado de insuficiência de soluções de acessibilidade, há um elevado nível de incompatibilidade entre deficiências visuais e os diversos artefatos digitais, inclusive em ambientes educacionais formais. Não obstante, as incompatibilidades apenas tenderão a aumentar com o passar do tempo, pois o desenvolvimento constante de computadores e *gadgets* é uma das maiores forças da economia global, na qual a acessi-

bilidade não é contemplada com a importância que lhe é devida, em grande parte devido aos baixos incentivos de capital financeiro privado.

Isto gera um aprofundamento crescente da desigualdade de inclusão digital entre a população enxergante e a população de DVs, estes estruturalmente excluídos das possibilidades de desenvolverem suas próprias soluções digitais. Tampouco esse grupo marginalizado é contemplado com quantidades suficientes de financiamento de pesquisa, desenvolvimento e inovação para solucionar a forma em que esse problema está configurado estruturalmente nessa população marginalizada.

1.2 Relato do autor de experiência prévia de aprendizagem de programação com um aluno cego

A origem da ideia da pesquisa aqui relatada é uma experiência anterior em estágio curricular de Ensino em Computação, em que este autor tinha o objetivo de ensinar Desenvolvimento Front-End para três turmas de estudantes de ensino médio no Centro de Ensino Médio 01 do Paranoá, no Distrito Federal. Um dos alunos contemplados, o qual será referido aqui por aluno E.J, é deficiente visual de nascença, leigo em computação, e capaz apenas de enxergar uma quantidade mínima de luminosidade, insuficiente para conseguir ler e escrever com sistemas visuais de grafia dentro do planejamento das aulas do curso proposto. Ademais, E.J. é egresso da instituição de ensino especial em que a parte de campo desta pesquisa foi efetuada, onde ele recebeu alfabetização em sistema Braille.

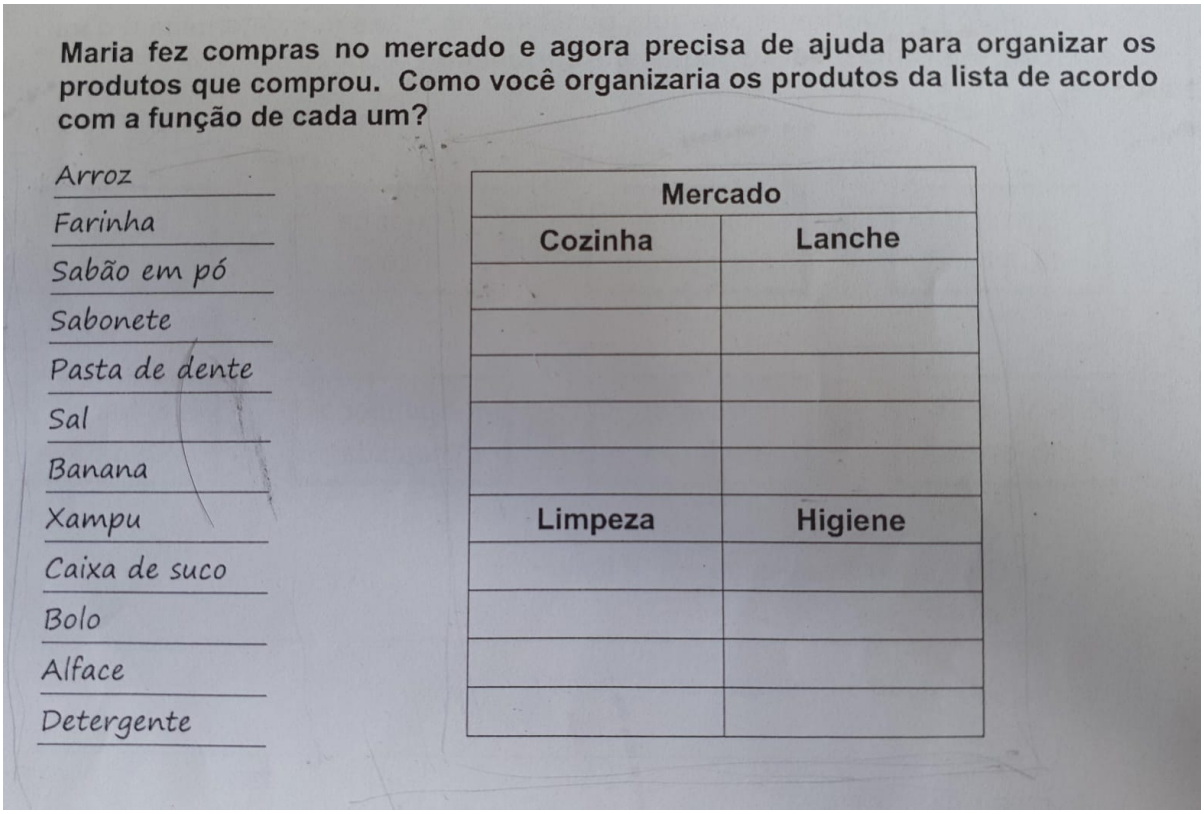


Figura 1.1: Exercício que compreende parte da avaliação diagnóstica aplicada ao aluno E.J..

O autor e seu colega regente aplicaram como avaliação diagnóstica uma lista de exercícios manuscritos de Computação Desplugada da Universidade de Santa Cruz do Sul (UNISC) [7] . Para fins de investigação exploratória, o autor convidou o aluno E.J. a fazer a avaliação diagnóstica com o auxílio de seu tutor, que lia e escrevia em sistemas visuais de grafia para ele. Um exercício em particular da lista, mostrado na Figura 1.1 produziu a observação mais relevante sobre as nuances da cegueira sobre a computação.

```python
mercado = {
    "produto1" : {"nome" : "arroz", "função" : "cozinha"},
    "produto2" : {"nome" : "farinha", "função" : "cozinha"},
    "produto3" : {"nome" : "sabão em pó", "função" : "limpeza"},
    "produto4" : {"nome" : "sabonete", "função" : "higiene"},
    "produto5" : {"nome" : "pasta de dente", "função" : "higiene"},
    "produto6" : {"nome" : "sal", "função" : "cozinha"},
    "produto7" : {"nome" : "banana", "função" : "lanche"},
    "produto8" : {"nome" : "xampu", "função" : "higiene"},
    "produto9" : {"nome" : "caixa de suco", "função" : "lanche"},
    "produto10" : {"nome" : "bolo", "função" : "lanche"},
    "produto11" : {"nome" : "alface", "função" : "cozinha"},
    "produto12" : {"nome" : "detergente", "função" : "limpeza"}
}
```

Figura 1.2: Código em linguagem de programação Python3 que representa a estrutura de dados manipulada no exercício da Figura 1.2 pelo aluno E.J. e seu tutor durante a avaliação diagnóstica.

Na Figura 1.1, temos uma atividade de organizar produtos de uma lista de compras domésticas de acordo com categorias representadas em uma matriz de quatro dimensões: Cozinha, Lanche, Limpeza e Higiene. Em termos computacionais, esta atividade consiste em transpor um vetor para um dicionário aninhado em linguagem de programação Python3 [8], conforme a figura Figura 1.2.

Observando o aluno E.J. e seu tutor resolvendo este problema, considerando que ambos eram completamente leigos em computação teórica, o desempenho conjunto foi excepcional. O tutor explicou claramente o problema e os dados ao aluno E.J., que alocou cada produto na categoria correta assim que lhe era anunciado com precisão e agilidade.

A solução deste problema não implica que o aluno E.J. aprendeu o conceito de vetores e matrizes como estruturas de dados, mas demonstrou que um indivíduo cego e leigo em computação pode ter a capacidade de operar sobre elas com saídas e entradas de dados digitais, caso os dados sejam representados e processados com adaptações de grafia não visuais. Ou seja, a questão não seria se DVs são capazes de tal, e sim se os designers de interfaces digitais são capazes de tornar isso possível.

Buscando incluir o aluno E.J. na prática de ensino em computação com um plano de aprendizagem individual, este autor elaborou uma proposta de ensinar algoritmos por meio de fluxogramas tangíveis em papel por impressão Braille. Os fluxogramas Braille consistiam em cópias fiéis de fluxogramas em impressão de tinta, substituindo a percepção visual de escrita e formas pela percepção tátil.

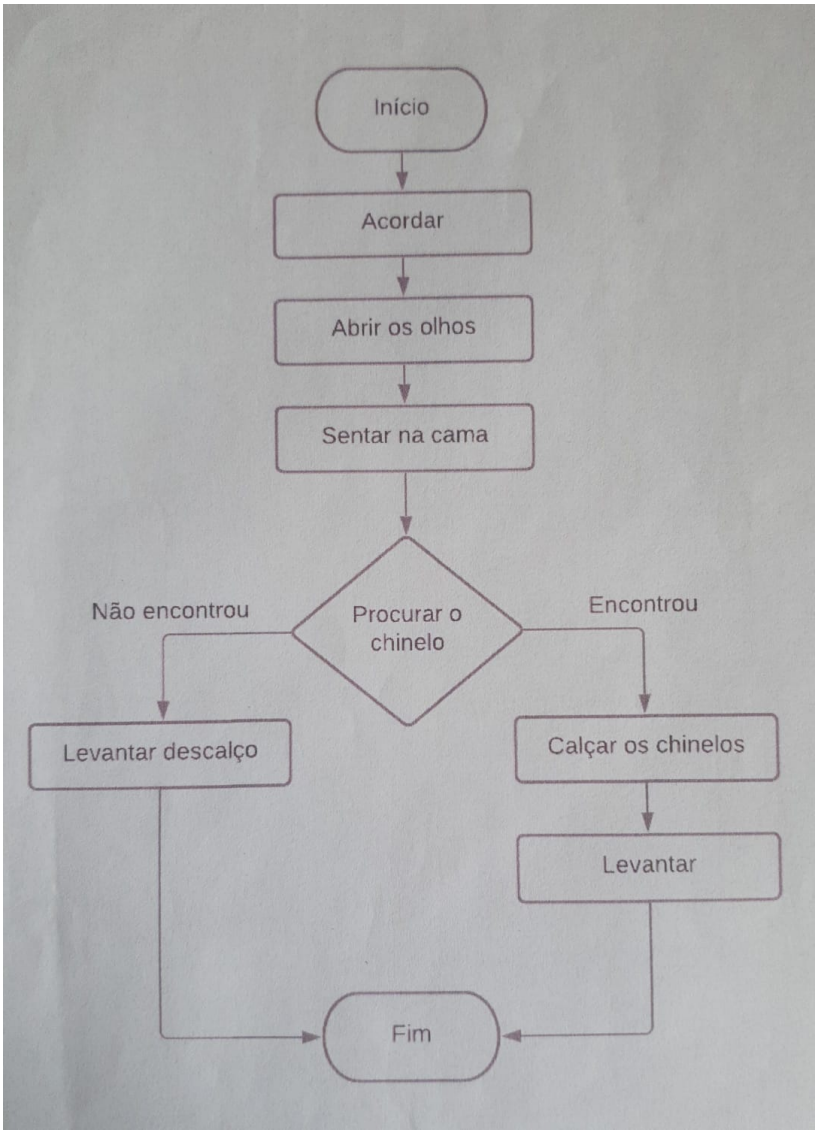


Figura 1.3: Fluxograma em impressão de tinta para enxergantes sobre ações e decisões a serem tomadas ao acordar.

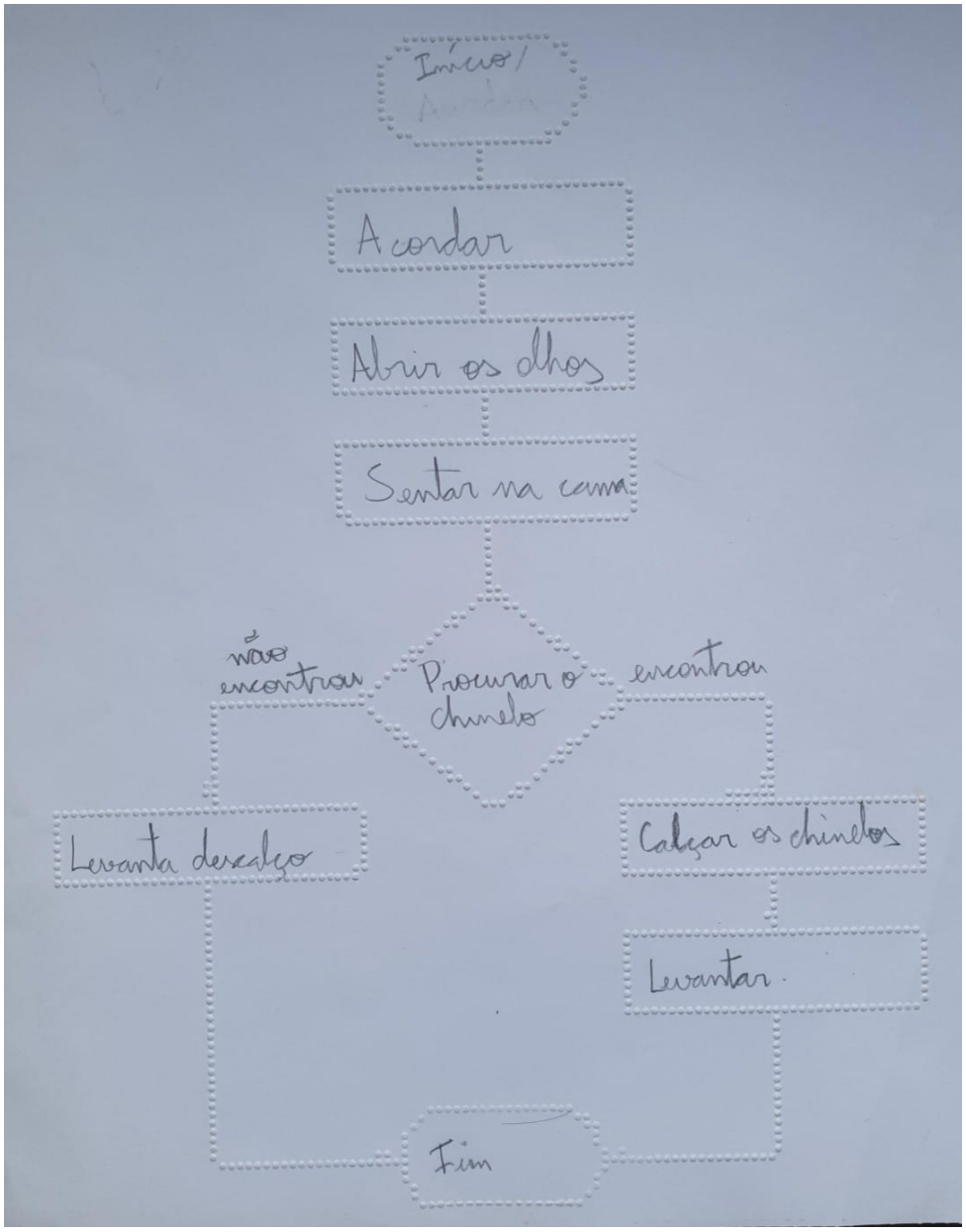


Figura 1.4: Versão Braille do fluxograma em impressão de tinta da Figura 1.3. A escrita foi feita com lápis pelo autor para o aluno E.J. por falta de espaço para o uso de punção de escrita Braille.

A Figura 1.3 mostra um dos fluxogramas usados em impressão em tinta sobre ações e decisões a serem comumente realizadas ao acordar. A Figura 1.4 mostra a versão braille do fluxograma em impressão de tinta da Figura 1.3. Após a explicação dos conceitos sobre fluxogramas e a leitura oral do fluxograma da Figura 1.3, o aluno E.J. percorreu manualmente e sequencialmente cada forma do fluxograma Braille e enunciou todas as informações a serem preenchidas pelo autor de forma manuscrita.

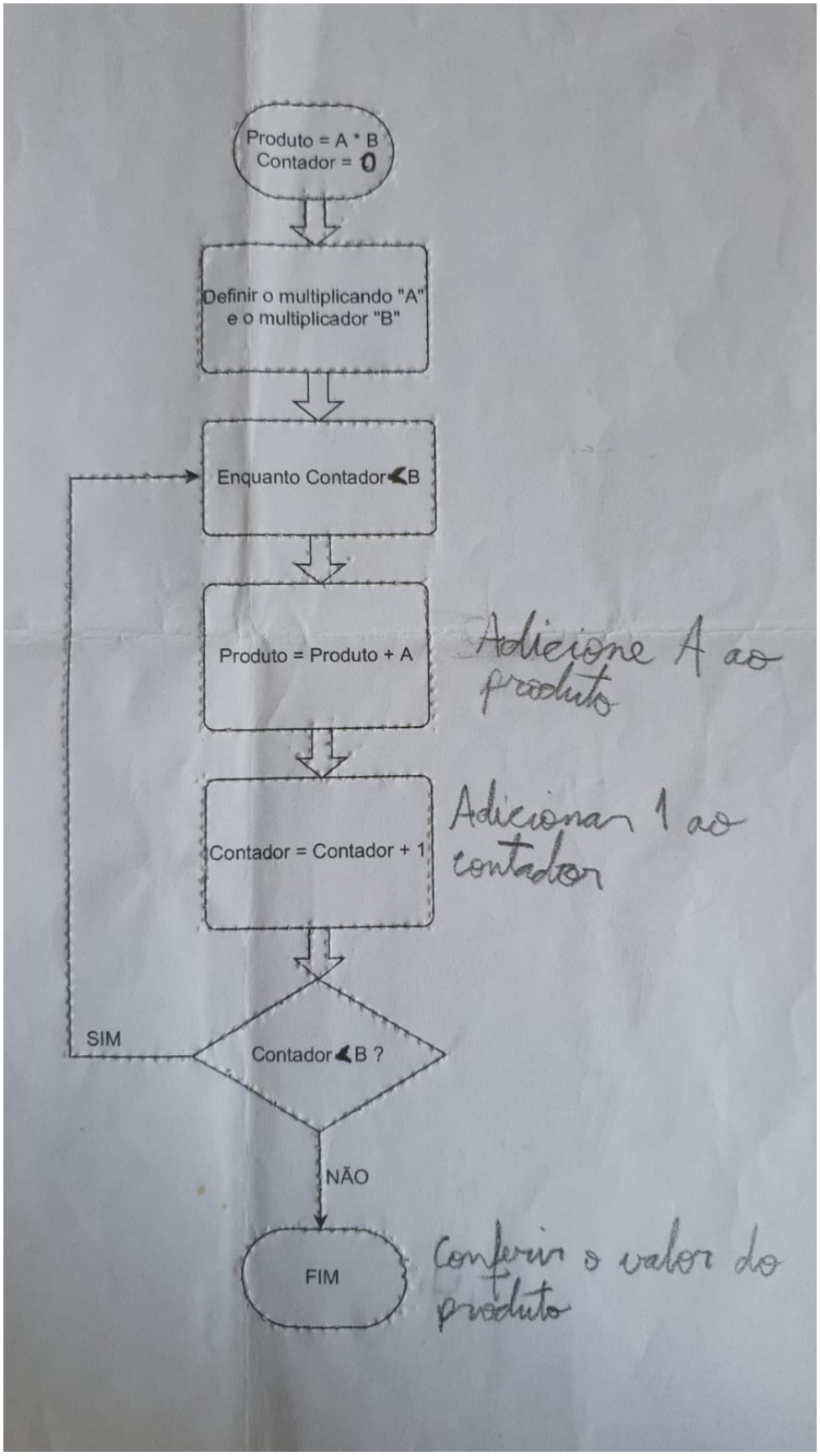


Figura 1.5: Fluxograma com formas em Braille com algoritmo de multiplicação por somas repetidas de um número por ele mesmo.



Figura 1.6: Brinquedo sensorial Pop-It! usado em conjunto com o fluxograma da Figura 1.5 como tabela de variáveis e representar tangivelmente a execução do algoritmo de multiplicação.

Esse resultado encorajou a avançar na exploração com um fluxograma de multiplicação por repetição de somas. O autor levou junto ao fluxograma um brinquedo *Pop-It!*, tal como o exemplo mostrado na Capítulo 1.6 para que ele servisse como uma tabela de variáveis editável e tangível de uso tátil, com cada fileira representando um dos componentes da multiplicação de dois algarismos (multiplicando, multiplicador e produto), além do contador de somas efetuadas. Cada protuberância do *Pop-It!* é utilizada por meio de pressão, que faz com que a superfície torne-se côncava ou convexa. Dessa forma, a quantidade de protuberâncias côncavas em cada fileira foi usada para registrar dinamicamente e tangivelmente o valor de cada variável envolvida no algoritmo de multiplicação na Figura 1.5.

```
1 def multiplicar(a, b):
2     produto = 0
3     for contador in range(b):
4         produto += a
5     return(produto)
```

Figura 1.7: Código Python de uma função que executa uma multiplicação de dois números de acordo com a lógica do algoritmo do fluxograma na Figura 1.5.

Na Figura 1.7, temos um código na linguagem de programação Python3 de uma função que executa uma multiplicação de dois números de acordo com a lógica empregada pelo algoritmo pelo fluxograma utilizado pelo aluno E.J.: somar o multiplicando por si, registrar o resultado da soma no produto, incrementar o contador e repetir a soma até que o contador atinja o multiplicador.

O resultado desta segunda prática também foi positivo: o aluno E.J. declarou que entendeu conceitualmente a multiplicação como operação matemática, o que nunca havia conseguido anteriormente, e que a experiência foi divertida. Ainda que nessa prática didático-pedagógica com o aluno E.J. não tenha sido realizada uma avaliação de aprendizagem para coletar dados mais aprofundados, ela futuramente serviu ao autor de referência para a pesquisa científica a ser descrita e discutida no restante desta monografia.

1.3 Centro de Ensino Especial para Deficientes Visuais de Brasília

A exploração em aprendizagem de programação com E.J. teve continuidade por uma pesquisa científica conduzida no Centro de Ensino Especial para Deficientes Visuais (CEEDV) de Brasília, uma instituição de referência nacional em inovação no ensino para DVs [**?**], na qual este autor realizou a pesquisa pela modalidade de Estágio Curricular Não Obrigatório do curso de Licenciatura em Computação da Universidade de Brasília (UnB).

O Centro de Ensino Especial para Deficientes Visuais (CEEDV) é a única instituição educacional do Distrito Federal e Entorno especializada em atendimento educacional para cegos, surdocegos e baixa visão. Em atividade desde 1966, seu trabalho é de referência nacional, o qual busca adaptar currículos de ensino regular para seus alunos DVs, os quais compreendiam 395 matrículas em 2021, com as devidas adaptações e procedimentos metodológicos, inclusive com o uso de equipamentos e materiais didáticos específicos às demandas de cada estudante e em conformidade com normas e diretrizes educacionais vigentes. A instituição declara que sua função social é atender pessoas DVs de todas as idades para promover a formação de cidadãos plenos, promover inclusão e respeito à diversidade.

Dentro da organicidade do Centro de Ensino Especial para Deficientes Visuais (CEEDV), o estágio foi conduzido como uma atividade do Centro de Apoio às Pessoas com Deficiência Visual (CAP), uma unidade de serviços cuja atribuição é providenciar apoio pedagógico e suplementação didática à Rede Pública de Ensino do Distrito Federal. Além dos estudantes deste grupo que são seu público-alvo, demais grupos beneficiados pelo Centro de Apoio às Pessoas com Deficiência Visual (CAP) são estudantes DVs matriculados da Educação Infantil até a Educação Tecnológica e Superior, professores regentes da

Rede Pública de Ensino do DF e parceiros do Centro de Ensino Especial para Deficientes Visuais (CEEDV), e demais pessoas da comunidade de DVs.

O quadro de pessoal do Centro de Apoio às Pessoas com Deficiência Visual (CAP) envolvidos na pesquisa de campo como *stakeholders* do projeto é composto por:

- **Adaptadores**: professores da Carreira Magistério Público do DF, exercendo cargos de servidores com a atribuição de adaptar materiais didático-pedagógicos para a grafia do sistema Braille e demais grafias não-visuais;
- **Revisores**: servidores responsáveis por revisar materiais em Braille produzidos para fins de controle de qualidade.

O adaptador e os dois revisores envolvidos na pesquisa de campo serão aqui respectivamente referenciados por Prof. H., revisor A. e revisora E.

A pesquisa de campo também contou com a colaboração de alunos do Prof. H.:

- Aluno G.: faixa etária entre 20 a 30 anos, nascido com cegueira completa, altamente proficiente com simbologia Braille e egresso da Universidade de Brasília (UnB) como bacharel em Direito, estudante de concursos públicos.
- Aluno A.: faixa etária entre 14 a 18 anos, acometido por cegueira completa, proficiente com simbologia Braille e estudante do terceiro ano do Ensino Médio.
- Aluno R.: faixa etária entre 14 a 18 anos, acometido por baixa visão que possibilita leitura somente por estreitos feixes de luz à poucos centímetros de distância aos olhos, sem domínio em simbologia Braille, estudante do terceiro ano do Ensino Médio e colega do aluno A..
- Aluno I.: faixa etária entre 20 a 30 anos, baixa visão que limita leitura à poucos centímetros de distância aos olhos por trajetórias lateralizadas, pouca proficiência com simbologia Braille, estudante de concursos públicos.

Estudos anteriores, tratados no Capítulo 4, já exploraram a criação de materiais didáticos e métodos pedagógicos para o ensino de lógica de programação e algoritmos em salas de aula. Tais resultados experimentais demonstram que o raciocínio lógico de algoritmos e programação de computadores é sim acessível aos DVs desde que sejam empregadas formas alternativas à visão de perceber e representar informações.

1.4 Definição do objeto de estudo

Para delimitar o alcance de possíveis abordagem ao problema do cenário descrito, seguiremos pela análise de duas abordagens distintas a serem descritas e discutidas nas duas subseções a seguir, e fazer a escolha de uma delas.

1.4.1 Computação Plugada

A Computação Plugada requer o uso de computadores, desenvolvimento e implementação de recursos de acessibilidade digitais em um ambiente digital de programação, como um Ambiente de Desenvolvimento Integrado. O *stakeholder* sugeriu a integração do leitor de tela NonVisual Desktop Access [9] a um ambiente digital de ensino de programação já existente. No entanto, essa abordagem é frágil porque põe o computador como material didático imprescindível que precisa ter esse software instalado e configurado por um enxergante para torná-lo acessível a DVs.

Esse pré-requisito à pratica educacional impõe a dependência de DVs por enxergantes, indo à contramão da autonomia dos DVs. Ademais, essa restrição põe competências de uso de leitores de tela como um requisito obrigatório ao ensino e aprendizagem de algoritmos para DVs em ambas partes docente e discente por um motivo estritamente tecnológico, sem relação didática direta.

Seguir pelo ensino em computação plugada implicaria em um esforço intenso de desenvolvimento de sistemas digitais para o autor e o desenvolvimento de um produto que o corpo técnico do Centro de Ensino Especial para Deficientes Visuais (CEEDV) não seria capaz de assumir com independência e autonomia posteriormente à realização da pesquisa. Dada a disponibilidade limitada de recursos materiais, financeiros e de tempo à realização da pesquisa deste estudo, optou-se por seguir para o caminho que dispensa o uso de recursos digitais como material didático.

1.4.2 Computação Desplugada

Enquanto há a computação plugada, em que as tecnologias digitais envolvidas trazem muitas limitações ao ensino de algoritmos para DVs, existe também a computação desplugada, em que o raciocínio é exercido primariamente pelo esforço mental e pela manipulação física de objetos, contrapondo ao emprego de máquinas computadoras, como calculadoras e computadores digitais de uso geral com seus dispositivos periféricos de entrada e saída de dados (telas, teclado, mouse etc.) [10] [11].

Essa abordagem é inclusiva tanto às numerosas escolas brasileiras que atualmente não detêm recursos informáticos digitais suficientes para a realização de práticas didático-pedagógicas em computação plugada, como também aos DVs por empregar recursos didáticos exclusivamente tangíveis já conhecidos pelo corpo discente e docente deste grupo de educação especial. Ademais, diminuir barreiras de produção, implementação e manutenção da solução facilita a transferência do conhecimento produzido por este autor para o Centro de Ensino Especial para Deficientes Visuais (CEEDV) e demais interessados.

1.5 Objetivos da pesquisa

1.5.1 Objetivo geral

- Estabelecer a plausibilidade da aprendizagem de algoritmos e lógica de programação por DVs leigos.

1.5.2 Objetivos específicos

- Projetar um linguagem de programação manipulável e programável por DVs.
- Levantar requisitos de design tangível sobre algoritmos e programação por computação desplugada para DVs.
- Explorar o potencial da ludicidade como contexto de execução tangível de código programável sendo uma abordagem de programação por computação desplugada para DVs.

1.6 Estrutura da monografia

O restante deste documento é composto por sete capítulos. Segue a relação de assuntos abordados por cada um:

- Capítulo 2: A metodologia científica escolhida pelo autor deste estudo para estruturar a pesquisa.
- Capítulo 3: Análise dos impactos e potencialidades da solução desenvolvida pelo autor.
- Capítulo 4: Discussão da base de conhecimento científico sobre aprendizagem, selecionada pelo autor para fundamentar a pesquisa e esta tese.
- Capítulo 5: Apresentação da base de conhecimento científico pré-existente sobre o tema de estudo tratado neste capítulo.
- Capítulo 6: Descrição dos processos iterativos de pesquisa e desenvolvimento do design da solução criada pelo autor.
- Capítulo 7: Julgamentos do autor sobre os resultados relatados nesta tese.
- Capítulo 8: Orientações para continuidade do projeto desenvolvido pelo autor, e demais estudos posteriores que usem este como referência para projetos de design.

Capítulo 2

Metodologia

Este segundo capítulo apresenta a metodologia científica escolhida para orientar o projeto e posteriormente discute como ela é usada para estruturar a pesquisa e o desenvolvimento da solução em um sistema unificado.

O autor utilizou o raciocínio indutivo ao partir de observações específicas e experiências concretas, como a prática didática prévia com o aluno E.J. , para inferir generalizações e formular as "Conjecturas Teóricas"que nortearam o projeto. A metodologia principal selecionada foi a *Design Science Research* (DSR), como descrita no artigo "Design science research: fazendo pesquisas científicas rigorosas atreladas ao desenvolvimento de artefatos computacionais projetados para a educação"[1], porque o objetivo da pesquisa era duplo: resolver um problema prático através da criação de um artefato (os blocos de programação) e, ao mesmo tempo, gerar conhecimento científico. Portanto, a DSR foi escolhida por ser a metodologia que permitia pegar essas inferências, obtidas indutivamente do específico para o geral, e usá-las ativamente para fundamentar o design, a implementação e a avaliação da solução tecnológica.

2.1 Contexto e problema

O contexto apresentado no Capítulo 1 claramente requer o desenvolvimento de uma tecnologia de ensino de algoritmos com pedagogia inclusiva para DVs. Metodologias de pesquisa clássicas das ciências humanas, como Estudo de Caso e Etnografia, não empregam a criação e implementação de tecnologias como ferramenta de experimentação e testagem de hipóteses. Já o desenvolvimento de uma tecnologia simplesmente direcionado a resolver o problema de acessibilidade no ensino de algoritmos e lógica de programação resulta em produtos proprietários de seu criador original; ou de efetividade limitada por apenas mitigar o problema mencionado em vez de buscar solucioná-lo. Em ambos os casos, o *stakeholder* não terá uma solução concreta porque ela requer que o beneficiado tenha

domínio de uma tecnologia suficiente para solucionar o problema pela transformação do contexto com autonomia.

Este cenário é determinante ao problema da falta de recursos para o ensino de algoritmos para DVs. Não se trata apenas de insuficiência teórica em pedagogia especial ou em prática de produção de material didático inclusivo, mas de ambos os *déficits* integrados aos percalços da educação especial para DVs no Brasil.

2.2 Artefato

A computação é composta por uma vasta variedade de tecnologias, dos quais hardware e software são os mais óbvios. Porém, todos esses produtos advêm e dependem de criações intermediárias e abstratas, como protocolos de comunicação, frameworks e linguagens de programação. Tecnologias digitais, tangíveis ou não, são artefatos conforme a seguinte definição:

> "(...) algo projetado, um engenho, uma artificialidade (...), projetado para resolver algum problema (o objetivo do artefato), num dado contexto, a partir de conhecimentos e conjecturas sobre o mundo (natural e social). A partir do uso do artefato, é possível avaliar se parecem válidas as conjecturas teóricas que levaram o designer a desenvolver o artefato daquela maneira."[1, p. 5-6]

O desenvolvimento do artefato na *Design Science Research* (DSR) é estruturado em um conjunto cíclico de atividades voltadas aos processos de design tecnológico, denominado por Ciclo de Engenharia e composto por:

1. "Investigação do problema;
2. Design da solução;
3. Validação da solução;
4. Implementação da solução;
5. Avaliação da implantação."[1, p. 7]

O tipo de artefato desenvolvido é um princípio de projeto, ou seja, "princípios-chave e conceitos para guiar o projeto"[1, p. 5]. O projeto refere-se a um método de ensino de algoritmos à DVs no Brasil. Este é um objetivo de longo prazo que envolve uma pesquisa de grande escala, com coleta de dados em vários grupos distintos de DVs e descritos ao longo dos processos sociais de aprendizagem.

Portanto, o artefato deste estudo restringe-se a uma prova de conceito deste projeto, com o objetivo de estabelecer se DVs são capazes de assimilar raciocínio computacional após adquiridos os prejuízos sobre a capacidade de enxergar. Para elucidar esta hipótese,

o artefato deste autor é direcionado por duas questões sobre comportamento humano de DVs por meio de um artefato para resolver o problema em questão.

2.3 Pergunta de Pesquisa e Hipóteses de Trabalho

Pondo os desafios apresentados em uma questão objetiva, o problema abordado pela pesquisa desse trabalho é:

> Há possibilidade de ensinar algoritmos a DVs de forma que eles assimilem e apliquem o conhecimento para construir e executar programas tangíveis?

No decorrer do projeto do artefato, foram repetidamente constatados pelos *stakeholders* do Centro de Ensino Especial para Deficientes Visuais (CEEDV) erros de design, para os quais foram desenvolvidas e implementadas soluções para melhorar a experiência em manusear os protótipos de material didático até atingir uma versão satisfatória suficiente para uso com alunos da instituição de ensino.

O design do artefato baseou-se em conhecimentos prévios sobre o ensino de lógica de programação para deficientes visuais que evidenciam a importância de aspectos sociais nas práticas de ensino de algoritmos a DVs, como trabalho em equipe e compartilhamento de produtos concretos de atividades em aula. Essas vivências propiciaram um ambiente altamente engajador, a transferência de conhecimento entre alunos e até mesmo cooperação para superar dificuldades pessoais únicas causadas por deficiências visuais. [12] [13]

Com esses resultados orientando uma pesquisa exploratória, foi escolhida uma abordagem gamificada para que aspectos materiais e conceituais levantados pela concepção e uso dos materiais didáticos sejam elucidados e explorados com mais facilidade. Experiências lúdicas são mais intuitivas e ágeis para o desenvolvimento de pesquisa didático-pedagógica em comparação a experimentos científicos rigidamente estruturados. Para tal estratégia, o artefato foi escolhido como blocos tangíveis de programação para obter-se representações táteis de pseudocódigo que reproduzam tangivelmente tanto a sintaxe básica quanto a estrutura textual típicas da programação de computadores. Isso foi um dos princípios base de design para que a proposta didático-pedagógica seja não somente inclusiva, mas também libertadora por não causar segregação pelo desenvolvimento de um tipo de programação exclusivo para cegos. A significância da ludicidade para esta pesquisa é discutida em maiores detalhes no Capítulo 4.

2.4 Conjecturas Teóricas

> Na *Design Science Research* (DSR), o pesquisador está comprometido com dois objetivos:

1. resolver um problema prático num contexto específico por meio de um artefato e
2. gerar novo conhecimento científico. [1, p. 6]

"

O primeiro objetivo implica na criação de uma tecnologia para resolver um problema. A distinção do desenvolvimento tecnológico pela *Design Science Research* (DSR) para resolver um problema específico em comparação a abordagens de engenharia é a finalidade científica do design e implementação da tecnologia. Para cumprir o segundo objetivo, uma solução desenvolvida pela *Design Science Research* (DSR) tem como objetivo testar hipóteses por meio da implementação e refinamento do artefato projetado para produzir conhecimento científico.

Assim como no desenvolvimento do artefato, o desenvolvimento da pesquisa científica é um conjunto cíclico de atividades, denominado por Ciclo Empírico, que consistem em:

1. "Contexto da pesquisa;
2. Análise do problema de pesquisa;
3. Pesquisa e inferência do design;
4. Validação;
5. Execução da pesquisa;
6. Análise de dados;
7. Contribuição da pesquisa"[1, p. 8]

Portanto, na *Design Science Research* (DSR) as conjecturas teóricas compõem a unidade de análise do artefato sobre como ele molda o comportamento humano, indo além de aspectos técnicos para como uma tecnologia constrói a identidade e as ações de pessoas e suas organizações.

2.5 Modelo cíclico-iterativo e avaliações da Design Science Research

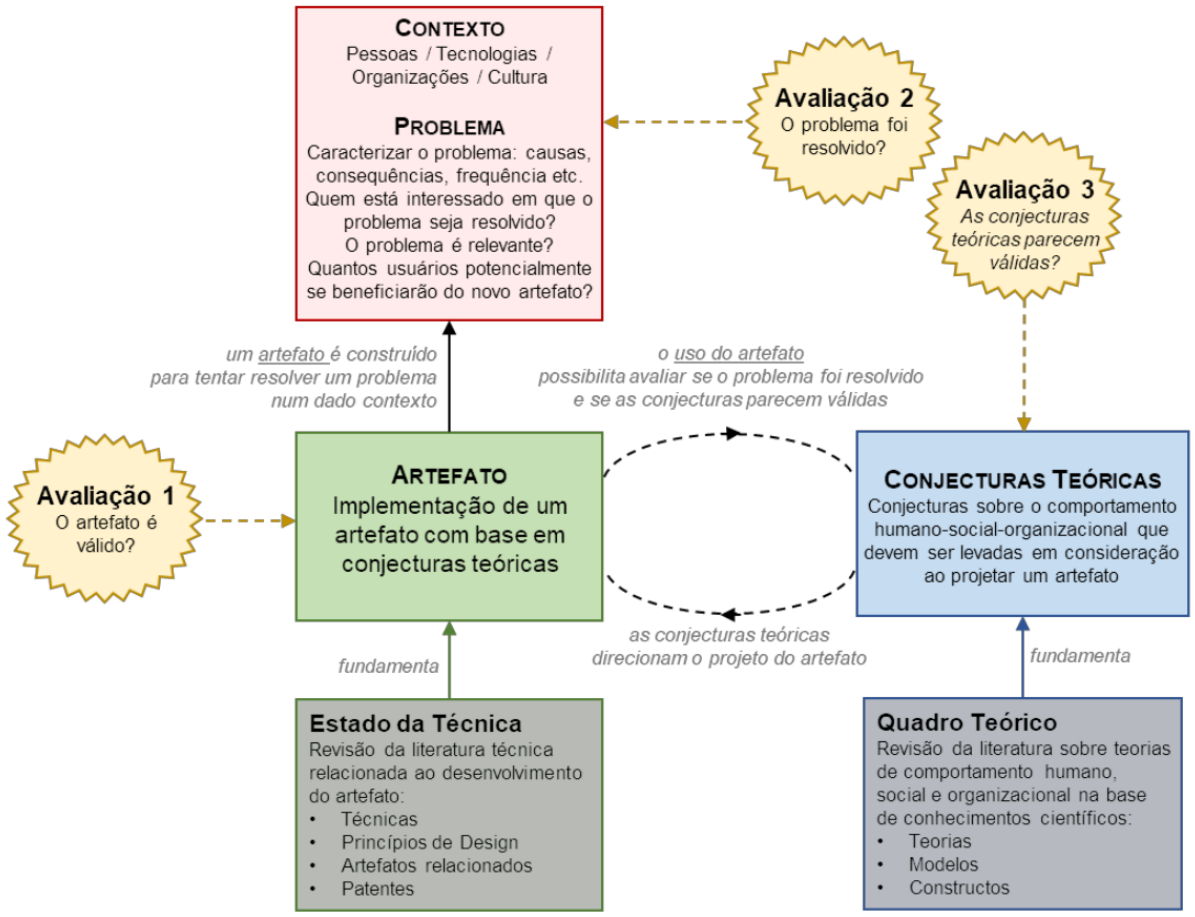


Figura 2.1: Fluxograma da estrutura do DSR, contendo suas unidades de análise e as relações entre elas. Fonte: Página 10 de [1].

A Figura 2.1 é um esquema da metodologia Design Science aplicada às pesquisas científicas em Computação, em que "o desenvolvimento de sistemas computacionais e de outros artefatos tecnológicos, por si só, não se caracteriza como uma pesquisa científica"[1, p. 3]. Enquanto abordagens convencionais de engenharia são definidas apenas por um contexto problemático e uma ferramenta tecnológica desenvolvida para como solução, o DSR busca inovação científica em conjunto com resolução de problemas práticos como frutos de um único projeto.

A relação entre conjecturas teóricas e o artefato é ciclicamente iterativa, ou seja, no decorrer de uma pesquisa nesta metodologia, teoria e técnica alternam-se entre etapas de pesquisa de campo e revisão de literatura para refinar as conjecturas e o artefato até que

sejam respondidas três avaliações sobre o trabalho científico, demonstradas nos balões de cor amarela na Figura 2.1.

2.6 Avaliação 1: O artefato é válido?

A validade do artefato consiste em "avaliar se o sistema projetado satisfaz os requisitos, tanto funcionais como não funcionais"[1, p. 9]. Ou seja, o artefato desta pesquisa precisa possibilitar o ensino e aprendizagem de algoritmos a DVs com construção e execução de programas tangíveis, tudo conforme os critérios de usabilidade inclusiva dos *stakeholders* envolvidos.

2.7 Avaliação 2: O problema foi resolvido?

A resolução do problema depende da avaliação positiva do "uso do artefato proposto no contexto para o qual foi desenvolvido". Ou seja, essa aprovação é um critério do *stakeholder* sobre a aceitação e viabilidade de adoção do artefato pelo seu público-alvo. O sistema pode funcionar na teoria enquanto não se aplica à prática, tal como os Tapetes Musicais que cumpriu todos os requisitos porém "os professores das escolas mostram-se hesitantes em adotá-lo por sua instalação lhes parecer complexa e demandar muito tempo"[1, p. 20]. Portanto, para que o artefato resolva o problema chave dessa pesquisa, precisa ser entregue em um estado que possa ser apropriado pelo *stakeholder* e professores do CEEDV em suas próprias atividades sem intervenção do pesquisador.

2.8 Avaliação 3: As conjecturas teóricas parecem válidas?

A validade das conjecturas teóricas depende do estudo ter valor para além do *stakeholder* pelas descobertas da pesquisa ou o próprio artefato serem úteis a outros contextos semelhantes. Ou seja, o artefato dessa pesquisa precisa servir de um ponto de partida a teorias de aprendizagem, e não se limitar à uma tecnologia específica para o CEEDV.

O próximo capítulo trata das conjecturas teóricas que fundamentam a pesquisa e desenvolvimento do artefato.

Capítulo 3

Conjecturas Teóricas (Sobre Blocos Manipuláveis)

Este capítulo trata das questões motivadoras do estudo com maior aprofundamento, trazendo premissas que norteiam a pesquisa e o desenvolvimento de uma solução para o ensino de algoritmos e lógica de programação para DVs na forma de Conjecturas Teóricas, conforme a metodologia de *Design Science Research* (DSR).

Discussão sobre limitações da experiência prévia de ensino de algoritmos com o aluno E.J

3.1 A execução tangível de um algoritmo em pseudocódigo por computação desplugada possibilita que DVs leigos sejam capazes de aprender a programar?

Conforme discutido no Capítulo 1 e demonstrada na Figura 1.2, o aluno E.J demonstrou ter capacidade de usar dados estruturados representados por Computação Desplugada com relação direta a estruturas de dados em linguagem de programação Python. Todavia, o uso de fluxogramas limitou a aprendizagem de E.J a algoritmos, sendo inacessível a progressão do ensino à construção e execução autônoma de programas.

O que é escrito em grafia Braille não pode ser apagado e nem reescrito de sua mídia física. Caso E.J cometesse um erro ao preencher o fluxograma Braille usando uma punção no papel, seria necessária a reprodução do fluxograma em uma folha nova. Esse empecilho é incompatível com o ensino de programação porque tornaria um código em tipologia Braille impossível de reconfigurar mantendo-se o mesmo ambiente de programação. Por-

tanto, a mídia do algoritmo tangível precisa garantir que ele seja reprogramável ao invés de exigir a produção de outro algoritmo.

Fluxogramas não são escaláveis para algoritmos extensos, principalmente com a presença de encadeamento de estruturas de decisão e repetição. Um kit de formas e linhas manipuláveis em simbologia Braille para a produção de fluxogramas acarretaria uma grande quantidade de formas e linhas interligadas que rapidamente formam um emaranhado lógico difícil de ler e executar. Essa limitação é agravada por deficiências visuais porque o tato não permite ler um fluxograma construído por peças dinâmicas sem acarretar no risco de alterar o conteúdo durante a manipulação; e intersecções de linhas independentes são serem diferenciáveis de uma linha que divide-se em mais de um caminho.

Somente com o uso de um artefato, externo ao algoritmo em fluxograma Braille de multiplicação por repetição de somas, para representar tangivelmente a execução passo-a-passo, que os conceitos teóricos da computação de variáveis, estruturas de decisão e repetição foram devidamente explicados e entendidos pelo aluno E.J. Por meio do brinquedo sensorial Pop-It!, E.J pôde executar o algoritmo no fluxograma de forma livremente programável, usando de quaisquer algarismos para fazer o cálculo mental uma multiplicação de tabuada, inclusive permitindo erros de manipulação de variáveis durante a computação.

Daí, este autor inferiu que DVs precisam que a execução de algoritmos seja tangível por artefatos manipuláveis para que o raciocínio algorítmico lhes seja acessível. Essa descoberta é a base da primeira conjectura teórica deste trabalho, a qual orienta e valida a parte tangível do design do artefato: a construção concreta da tecnologia e como os conceitos teóricos se relacionam com os DVs por meio da manipulação do artefato:

> A execução tangível de um algoritmo em pseudocódigo por computação desplugada possibilita que DVs leigos sejam capazes de aprender a programar?

Esse questionamento tem implicações complexas e desafiadoras sobre a adaptação da programação para DVs leigos: Não basta adaptar a execução e manipulação digital de código por ferramentas de acessibilidade como escrita em sistema Braille e audiodescrição porque o ato de programar precisa ser tangível e manipulável, e isso inclui a manifestação da existência do código pela execução algorítmica. A relação de DVs com o mundo externo depende de informações sensoriais porque o ambiente precisa ser explorado e conhecido, sem o grande volume de informações que a visão é capaz de providenciar como o principal via sensorial, para que abstrações possam ser assimiladas. Essa peculiaridade enfraquece abordagens digitais para DVs leigos e traz mérito à computação desplugada por esta oferecer mais agência em proporcionar um ambiente de aprendizagem inclusivo.

Portanto, para que um artefato que torne a programação acessível a DVs leigos, ele precisa ser composto de duas partes manipuláveis distintas e interdependentes: código tangível e dispositivo analógico para representar atomicamente cada instrução do código,

ou seja, capaz de representar o efeito de cada instrução atomicamente, sem produzir menos efeitos do que os explícitos e nem mais de forma colateral. Desta forma, DVs podem conseguir estabelecer uma relação de causa e efeito entre um pedaço de código executado e o resultado da execução dele pela tangibilidade deste.

Avaliando as interações com o aluno E.J., o autor também inferiu que essas ferramentas se beneficiam da ludicidade, pois o sentimento de diversão atraiu e motivou E. a querer aprender novos conceitos por meio da exploração tangível de materiais didáticos inéditos. É importante frisar que DVs entendem as funcionalidades de computadores digitais em suas diversas formas (PCs, celulares, tablets, dispositivos inteligentes etc.) e aprendem a agir de acordo para interagir adequadamente conforme o objetivo de uso do dispositivo. Mas a os processos computacionais executados digitalmente lhes são ininteligíveis por incompatibilidades decorrentes da dependência da visão por parte da computação para troca de informações nas interfaces com humanos.

Caso o aluno E.J realmente estava se divertindo durante a dinâmica do fluxograma de multiplicação enquanto executava um algoritmo tangivelmente de forma computacional ao invés de ser inundado por dúvidas, então provavelmente sua limitação sobre a compreensão da existência da computação digital estava sendo rompida. Ou seja, ele sentiu que estava começando a entender as abstrações do mundo digital, pois o algoritmo de multiplicação por repetição de somas é análogo a como um processador digital efetua multiplicações de dois algarismos e é diferente do uso da tabuada para tal, como ele aprendeu a multiplicar por memorização simbólica sem compreender essa operação matemática conceitualmente.

3.2 A ludicidade é capaz de dar significância aos conceitos digitais da computação a fim de promover a aprendizagem significativa de algoritmos em linguagem computacional por DVs?

Essa outra descoberta é a base da segunda conjectura teórica deste trabalho, a qual põe à prova a validade social do artefato, ou seja, se ele é ou não capaz de resolver o problema, discutido no Capítulo 2, da ausência de recursos para o ensino de algoritmos para DVs:

> A ludicidade é capaz de dar significância aos conceitos digitais da computação a fim de promover a aprendizagem significativa de algoritmos em linguagem computacional por DVs?

A segunda conjectura teórica põe à prova a validade do artefato: se ele será capaz de resolver o problema explicitado no Capítulo 2: "falta de recursos para o ensino de algoritmos

para DVs". O artefato não somente deve ser capaz de atender aos desafios didáticos postos à prova pela primeira conjectura teórica, mas também precisa ser atrativo o suficiente para ser adotado pelo *stakeholder* por incorporação ao currículo da formação educacional ofertada pelo Centro de Ensino Especial para Deficientes Visuais (CEEDV). Caso contrário, o artefato não estará em um estado suficiente para ser adotado em práticas educacionais formais mesmo tendo sucesso em contribuir com descobertas sobre ensino de algoritmos para DVs para a comunidade científica e educadores interessados.

Capítulo 4

Quadro Teórico

Este capítulo trata das teorias científicas que embasam as Conjecturas Teóricas apresentadas no Capítulo 3 e são as principais referências para que elas atuem como instrumento de decisão e avaliação do design e da implementação do artefato no decorrer da pesquisa iterativa efetuada pela metodologia *Design Science Research* (DSR).

4.1 Raciocínio Computacional e Computação Desplugada

Um grupo de pesquisadores da Universidade Federal da Bahia (UFBA) conduziu o projeto exploratório de "Prática Interdisciplinar em Atividades de Computação Desplugada na Educação Básica"[11]. A iniciativa consistiu em um breve diagnóstico da realidade da escola pública estadual preceptora, localizada em um bairro da periferia de Salvador, BA, seguidamente de sete intervenções na forma de aulas didáticas de computação desplugada sobre temas elementares variados dentro da Ciência da Computação.

O objetivo do projeto foi explorar o uso da computação desplugada como método de ensino de Computação na Educação Básica. O ponto de investigação dessa iniciativa é o raciocínio computacional: "...a capacidade de resolução de problemas de forma sistemática, usando dedução e abstração, habilidades muito bem trabalhadas na Computação"[11, p. 2]. O principal fundamento teórico do projeto é a definição da Computação Desplugada como "...alternativa para a execução de atividades que estimulam o raciocínio computacional sem o uso de computadores ou quaisquer outros recursos eletroeletrônicos"[11, p. 2].

Para alavancar a Computação Desplugada como método para exercitar o raciocínio computacional, o projeto utiliza a interdisciplinaridade para abordar os conceitos da Computação de forma acessível ao público-alvo do projeto. Esse uso estratégico da interdisci-

plinaridade com o público-alvo foi definido pelo grupo como "uma postura do educador na integração de práticas e conhecimentos, seja para a construção de novos conhecimentos, como para o desenvolvimento de habilidades e manutenção de competências"[11, p. 3].

Ou seja, conhecimentos de áreas afins à Computação foram usados como subsunçores para introduzir conceitos como algoritmos, lógica proposicional e compactação de arquivos de forma acessível a um público leigo [11, p. 4]. Os resultados do projeto demonstram o mérito do uso da Computação Desplugada: os docentes da escola preceptora foram atendidos na demanda identificada por "necessidade e compreensão de como funciona a computação por parte dos alunos e deles também"[11, p. 8]. Por meio de uma pedagogia acessível, interdisciplinar e sócio-interacionista que colocou o público-alvo como protagonista, o grupo de pesquisadores tiveram sucesso em produzir uma aprendizagem crítica e enriquecedora dentro do currículo educacional da Computação.

Ainda que o projeto tenha sido voltado a um público-alvo enxergante, os fundamentos teóricos de Computação Desplugada e Raciocínio Computacional corroboram fortemente com a conjectura teórica que trata dos artifícios tangíveis a serem projetados para ensinar algoritmos a DVs. Já as estratégias empregadas nas sete intervenções realizadas no projeto corroboram com a conjectura teórica que trata da ludicidade como a forma de dar significância ao ensino contemporâneo de Computação, pois praticamente todas as intervenções empregaram sistematicamente a ludicidade tanto para cativar os participantes, como também para representar conceitos e regras computacionais de forma acessível.

Essa visão construtivista de aprendizagem e ensino como uma forma de construção social do ser que norteou esse grupo de pesquisa tem muitas proximidades com uma teoria da aprendizagem de um reverenciado contribuidor da pedagogia científica, tratada na próxima subseção.

4.2 Teoria Sociocultural de Lev Vygotsky

Lev Vygotsky foi um psicólogo soviético, conhecido por suas contribuições científicas sobre desenvolvimento psicológico em crianças. Em sua Teoria Sociocultural, Vygotsky entende o desenvolvimento humano como um processo histórico e social: as funções psicológicas superiores emergem de interações concretas entre o indivíduo e o meio social em que ele participa.

Como evidenciado pelo grupo de pesquisadores da UFBA, a necessidade de perceber e interagir com o ambiente, incentivada pelas intervenções de Computação Desplugada, ativou as funções responsáveis pela atenção, memória e linguagem para possibilitar a assimilação de conceitos da Computação. Ou seja, a falta de relação direta de situações-problema com a Computação não foram um empecilho, mas uma forma de integrar o

mundo da Computação ao meio social em que os participantes vivem para proporcionar uma aprendizagem significativa e acessível.

Para Vygotsky, essas interações entre indivíduo e ambiente são mediadas por instrumentos (artefatos, no contexto tecnológico desta tese) e signos (símbolos pictográficos, alfabéticos ou fonéticos). Aqui, a linguagem tem um papel de destaque, pois ela integra instrumentos e signos em um sistema histórico-cultural, dotado de regras, alfabeto e métodos para materializar informações elaboradas por processos cognitivos e transportá-las através do espaço e do tempo por meio de registros e mídias.

Pela Computação ser um sistema lógico-matemático de signos, regras e métodos, o Raciocínio Computacional age como ponte entre mundo digital e a aprendizagem de Computação. Através da transcrição sistemática entre as linguagens humanas e computacionais, o Raciocínio Computacional é um poderoso fundamento didático-pedagógico para trazer o socio-construtivismo ao ensino da Computação.

Ademais, o conceito de Zona de Desenvolvimento Proximal define uma área limiar de habilidades em que um indivíduo não tem domínio, mas é capaz de executar com auxílio de outros, principalmente com apoio capacitado de professores e demais profissionais de ensino ou artefatos para mediar essa relação com o ambiente. "Isso mostra a necessidade da mediação de um parceiro mais experiente, que mostre a ela a função social dos objetos materiais e imateriais, de maneira a transmitir os significados dos objetos do mundo da cultura. A história humana se constrói nessa perspectiva, com a transmissão das aquisições da cultura humana para as gerações que se sucedem umas às outras"[14, p. 4].

Finalmente, Vygotsky enfatiza a significância do jogar para o desenvolvimento infantil, pois são nas brincadeiras em grupo que crianças são imersas em sistemas simbólicos estruturados por um sistema de regras, em que a rigidez encontra-se na formação, ou quebra, de confiança nas relações interpessoais que contextualizam o ato de brincar.

A ludicidade é um aspecto comum tanto na experiência relatada no uso de Computação Desplugada pelo grupo de pesquisadores da UFBA, como também para a teoria de Lev Vygotsky. Em ambos os casos, ela é determinante para o sucesso deles por desempenhar papéis centrais no engajamento de pessoas em torno de uma atividade em comum dentro de um contexto social. Portanto, é necessário entendê-la não somente como diversão, mas como uma função social primordial, conforme a próxima subseção.

4.3 Homo Ludens

Johan Huizinga foi um historiador e linguista holandês. Em seu livro "Homo Ludens: o jogo como elemento da cultura", a prática de jogar é extensivamente definida e analisada como um fenômeno cultural. [15] Para esta discussão, é pertinente apenas o primeiro

capítulo, que se inicia estabelecendo que o ato de "jogar é mais antigo que cultura"e "até em suas formas mais simples entre animais, (...) é mais do que um fenômeno fisiológico ou reflexo psicológico".

Huizinga elenca quatro características fundamentais do ato de jogar:

1. "Conscientemente separado da vida ordinária sendo frívolo, mas simultaneamente absorvendo o indivíduo intensamente e completamente"
2. "Uma atividade desconectada à interesses materiais, da qual nenhum lucro é obtido"
3. "Procede com as próprias fronteiras de tempo e espaço de acordo com um sistema fixo e organizado de regras"
4. "Promove a formação de agrupamentos sociais que tendem a (...) enfatizar diferenças em relação à realidade mundana por disfarce ou demais formas de fantasias"

Portanto, jogar é historicamente a gênese de comportamentos pró-sociais entre seres sencientes, ou seja, capazes de experienciar estados emocionais idiossincraticamente de forma a compor memória e identidade pessoais. Foi através da necessidade por diversão que nossos ancestrais tiveram estímulos criativos necessários para manipular a matéria da natureza e dos próprios corpos para razões além da sobrevivência, impulsionando o desenvolvimento de linguagem e atividades lúdicas mediadas por tecnologias criadas como brinquedos, instrumentos musicais, meios de comunicação e sistemas linguísticos de fala e escrita. Destas práticas, forma-se coesão social, cultura e, consequentemente, sociedades.

Todas as quatro características sociais do ato de jogar foram exploradas nas intervenções do grupo de pesquisadores da UFBA [11, p. 4]. As atividades foram inusitadas, apresentando desafios triviais de forma desafiadora pelo uso de vocabulário, regras e símbolos de acordo com o Raciocínio Computacional. Claramente não houve qualquer interesse material envolvido no projeto. Devido à abordagem computacional de ideias e conceitos, as delimitações de tempo e espaço durante as intervenções foram trabalhadas de forma diferente do convencional por meio da interdisciplinaridade. Ou seja, estratégias convencionais oriundas de raciocínios monodisciplinares para interagir com o meio físico, dissociados da Computação, estavam fora das regras e o uso delas implicariam em uma violação da ordem computacional do ambiente lúdico e criado nas intervenções. Finalmente, as intervenções foram pró-sociais: promoveram experiências únicas e enriquecedoras entre todos os participantes da comunidade escolar atendida.

Para estudantes cegos, o jogo com uso de computação desplugada tem ainda mais potencial de promover inovações em ensino de algoritmos. Tecnologias digitais, tanto computadores tangíveis como suas abstrações esquemáticas como autômatos e estruturas de dados, são fundamentalmente visuais e adaptá-las a ponto de que deficientes visuais

sejam capazes de desempenhar computações mentalmente é difícil a ponto de ser inviável haver uma metodologia padrão para tal replicável em massa. A execução tangível de código como um jogo proporciona um ambiente de aprendizagem em que DVs estão imersos em um conjunto novo de entidades, símbolos e regras distintas da realidade rígida, baseada em previsibilidade e calculismo, à qual estão habituados. Isso abre espaço para que o DVs se sujeite a cenários inusitados e imprevisíveis, necessários para a aprendizagem proposta pelo artefato.

Porém, este estudo busca explorar a viabilidade de uma linguagem de programação manipulável como um modelo de deficientes visuais poderem ensinar programação a outros deficientes visuais. A ideia dos blocos tangíveis em LEGO é proporcionar um modelo pedagógico que, através do jogar, possa trabalhar interdisciplinarmente cenários diversos que a imaginação permitir construir com blocos de vocabulários diversos, assim como outros modelos tangíveis e interativos além do labirinto para concretizar a execução de algoritmos manipuláveis.

Ao invés de enxergantes projetarem uma linguagem de programação com design universal para um universo infinitamente discreto de peculiaridades de deficiências visuais, este autor enxergante tem como visão para o artefato o desenvolvimento de um design específico ao CEEDV com a finalidade de ser livremente adaptado e modificado pelos próprios deficientes visuais com maior qualidade e eficiência por apropriação cultural. Afinal, o jogo é uma forma de culturalmente sincretizar simbolismos de forma socio-construtivista pelo uso de linguagem.

Estabelecida a ligação entre a ludicidade e os elementos de design de linguagem de programação tangíveis, ainda é necessário definir como operacionalizar este relacionamento de saberes para que sejam possíveis desenvolvimentos futuros de outras versões de linguagens de programação tangíveis a partir do protótipo em LEGO.

Capítulo 5

Estado da Técnica

Este capítulo aborda pesquisas científicas conduzidas por outros pesquisadores, com similaridades de contexto e problema apresentados nos capítulos anteriores e que resultaram na produção de um artefato voltado ao ensino de algoritmos e lógica de programação para DVs.

Junto à descrição dos trabalhos anteriores, também será discutida a relação deles com a base de conhecimento teórico abordada no Capítulo 4.

5.1 Aprendendo a programar de forma geral

5.1.1 Block-Based Programming

A aprendizagem de programação frequentemente está associada a um considerável gasto inicial de tempo para apenas assimilar conceitos básicos e aplicá-los em tarefas simples. Para reduzir a abstração e simplificar regras de sintaxe intrínsecas às linguagens de programação, temos a Programação Baseada em Blocos (PBB) como recurso educacional. Voltado para crianças e adolescentes, ambientes como Blockly e Scratch são usados para o ensino introdutório de algoritmos e lógica de programação de uma forma lúdica, interativa e cativante.

5.1.2 Blockly

Blockly [16] é uma biblioteca da linguagem de programação *JavaScript* desenvolvida pela *Google for Education* que permite construir interfaces visuais de programação em blocos a uma aplicação web. Os blocos correspondem a variáveis ou palavras-chave do *JavaScript*, encaixam-se lateralmente para formar código em linhas horizontais, e verticalmente para compor algoritmos. Desta forma, a edição do código é feita com ações de arrastar e soltar com o mouse, significativamente mais simples em comparação à digitação. Isto

é importante para estudantes em idade escolar, que precisam de estímulos psicomotores para desenvolver a escrita e atingir o letramento.

Enquanto quase todos os estímulos em uma IDE resumem-se a digitar no teclado, a construção de código no *Blockly* é operada e percebida de forma com mais estímulos sensoriais por *feedback* audiovisual e movimentos com o mouse coordenados com a visão. Este feedback instantâneo menos abstrações produzido reduz barreiras e é proporciona uma aprendizagem acessível e engajadora em ambientes escolares.

5.1.3 Scratch

Scratch é um ambiente de Programação Baseada em Blocos (PBB) derivado do *Blockly*, desenvolvido e livremente distribuído pelo *Massachusetts' Institute of Technology*. Mundialmente popular entre escolas como recurso educacional introdutório de algoritmos e programação de computadores, seu principal apelo é a criação de animações e minijogos com uma grande variedade de recursos digitais audiovisuais interativos.

5.1.4 AlgoBlock

AlgoBlock é uma linguagem de programação como artefato tangível. Seu fundamento teórico é a cognição situada: "as ações de um indivíduo são interativamente alcançadas através de um relacionamento com o mundo exterior". [13] Essa conexão entre humano e ambiente é construída e mediada por artefatos, que são projetados e utilizados com propósito determinado pelo contexto entre indivíduo e exterior.

O design do *AlgoBlock* enfatiza a importância de artefatos educacionais serem ferramentas abertas: recursos tangíveis cujo uso é diretamente observável e aberto a intervenções ativas por meio de cognição localizada. Ou seja, uma ferramenta aberta cria uma coesão entre seus usuários pela atração dos mecanismos de percepção e raciocínio em torno da aplicação de sua função, sendo crucial a coerência entre os envolvidos para o sucesso das atividades. Em contextos didático-pedagógicos, uma ferramenta aberta em sala de aula como material didático tem o poder de formar uma comunidade de aprendizes, pois a aprendizagem ativa coletivizada por intermédio de artefatos é um processo de aculturamento.

Crianças de 12 anos de idade brincaram com os *AlgoBlocks* em um experimento. Com a supervisão de monitores, elas tinham à disposição blocos metálicos com símbolos mnemônicos de um dos cinco significados: "*GO-FORWARD, GO-RIGHT, GO-LEFT, ROTATE, LEG FOLD*". Por meio da combinação deles através de conectores eletrônicos e um console como terminal de processamento, as crianças controlaram um submarino em um jogo eletrônico exibido em um monitor CRT. Por meio de trabalho em equipe, elas foram

capazes de criar algoritmos iterativamente, inclusive com o uso de depuração de erros, para conseguirem fazer o submarino navegar sem acidentes por um caminho com início e fim. Conversas e linguagem corporal das crianças participantes durante o experimento demonstraram que a cognição situada é um fundamento poderoso para a aprendizagem significativa de algoritmos, integrando didática a uma pedagogia engajante que promove habilidades motoras e sociais para resolver problemas algorítmicos.

5.2 Aprendendo a programar com deficiência visual

5.2.1 Tangible Block Editor

A PBB é um recurso educacional poderoso para o ensino de algoritmos e lógica de programação. No entanto, o design de interface visual, tanto em predominância em relação ao conjunto de estímulos sensoriais quanto à intensidade deles em quantidade de informação, torna-o incompatível para deficientes visuais. Os blocos do Blockly e Scratch são manipulados pelo mouse ao clicar e arrastar, e adaptar as informações visual-espaciais em áudio-espaciais é impraticável, pois o pixel é uma unidade de medida espacial demasiado reduzida para ser percebida e processada por deficientes visuais.

Uma versão tangível de PBB tem design e execução permeados de desafios de acessibilidade. Primeiramente, a identificação dos blocos tem de ser adaptada para ser percebida por outro sentido além da visão. Não obstante, o meio de execução do código também precisa ser adaptado para ser percebido sem a visão. Se o resultado do código é digital, ele precisa de ferramentas de acessibilidade para que o deficiente visual possa compreendê-lo e analisá-lo. Se for tangível, o artefato necessita de um design ainda mais complexo para ser capaz de representar computação programável de forma desplugada e sem o uso da visão.

Pesquisadores dos EUA desenvolveram um protótipo de editor de PBB para o *Scratch* [17]. O design do artefato consiste em um conjunto de blocos do *Scratch* em versão tangível por impressão 3D , uma base fixadora para os blocos e um kit de visão computacional de *webcam* integrada a um *software* que detecta os blocos tangíveis interconectados na área de edição do código e codifica com audiodescrição o algoritmo construído em uma versão equivalente na interface digital do *Scratch*.

As principais escolhas de design deste artefato, e que servem de referência para demais projetos de artefatos tangíveis de PBB, foram:

- Símbolos mnemônicos para identificar os blocos de acordo com o significado léxico dos blocos digitais do *Scratch*, ao invés do uso de escrita Braille ou em tinta adaptada para baixa-visão.

- Organização do espaço de trabalho entre base fixadora em acrílico para blocos na área de detecção do algoritmo, e paletas para armazenar blocos inativos de forma organizada.
- Cores e texturas específicas para categorias do vocabulário do *Scratch* para identificação visual desses subconjuntos para indivíduos cegos ou com baixa-visão.

5.2.2 Torino

O projeto Torino é uma iniciativa científica de 2020 conduzida na Grã-Bretanha para introduzir uma linguagem de programação tangível a estudantes de 7 a 11 anos [12] . Ele é fundamentado em três princípios da metodologia de Design Inclusivo:

1. "reconhecer exclusão;
2. aprender pela diversidade;
3. projetar para um, estender a vários."

Reconhecer exclusão diz respeito à redefinição do conceito de deficiência como "um problema que surge por incompatibilidade entre tecnologia e o corpo de seu usuário, em vez de uma característica inerente a este"[12, p. 11]. O segundo enfatiza que trabalhar para contornar deficiências específicas a um indivíduo ajuda a criar ideias inovadoras de design por levantamento de requisitos explícitos obscuros aos projetistas como requisitos implícitos. Já o terceiro encoraja desenvolver um design inclusivo que possa ser difundido a outros públicos, ainda que seja prioritário atender consistentemente os requisitos do público-alvo.

O artefato do projeto Torino consistiu em um jogo de cápsulas eletrônicas que conectam-se por cabos de áudio para produzir música. Há uma cápsula de controle central, da qual outras conectam-se para produzir sons, configuráveis em parâmetros físicos diversos como tom, intensidade e duração. Algumas cápsulas tinham a função de formar estruturas de repetição, um conceito fundamental para a formação de melodias, também configuráveis com o número de repetições.

O material foi disponibilizado a escolas britânicas voluntárias com material de treinamento aos docentes envolvidos em atividades de regência com o artefato do projeto Torino. Foram conduzidas aulas, e os resultados demonstraram que as crianças contempladas aprenderam os fundamentos da lógica de programação pela linguagem tangível do Torino. Não obstante, as experiências foram lúdicas, com a maioria engajando-se em produção musical com as cápsulas por uso coletivo.

O projeto Torino demonstra o grande potencial de linguagens de programação tangíveis para promoverem coletividade e criatividade por meio de um artefato com design especificamente responsivo e inclusivo para deficientes visuais.

5.2.3 Outros

Pesquisadores dos EUA tiveram êxito no desenvolvimento experimental de uma ferramenta de acessibilidade especializada ao *Blockly* [18]. Por meio de modificações ao código-fonte *JavaScript*, de licenciamento livre, a interface digital do *Blockly* foi adaptada para ser operada por comandos de teclado, e providenciar *feedback* sonoro por audiodescrição e pistas auditivas de ações como soltar e encaixar blocos.

O uso da audição como principal via de informação pelo artefato é bem mais desafiador para designs tangíveis, pois requer a integração de acústica aos elementos léxicos e psicomotores de blocos tangíveis. No entanto, ela pode ser muito efetiva pela audição ser o sentido mais apurado de um deficiente visual como a visão é o mais apurado para um enxergante.

Capítulo 6

Desenvolvimento do Artefato

Este capítulo aborda, em sete seções, os processos de design e implementação do artefato realizados em conformidade com a *Design Science Research* (DSR) durante a pesquisa de campo na modalidade de estágio curricular em Licenciatura em Computação da Universidade de Brasília (UnB) no acrfullCEEDV como Professor convidado durante o 2º semestre de 2024.

A primeira seção detalha os *stakeholders* e o papel exercido na pesquisa de campo. As seções seguintes descrevem as várias iterações do artefato criadas e as atividades cumpridas para desenvolvê-las.

6.1 Envolvimento de stakeholders

A pesquisa de campo efetuada no CEEDV teve um desenvolvimento conjunto com alguns membros da instituição educacional: Prof. H, enxergante e servidor do setor de apoio pedagógico responsável pelo projeto e produção de recursos didáticos para os alunos, participou junto ao design do artefato; e os revisores E. e A., funcionários cegos que testaram minuciosamente todas as iterações do artefato por serem responsáveis pela validação da usabilidade de materiais didáticos projetados para alunos da instituição.

O desenvolvimento do artefato partiu de um desenvolvimento de protótipos que receberam ajustes iterativamente conforme *feedback* recebido pelos *stakeholders* e alunos voluntários. O ponto de partida foi a apresentação dos labirintos do *Blockly* Games ao Prof.. H., que demonstrou interesse em participar do design de uma versão tangível do jogo.

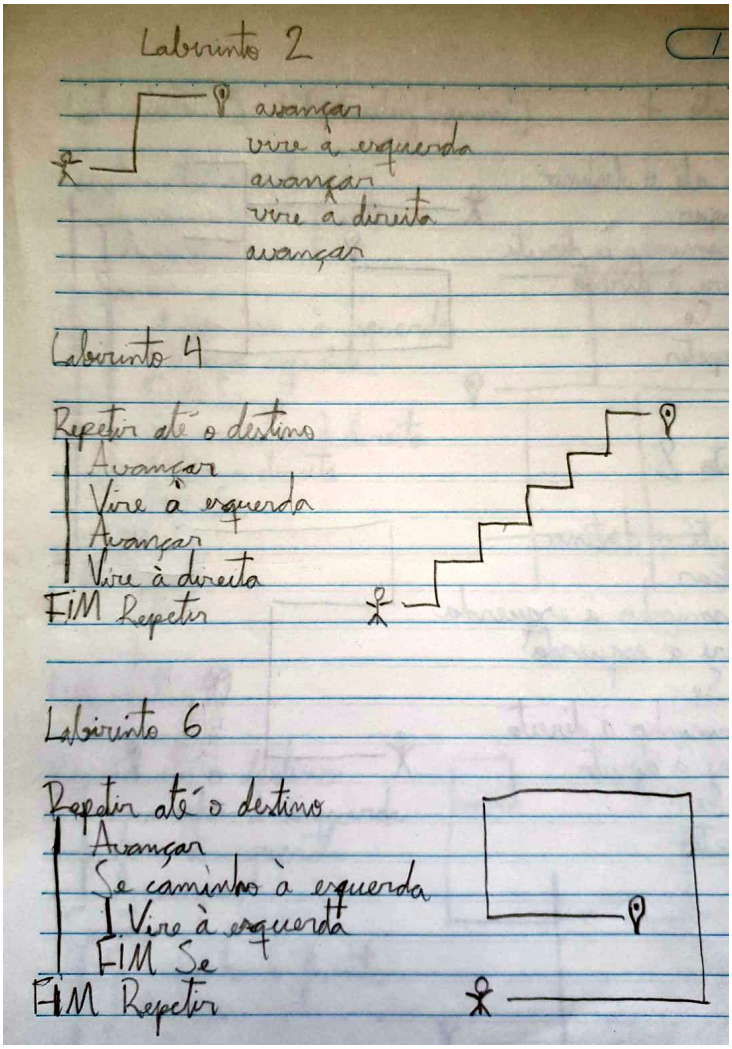


Figura 6.1: Esboço manuscrito de labirintos e algoritmo do *Blockly Games*.

Para demonstrar a ideia visualmente, foi feito um esboço inicial do conceito dos blocos tangíveis com desenhos na Figura 6.1. Ela apresenta três imagens de labirintos e três programas em pseudocódigo que resolvem os labirintos orientando a pessoa do ponto de origem para o de destino.

6.2 Escrita Braille sobreposta à impressão em tinta

O design do artefato, buscando produzir uma versão tangível dos blocos de programação do *Blockly Games*, iniciou-se buscando materiais e técnicas de escrita Braille sobre os blocos tangíveis a fim de identificá-los conforme o vocabulário pré-existente do jogo de labirinto: Avançar, Vire à, Se-Então, Repetir até o destino. O material e método de impressão escolhidos foram folhas de acetato batidas em impressora mecânica Braille. As

palavras-chave escritas em Braille foram recortadas das folhas em acetato e tiveram a qualidade de leitura validadas por *stakeholders* cegos. Desta forma, estas *tags* podem ser coladas sobre blocos tangíveis para identificá-los para cegos.



Figura 6.2: Visão superior do bloco avançar, montado por 3 peças de LEGO e uma tag de acetato e papel com escrita braille fixada por fita dupla face.

No entanto, Prof. H. pôs como requisito que os blocos também sejam identificados para indivíduos com baixa visão por impressão em tinta com fonte Verdana tamanho 18, conforme padrões que utiliza em suas atividades de apoio pedagógico no CEEDV. Como as folhas de acetato utilizadas foram transparentes e a escrita Braille ocupa mais espaço que a em tinta, decidiu-se unir as *tags* em Braille e Verdana tamanho 18 com fita dupla face de fixação forte para produzir uma única *tag* identificadora.

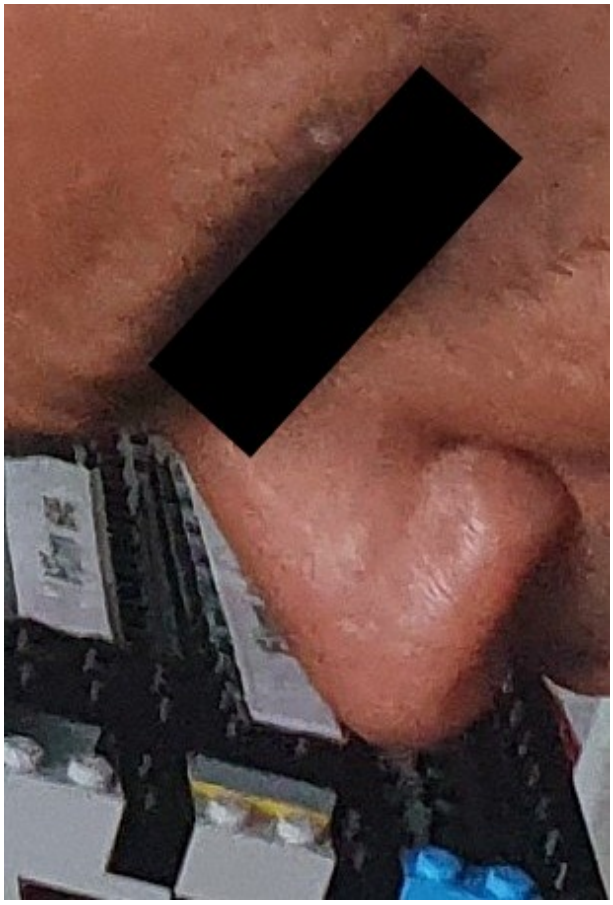


Figura 6.3: Usuário com baixa visão lendo o código tangível construído sobre a base fixadora com o canto externo do olho direito a poucos centímetros de distância da escrita negrito em fonte Verdana tamanho 18.

Para os cegos, um extenso programa inteiro escrito em somente uma linha é perfeitamente legível e o uso de novas linhas é um artifício estranho. Isso foi evidenciado por todos os usuários cegos do artefato, ao conhecê-lo, terem continuado a percorrer as linhas para continuar a leitura mesmo com o dedo encostando na base fixadora, o que indica pelo tato o fim da linha. Ainda assim, os testes de leitura por indivíduos cegos correram bem porque a impressão braille foi considerada legível e com manipulabilidade satisfatória.

No entanto, em testes de leitura visual com indivíduos com baixa visão sem proficiência com braille, os blocos precisavam ser trazidos rente aos olhos em ângulos bem lateralizados. Isto não é um problema para blocos individuais, mas tornou-se um fator limitante para a leitura de blocos organizados em uma composição de um bloco de código tangível, pois a proximidade extrema faz com que percam o referencial de posicionamento do que lê-se em relação ao resto do código.

Na Figura 6.3 pode-se visualizar esse fenômeno: o aluno I. está lendo posicionando

o olho direito a poucos centímetros de distância dos blocos e em um ângulo oblíquo ao invés de olhar diretamente. Essa forma peculiar de leitura deve-se à fisiologia própria da baixa visão do aluno I., o qual só tem visão no olho direito em um estreito feixe no canto externo. Outros indivíduos com baixa visão podem ter outras fisiologias de visão reduzida que levam a outras posições de leitura visual.

6.3 Labirinto em peças de *LEGO*

Para que o artefato tenha um análogo ao labirinto do *Blockly Games* para visualizar a execução do código, foi incluído no design do artefato um labirinto em peças de *LEGO*. O labirinto foi composto por peças de *LEGO* encadeadas de forma a montar um caminho tortuoso sobre uma base plana de *LEGO*, e com peças identificadoras do boneco que percorre o labirinto e a saída dele.

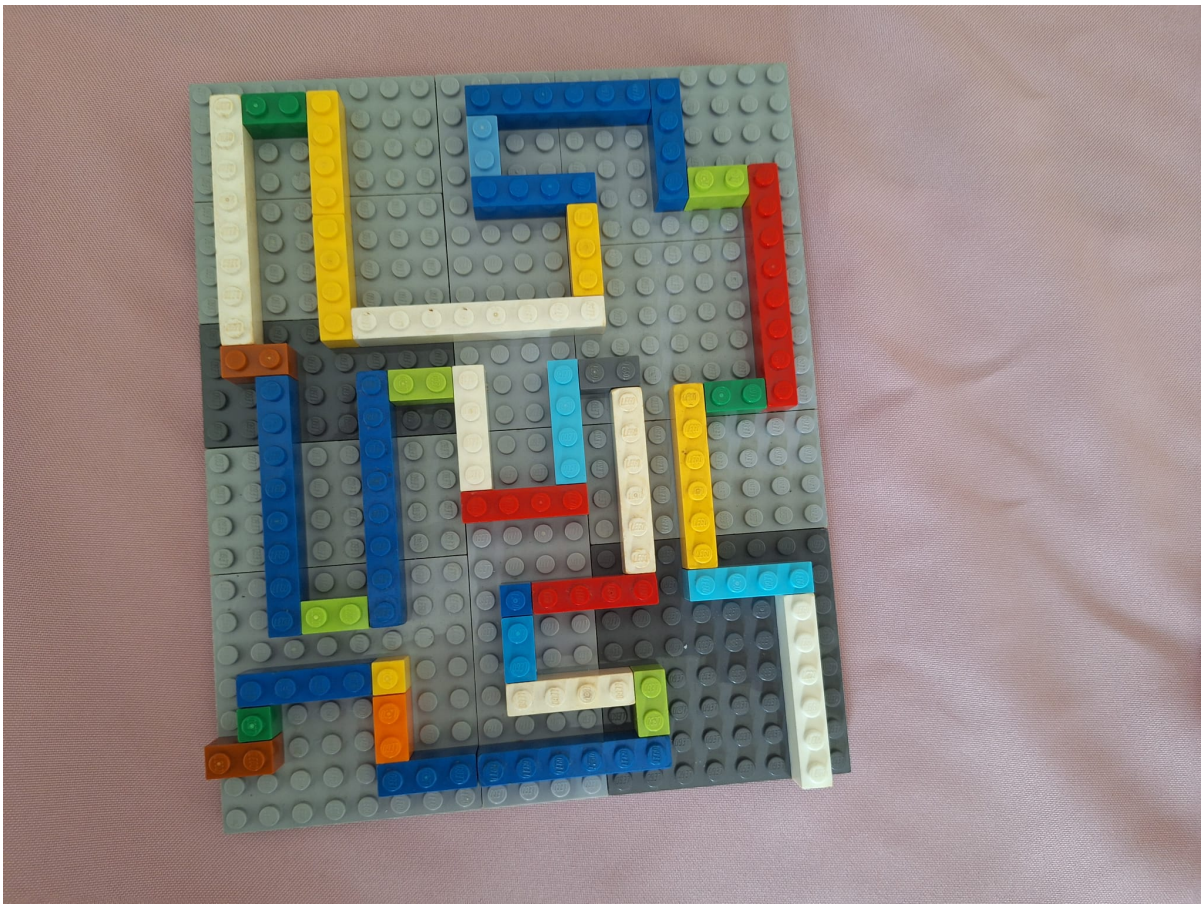


Figura 6.4: Labirinto de *LEGO* com peças de 1 unidade de largura.

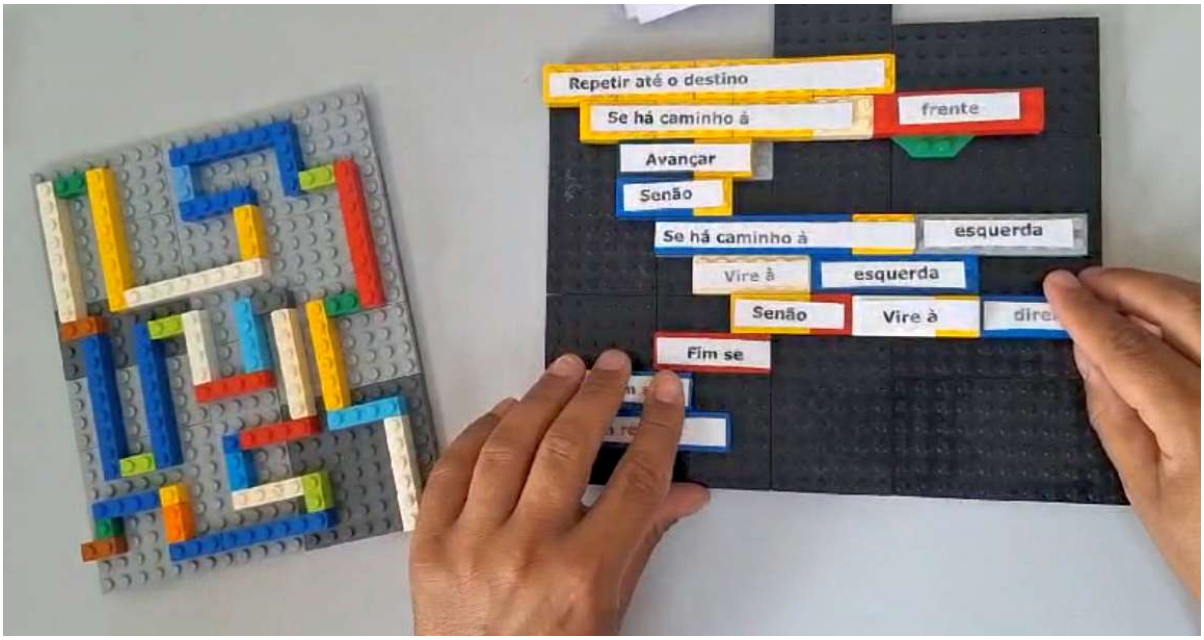


Figura 6.5: Par de elementos do artefato lado a lado: labirinto e algoritmo tangível.

Dessa forma, não somente o usuário deficiente visual pode tangivelmente interpretar o código contido nos blocos montados pelo tato, como é possível também que o artefato seja um material didático para atividades em duplas, em que um atua como o programador manuseando os blocos tangíveis e o outro atua como a intérprete do código movendo o boneco de acordo com os comandos enunciados pelo programador, declarando se cada instrução executada em sequência é ou não válida.

A abordagem de atividade em duplas aumenta a ludicidade do artefato e incentiva a formação de vínculos por proporcionar um jogo com um parceiro, e também diminui a carga cognitiva psicomotora sobre o usuário por ele não precisar estar constantemente mapeando com precisão a localização espacial de ambos conjuntamente em memória de curto prazo.

Ademais, essa dinâmica de duplas também tem muitos paralelos com programação em pares e exercita o raciocínio computacional pela necessidade de criar representações mentais espaciais ou léxicas das informações que o parceiro está obtendo pelo tato e mantê-las sincronizadas conforme a execução desplugada do código.

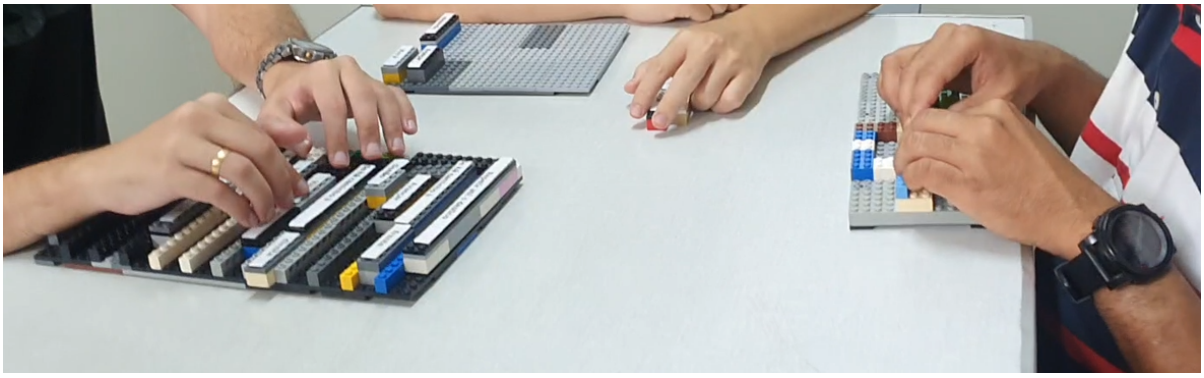


Figura 6.6: Uso do artefato por dupla em pair-programming: programador à esquerda e navegador à direita.

No entanto, o design de construção do labirinto veio com seus desafios e descobertas. Praticamente todo usuário cego teve muita dificuldade em orientação espacial sobre caminhos construídos com peças de LEGO finas de uma unidade de largura. Nesta situação, eles frequentemente perdiam a noção sobre a posição do boneco no labirinto e, consequentemente, o referencial de direita ou esquerda para virá-lo a um dos lados. Observações sobre esses eventos levaram à conjectura de que cegos possuem o tamanho da ponta do próprio dedo indicador como a menor unidade de espaço capaz de ser interpretada pelo tato.

Uma analogia desse fenômeno para uma pessoa enxergante é tentar navegar com os olhos vendados pelos corredores de um labirinto de verdade. Já com labirintos construídos com peças de duas unidades de largura, isso não ocorreu por esta medida ser aproximadamente igual a um dedo indicador, em média.

6.4 Artefatos com velcro

A versão inicial dos blocos tangíveis foi produzida com recortes retangulares de feixes de madeira MDF com velcro na parte inferior, os quais são dispostos, tal como na área de execução dos blocos no Blockly, em uma placa MDF com tiras de velcro fixadas em colunas verticais (Figura 6.7).

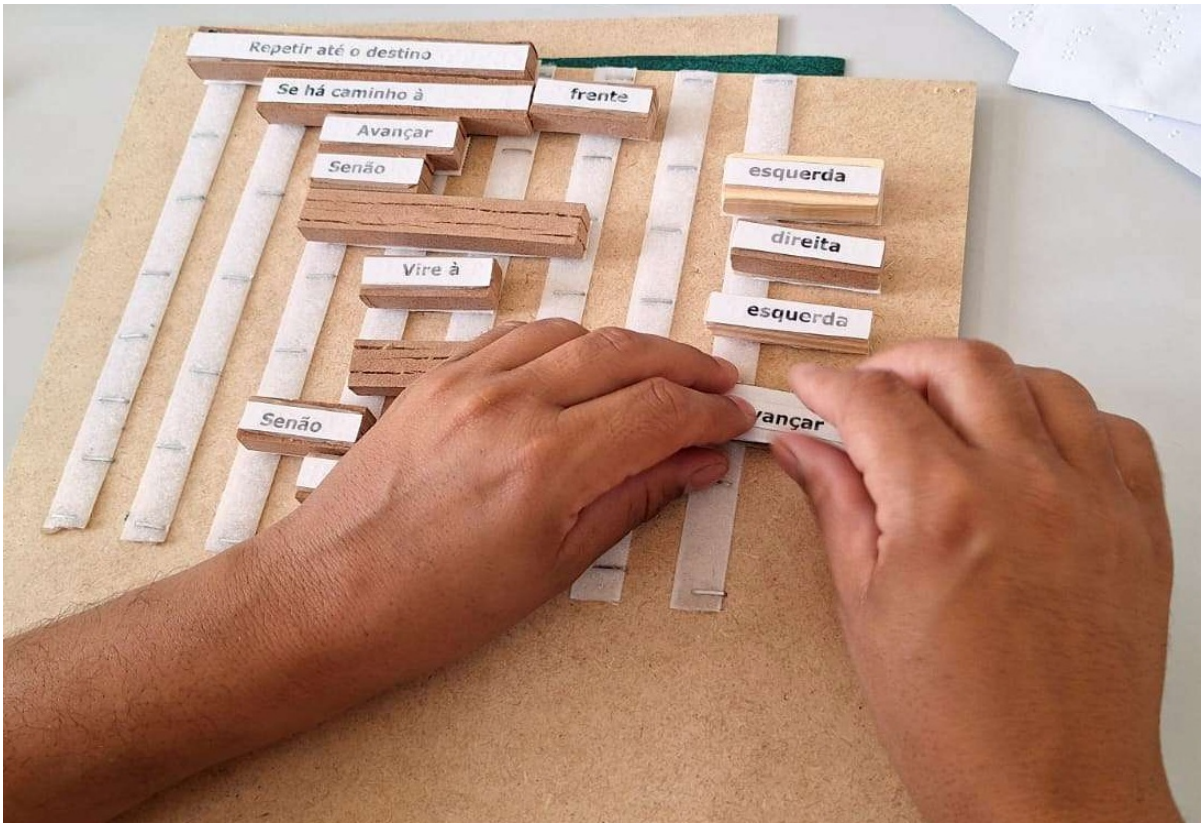


Figura 6.7: Primeira iteração do artefato, com blocos tangíveis em madeira MDF.

As colunas de velcro não cumprem somente papel estrutural, mas também textual, pois cada uma representa um nível de indentação. Dessa forma, a montagem do algoritmo tangível é fiel à estrutura textual de um código virtual para fins de compatibilidade tanto para leitura visual, como também com uma possível transição dos aprendizes em Programação Baseada em Blocos (PBB) para ambientes virtuais de programação.

Para verificar os impactos da largura dos blocos tangíveis sobre a montagem, manuseabilidade e leitura Braille, foram testadas três dimensões: 1 cm, 1,5 cm e 2 cm (Figura 6.8).



Figura 6.8: As três variações em largura dos blocos MDF: 2 cm (superior); 1,5 (meio); 1 cm (inferior).

Testes conduzidos pelos revisores Erika e Alex apontaram que blocos de 1 cm eram demasiadamente finos por escaparem com facilidade entre os dedos e caírem, e prejudicarem fortemente a leitura Braille pelo dedo leitor balançar verticalmente, na direção dos velcros da placa fixadora, além de ser fino demais para usar toda a área tátil disponível para ler. Estes defeitos foram constatados como graves por tornarem o bloco inadequado a usuários cegos. A largura de 1,5 cm não recebeu a avaliação negativa da de 1 cm, sendo constatada como adequada por ambos os revisores, e ideal por Erika. Já a largura de 2 cm também foi igualmente aprovada por ambos os revisores, e preferida por Alex, que tem dedos mais grossos que Erika e por isso se sentiu mais confortável assim.

No entanto, a solução de fixação em velcros foi ultimamente considerada inadequada pelos autores em conjunto com o Prof.. H., por não providenciar auxílios ao posicionamento dos blocos na base fixadora, que precisavam ser fixados com estrutura textual por alinhamento tanto horizontal como vertical, tal como no Blockly Games; assim como a continuidade da superfície de leitura, sendo inviável neste design de velcros para um deficiente visual consistentemente e repetidamente encadear blocos na mesma linha sem lacunas. Em suma, as principais descobertas dessa etapa são que a superfície de lei-

tura tátil dos blocos deve obrigatoriamente ser estável, plana e contínua em todo o plano tridimensional. Portanto, ela não pode balançar para qualquer direção, tampouco ter alterações de relevo.

6.5 Artefatos com fixação em pinos

Com a invalidação da primeira versão do artefato, a revisora Erika apresentou um tabuleiro de xadrez para cegos, o qual tem um design de pinos fixadores para as peças. Cada célula do tabuleiro tem um furo onde a peça encaixa-se por inserção do pino no orifício de uma célula. (Figura 6.9)



Figura 6.9: Tabuleiro de xadrez para cegos, adaptado com fixação das peças em pinos.

Com uma solução em pinos, o encaixe torna-se mais preciso e consistente pela base fixadora dos blocos ter espaçamento estaticamente pré-definido, tanto dentro das linhas como também entre elas. Este modelo de fixação é composto por pinos duplos de LEGO encaixados nos orifícios circulares na parte inferior de blocos LEGO com escrita na parte

de cima, e em blocos furados de uma unidade de largura que compõem uma base fixadora. (Figura 6.10)

Com os blocos tendo 16 milímetros de largura, foi constatado que as placas de suporte para pinos precisam ter uma distância horizontal também de 16 milímetros entre as furações, para que haja espaçamento suficiente para delimitar linhas e agarrar com as pontas dos dedos os blocos entre outros blocos fixados em linhas acima e abaixo. (Figura 6.12)

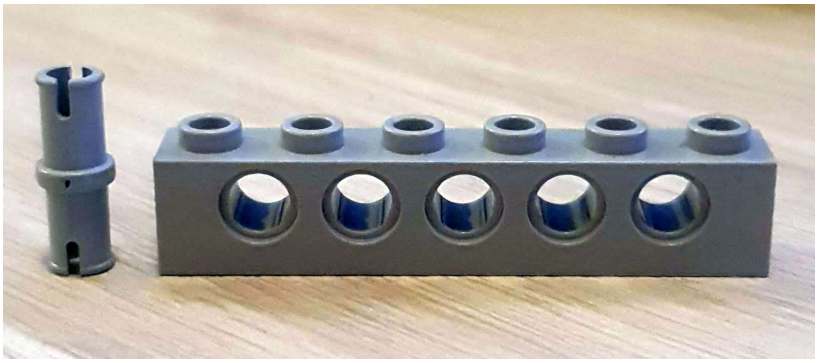


Figura 6.10: Pino duplo de LEGO e bloco com furações laterais.

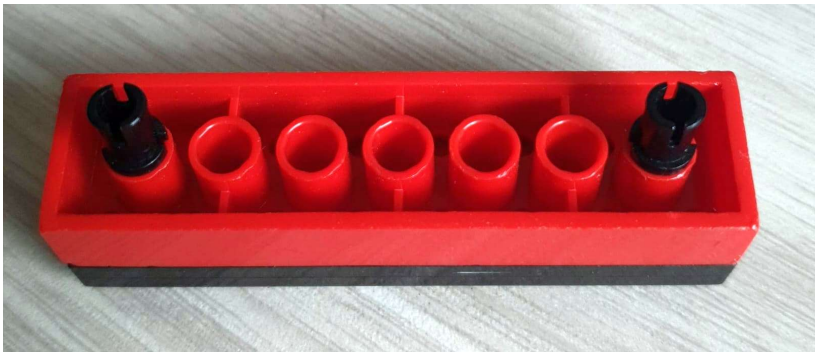


Figura 6.11: Bloco tangível em LEGO com pinos duplos fixados em cada ponta na parte inferior.



Figura 6.12: Blocos em LEGO fixados por pinos duplos na base fixadora de blocos furados, formando três linhas de código tangível.

Este design foi validado com sucesso pelos revisores e possibilitou o início de atividades com alunos do CEEDV selecionados pelo Prof.. H., os quais foram muito receptivos e cooperativos. A estratégia para introduzir o artefato aos alunos foi iniciar apresentando o labirinto em LEGO e deixá-los encontrar o caminho sem o uso de blocos tangíveis, para apenas então introduzir o vocabulário do Blockly e os blocos. Então a prática seguiu progressivamente usando os labirintos já disponíveis no Blockly Games, começando pelos mais simples e indo até os avançados para introduzir conceitos mais complexos como aninhamento de estruturas de decisão e repetição.

No entanto, ao longo de algumas sessões exploratórias com os alunos, o encaixe em pinos mostrou-se uma solução consideravelmente mais atrapalhada do que foi observado com os testes de validação com stakeholders, pois os alunos tiveram mais dificuldades do que o esperado em mirar os pinos com a furação da placa de suporte. Uma possível causa para isso é a maior desenvoltura dos revisores Érika e Alex em processar formas tridimensionais e navegar por essas estruturas com um conjunto de representações mentais e feedback tátil, graças às atividades como revisores de materiais didáticos do CEEDV. Isto

não é estritamente impeditivo para deficientes visuais com menos destreza, mas evidencia que é necessária a aplicação prévia de avaliação diagnóstica e treinamento das habilidades manuais para que seja possível introduzir o artefato, quaisquer que sejam os níveis iniciais de aptidões cognitivas.

É importante também notar que os blocos com furos de LEGO usados nas placas têm diâmetro muito menor que o de um dedo indicador, assim como a distância deles entre si (Figura 6.12). Provavelmente, isso aumentou a dificuldade em encaixar os pinos nas furações corretas para que o texto Braille ficasse contíguo. Constatou-se, portanto, a necessidade de uma nova versão do artefato para reduzir a carga cognitiva psicomotora imposta pelo encaixe de pinos e otimizar o esforço à aprendizagem da lógica de programação em si.

6.6 Artefatos com encaixe guiado por lacunas

Para obter um design com encaixe vertical entre bloco e base com menor carga cognitiva, foi feita uma nova versão do artefato. Ao invés de pinos, o bloco passa a ser encaixado dentro de lacunas na base fixadora, oferecendo uma grande redução da precisão tridimensional necessária para efetuar o encaixe. As lacunas são montadas com peças de LEGO de uma unidade de largura dispostas em linhas horizontais, tal como em folhas de caderno (Figura 6.13).



Figura 6.13: Design de encaixe vertical dos blocos guiado por lacunas.

Esta versão do artefato teve sucesso em trazer praticidade ao uso do artefato, permitindo aos usuários destinarem mais atenção à aprendizagem de programação. Com isso, foi possível realizar atividades proveitosas em dupla com o artefato, porque o papel de programador tornou-se mais fluido, já que com os pinos gastava-se muito tempo no encaixe e o navegador passava por repetidas e longas esperas.

Ademais, este design permitiu a introdução de indentação guiada tangivelmente. Os blocos de "Repetir até o destino"e "Senão-se"formam par com os blocos "FIM REPETIR"e "FIM SE", respectivamente, os quais delimitam o final do subconjunto de execução não linear. Com uma peça fina no canto esquerdo de cada bloco de estrutura condicional ou de repetição correndo verticalmente sobre as lacunas da placa de suporte, o usuário cego

não precisa ler cada bloco interno até encontrar o bloco FIM, pois basta seguir abaixo pela indentação até o fim (Figura 6.14).

Desta forma, dispensa-se a leitura de um subconjunto de blocos dentro de uma estrutura condicional caso a condição associada seja falsa, e o início do bloco de repetição é facilmente encontrado para a execução de comandos em loop. Isto não somente traz facilidade e praticidade de uso ao artefato, mas também representa fielmente como as linguagens de programação operam sobre loops e if-else, valorizando o potencial didático dele sobre algoritmos e lógica de programação.



Figura 6.14: Algoritmo construído por blocos de encaixe vertical guiado por lacunas, com indentação completa tangível.

6.7 Artefatos em impressão 3D

Dado o sucesso dessa terceira iteração do artefato, o Prof.. H. fez um projeto 3D do artefato de blocos deslizantes em lacunas e imprimiu uma amostra em uma impressora 3D para testagem do encaixe. A amostra consiste em um bloco com escrita Braille em plástico ABS com formato retangular e uma base trapezoidal abaixo (Figura 6.15). A base do bloco encaixa-se com deslizamento horizontal em lacunas na base de suporte, também em impressão 3D com plástico ABS (Figura 6.16).

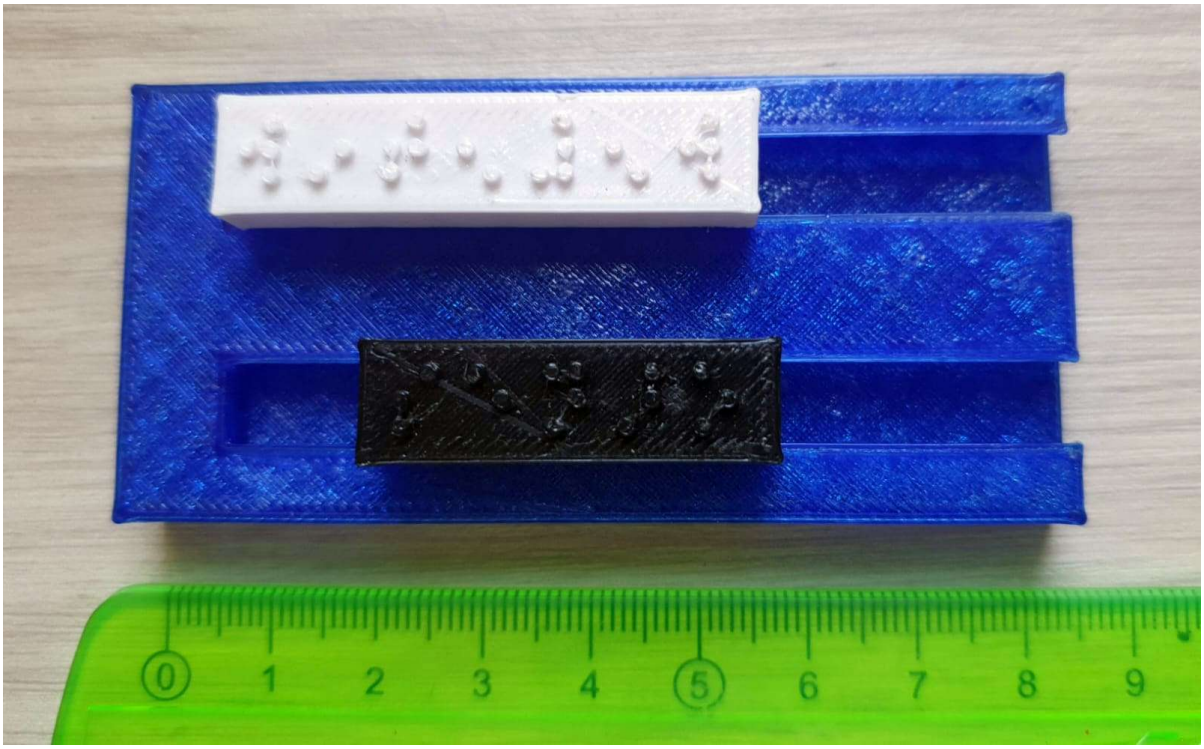


Figura 6.15: Vista superior de amostra do artefato com encaixe deslizante, feito em impressora 3D e plástico PBS.

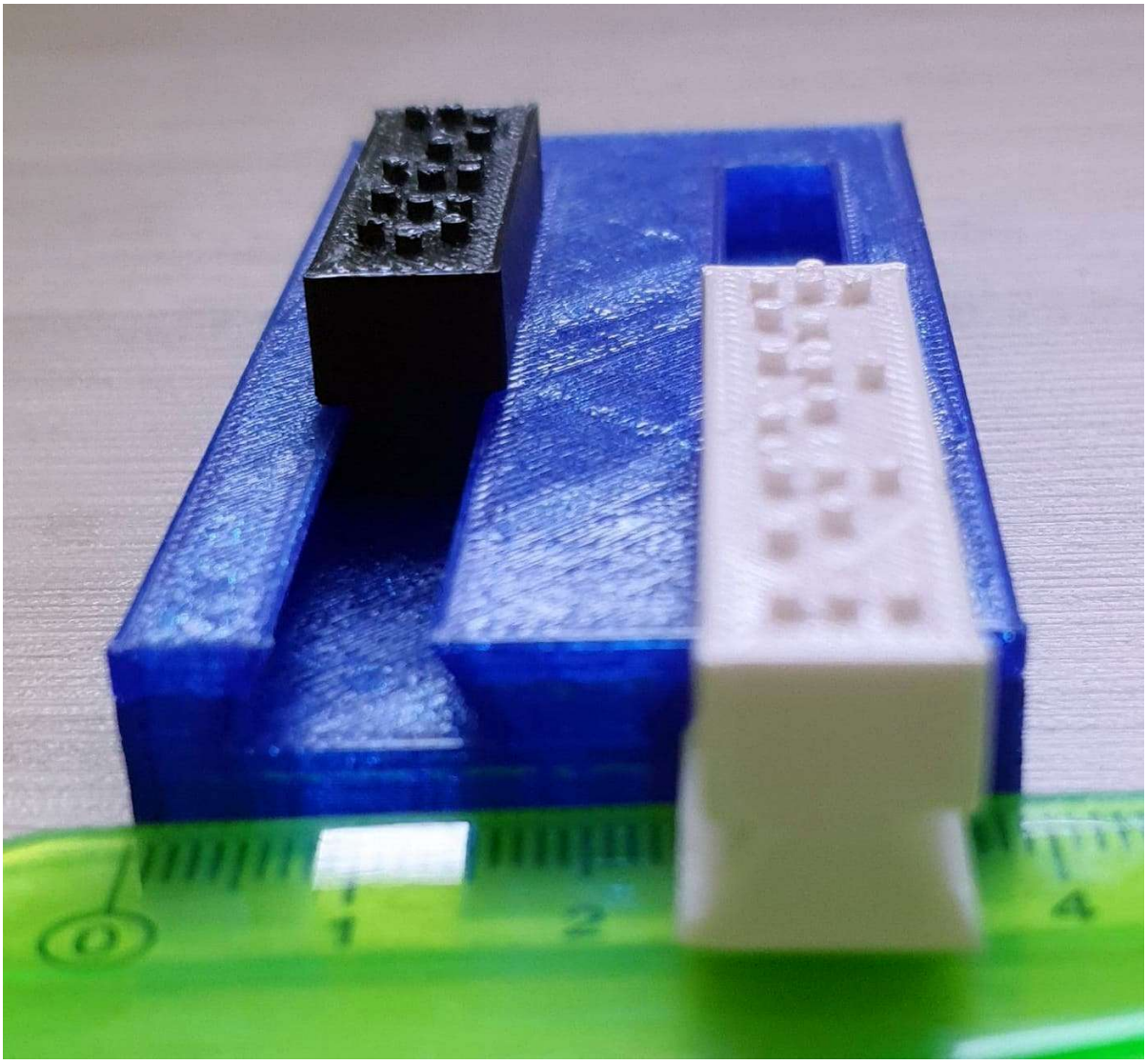


Figura 6.16: Vista lateral de amostra do artefato com encaixe deslizante, feito em impressora 3D e plástico PBS.

Obteve-se uma boa qualidade da impressão, tanto em estrutura como em qualidade da leitura Braille. No entanto, o encaixe não é capaz de travar o bloco em uma posição fixa sem criar atrito excessivo ao deslizar, nem possibilita implementar algum tipo de trava com outra peça que seja durável e intuitiva a um usuário cego. Isso faz com que o bloco deslize enquanto é lido ou é suspenso junto à base quando movido fora da superfície de apoio. Não somente isso prejudica fortemente a leitura Braille, ambas as situações podem fazer com que o bloco se desprenda totalmente da base. Portanto, essa versão do artefato foi avaliada como inválida.

Foi realizada uma modificação do sistema de encaixe, adicionando-se pinos fixadores horizontais aos blocos. O encaixe das formas trapezoidais e o subsequente deslize dispen-

sam a necessidade de mirar o pino no orifício como na versão em pinos de LEGO, tornando a solução intuitiva para usuários deficientes visuais. Porém, a durabilidade da estrutura do pino é um fator limitante, pois está sujeita a desgaste acelerado com plásticos menos resistentes e quebras com falta de precisão no encaixe (Figura 6.17, Figura 6.18).

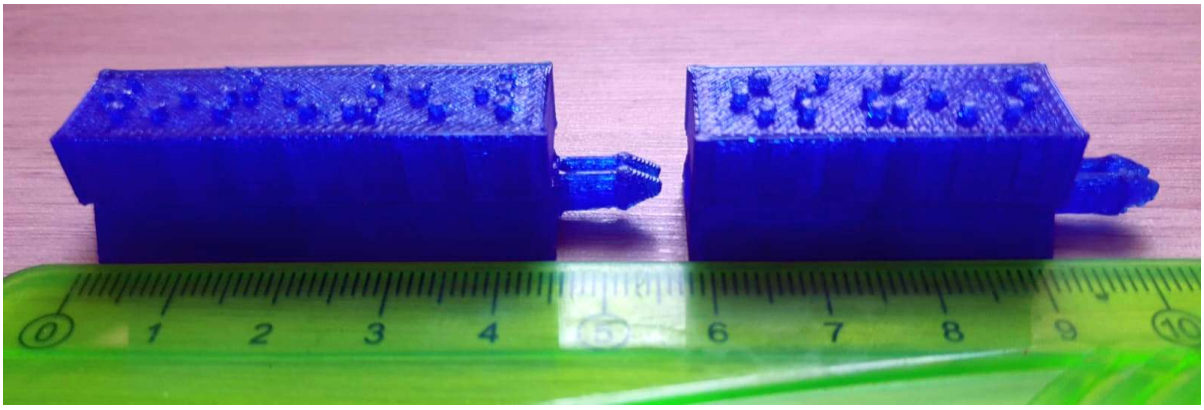


Figura 6.17: Vista de frente de amostra do artefato com encaixe de pinos, feito em impressora 3D e plástico PBS.

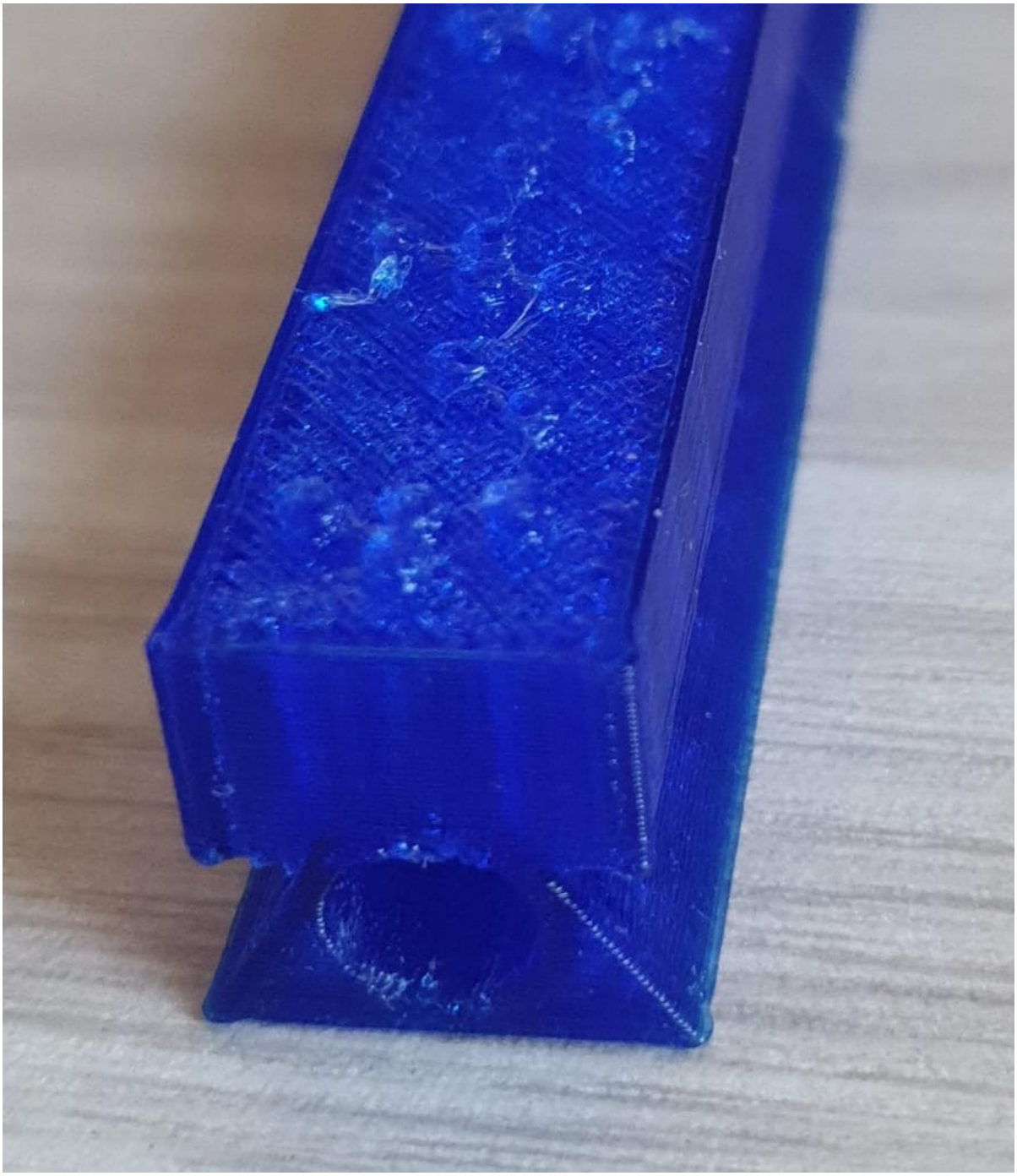


Figura 6.18: Vista lateral de amostra do artefato com encaixe de pinos, feito em impressora 3D e plástico PBS.

Capítulo 7

Avaliações

Este capítulo discute os resultados da pesquisa por meio das três avaliações previstas na metodologia *Design Science Research* (DSR) conforme o esquema estruturado contido na Figura 2.1 apresentada no Capítulo 2.

7.1 Os artefatos desenvolvidos são válidos?

O artefato desenvolvido encontra-se em uma versão inicial e imatura de protótipo que, limitada a uma prova de conceito, é válida para este fim por conseguir investigar cientificamente a capacidade de DVs leigos em programação a aprender e manipular algoritmos. Nas duas subseções a seguir, a validade do artefato é discutida individualmente para as duas conjecturas teóricas discutidas no Capítulo 3.

7.1.1 Validade didática da Programação Baseada em Blocos Tangíveis em LEGO para ensino de algoritmos à DVs leigos

Desenvolver Programação Baseada em Blocos (PBB) tangível para deficientes visuais implica em uma série multidisciplinar de desafios. Houve problemas de design tangível pertinentes ao processamento espacial dos blocos e do labirinto, pois uma mera representação tangível do virtual existente no Blockly não foi suficiente por ainda assim ser dependente de informações primariamente visuais. Enquanto o uso do LEGO foi um fator limitante do artefato em questão de acessibilidade e viabilidade de produção, os resultados do protótipo foram promissores e o Centro de Ensino Especial para Deficientes Visuais (CEEDV) tem expectativas de continuidade do projeto para progredir e refinar o design do artefato com novas iterações.

O principal problema do uso do LEGO como material base dos blocos tangíveis encontra-se em seu design inadequado para deficientes visuais: encaixes com dimensões

diminutas, menores que a média da ponta de um dedo indicador adulto, e com força aderente excessiva que torna enfadonhos e desajeitados os encaixes e desencaixes sucessivos para construir o código tangível como um jogo.

Uma possível melhoria de design dos blocos tangíveis é a troca da escrita Braille por símbolos pictográficos. No entanto, protótipos em impressão 3D indicam que um design intuitivo com pontos móveis reduzidos, encaixe firme e deslize entre as peças com pouco atrito, pode sim ter resultados promissores de um design adequado para sair de protótipo a uma versão estável. É importante considerar e investigar a dimensão tangível mínima do tamanho da ponta de um dedo indicador para quaisquer que sejam as escolhas de design para iterações futuras do artefato.

Portanto, a aplicação da Cognição Situada nos modelos do AlgoBlock [13] deve nortear as escolhas de design tangível e a execução de práticas didáticas por fundamentar a importância fundamental da execução tangível de código de forma lúdica para a aprendizagem de algoritmos por DVs, integrando aspectos técnicos e sociais no design de um artefato manipulável de uso colaborativo.

Com este sistema, deve-se incentivar os usuários a criarem suas próprias versões de blocos tangíveis, iniciando-se pela escolha de outro contexto para o logo além do labirinto, seguindo pela escolha de vocabulário da linguagem de programação tangível e design da outra parte do artefato de Programação Baseada em Blocos (PBB) que torna tangível a execução do código de acordo com o novo jogo proposto.

Ademais, o uso de programação imperativa pelo Blockly Games, devido à sua interface visual, levou a incompatibilidades do artefato na relação de interdependência dos blocos tangíveis e o labirinto tangível para manuseio por DVs. O paradigma de programação imperativa consiste em o programador ter o controle direto do programa por comandos explícitos de como o código deve ser executado. Para o artefato, essa abordagem pode levar a uma experiência enfadonha, pois o protagonista é o programador, que está no papel de manipular os blocos, enquanto a significância do uso do artefato por DVs está no labirinto, manipulado pelo navegador.

7.1.2 Validade pedagógica da Programação Baseada em Blocos Tangíveis em LEGO como método de aculturamento em programação para DVs leigos

As aulas exploratórias em duplas com os blocos em LEGO indicam que a Programação Baseada em Blocos (PBB) tangível consegue, para portadores de deficiência visual também, proporcionar atividades lúdicas como um jogo cativante e imersivo, tal como demonstrado pelo AlgoBlock [13]. Porém, aplicar a Cognição Localizada na ausência de visão

é um grande desafio neste projeto, pois ele requer que os usuários cruzem informações dinâmicas entre código tangível e labirinto durante montagem, execução e depuração.

Isto é favorável no sentido de desenvolver o Pensamento Computacional por tornar o aluno em um computador que executa o código tangível, mas é um território pedagógico inexplorado e que requer um trabalho mais conjunto e minucioso com docentes e pedagogos da comunidade de deficiência visual do que foi realizado neste estudo para atingir um rigor científico satisfatório para compor novas teorias formais de ensino de computação.

A aplicação do conceito de Zona de Desenvolvimento Proximal, da Teoria Histórico-Cultural de Lev Vygostky [14] pode ser fundamental para mais explorações destes aspectos socialmente interativos, pois a deficiência visual é um espectro que afeta cada indivíduo portador de forma exclusivamente idiossincrática. Nas aulas exploratórias, buscou-se formar pares de alunos com competências e deficiências visuais distintas. Por exemplo, a decisão por uma dupla de um portador de cegueira congênita com um portador de baixa visão foi uma escolha acertada porque a baixa visão confere vantagem ao usar o labirinto no papel de navegador pela presença residual de percepção espacial pela visão, enquanto a cegueira congênita condiciona o portador a ter alta proficiência em Braille, fundamental para o papel de programador usando os blocos tangíveis.

Beneficiando-se dos pontos fortes de cada um para introduzir o artefato, a dupla rapidamente se adaptou ao jogo e ambos conseguiram ensinar o parceiro a usar a outra parte do artefato dentro da mesma sessão. Esta espontaneidade em explorar o jogo e aprender com um parceiro de forma cooperativa é um indicativo promissor de que o artefato tem potencial para formar uma comunidade de aprendizes de algoritmos entre deficientes visuais, ou até mesmo entre eles e enxergantes caso haja sucesso em produzir um design híbrido do artefato que contemple as necessidades de baixa-visão também.

7.2 O problema foi resolvido?

A versão atual do artefato é insuficiente tanto para uso didático dentro de um currículo escolar com avaliação formativa, como também para apropriação cultural pela comunidade do Centro de Ensino Especial para Deficientes Visuais (CEEDV) e uso independente por deficientes visuais. Portanto, o problema não foi resolvido, mas pode chegar a ser com iterações futuras desenvolvidas dentro da metodologia da *Design Science Research* (DSR).

7.3 As conjecturas teóricas foram desafiadas pela sua experiência?

As conjecturas teóricas sobre aspectos técnicos de design tangível foram plenamente desafiadas e testadas. Para que os aspectos sociais da ludicidade sejam desafiados, relacionados à segunda conjectura teórica, este autor julga que a qualidade do artefato precisa melhorar pelo design de uma iteração que resulte em um protótipo funcional o suficiente para executar aulas com um planejamento estruturado e metódico, buscando evoluir da abordagem exploratória empregada até então nas atividades em campo.

Capítulo 8

Conclusões

Este capítulo encerra este relato com deliberações finais e propostas futuras para dar continuidade ao projeto iniciado por esta pesquisa, a fim de que ele atinja o impacto social idealizado de uma sociedade mais justa e igualitária para DVs por meio do domínio do pensamento computacional para possibilitar a apropriação cultural do mundo digital.

8.1 O trabalho cumpriu os objetivos?

O trabalho não atingiu o objetivo geral de beneficiar diretamente o CEEDV e produzir teorias sólidas, mas providenciou pontos de investigação e design promissores e pouco documentados na literatura científica. As iterações de design revelaram detalhes importantes sobre a usabilidade tangível de Programação Baseada em Blocos (PBB) por deficientes visuais que podem nortear estudos futuros. As experiências e feedback das aulas exploratórias frisam a importância impreterível do artefato ser também um instrumento de socialização e aprendizagem coletiva.

8.2 Limitações do trabalho

O trabalho foi limitado por ausência de verba financeira, que levou ao uso de LEGO para a construção dos blocos tangíveis e labirintos ao invés de uma solução personalizada. Também foi um limitante a duração de apenas um semestre para efetuar a pesquisa de campo, que compreendeu design e as aulas exploratórias.

8.3 Propostas futuras

Iniciativas futuras a partir deste estudo devem projetar novas iterações do artefato em conformidade com a metodologia *Design Science Research* (DSR), com materiais que

deem suporte a um design totalmente personalizado, como plástico em impressão 3D, ou madeira cortada com uso de maquinário de precisão. Este autor também recomenda substituir a escrita de linguagem humana em sistema Braille por símbolos pictográficos personalizados, alternativa esta que deve ser explorada para diminuir a carga cognitiva do artefato e foi validada em outro estudo já aqui brevemente descrito [17].

O uso de mnemômicos em sistema Braille também deve ser uma alternativa a se considerar. Ela usa a proficiência com este sistema de grafia como subsunçor por parte de DVs e membros do corpo técnico e docente, mas pode fazer com que perdurem os problemas de encaixe de blocos na superfície fixadora devido ao comprimento variável deles.

Estendendo essa recomendação, este autor também considera relevante explorar o uso do paradigma de programação funcional no design de futuras iterações. Os blocos em LEGO são identificados pelo vocabulário do labirinto do Blockly Games com exatamente a mesma escrita usando o sistema Braille, e também têm dependência em estruturas de decisão e repetição que foram difíceis de construir e manipular. Uma abordagem para a sintaxe de código inspirada no LISP, a qual é baseada em funções puras escritas em notação S-expressão, pode tornar o artefato mais compacto para melhorar diversos de seus aspectos [19].

Transformando as palavras Braille por mnemônicos Braille ou pictográficos, os blocos tangíveis podem ser reduzidos a cubos de tamanho único como no artefato do Scratch tangível [17]. Os parênteses podem ser representados por blocos com o próprio parêntese em alto-relevo, ou por mnemônico Braille ou símbolo pictográfico de escolha do *stakeholder*. Desta forma, este autor conjectura que o código pode ser escrito por S-expressões adaptadas da linguagem de programação *Common LISP* com uma carga cognitiva menor em comparação ao artefato em blocos de LEGO, por empregar menor quantidade de símbolos e espaço tangível em memória de curto prazo [20]. Ou seja, essa abordagem pode diminuir a verbosidade do código produzido pelo artefato e torná-lo mais compacto tangivelmente para melhorar sua manipulação.

Quanto ao planejamento da aplicação do artefato, recomenda-se que, para evoluir da abordagem exploratória para uma rigorosa, seja feito e executado um planejamento didático-pedagógico de cada aula para melhorar a coleta de dados estruturada e possibilitar uma avaliação formativa de aprendizagem. Este autor considera crucial esse rigor para que o artefato continue a contribuir para a ciência da aprendizagem em computação e possa ter sucesso em resolver o problema pela adoção em currículos de instituições de ensino que atendem DVs.

Após um protótipo funcional ser implementado e adotado com sucesso, deve-se incentivar que *stakeholders* DVs desenvolvam um novo jogo para o artefato. Ou seja, um

ambiente de execução tangível de código com criação autoral de vocabulário, regras, nomes e entidades manipuláveis ao invés de uma adaptação direta do *Blockly Games*, ou outro ambiente de PBB já existente. Isto é fundamental para que o problema da insuficiência de recursos didáticos para ensino de algoritmos à DVs seja resolvido, porque o artefato precisa ser adotado em currículos de instituições de ensino para tal e, portanto, a liberdade de planejamento didático é um requisito imprescindível.

Referências

[1] Pimentel, Mariano, Denise Filippo e Flávia Maria Santoro: *Design science research: fazendo pesquisas científicas rigorosas atreladas ao desenvolvimento de artefatos computacionais projetados para a educação*. Em Jaques, Patricia, Mariano Pimentel, Sean Siqueira e Ig Bittencourt (editores): *Metodologia de Pesquisa Científica em Informática na Educação: Concepção de Pesquisa (Volume 1)*, volume 1 de *Metodologia da CEIE-SBC*, página 29. SBC, Porto Alegre, 2018. https://ceie.sbc.org.br/metodologia/wp-content/uploads/2018/10/cap1_5.pdf, acesso em 2022-09-23. xi, 13, 14, 16, 17, 18

[2] Garcia, Fabiane Maia e Aissa Thamy Alencar Mendes Braz: *Deficiência visual: caminhos legais e teóricos da escola inclusiva*. Ensaio: Avaliação e Políticas Públicas em Educação, 28:622–641, maio 2020, ISSN 0104-4036, 1809-4465. https://www.scielo.br/j/ensaio/a/6D8gzB5Dd7vnLG3FXmvN4bw/, acesso em 2025-06-09, Publisher: Fundação CESGRANRIO. 1

[3] Silva, Rafael Soares, Agnaldo Ronie Pezarini, Manoel Aguiar Neto Filho e Fabio José Antonio Da Silva (editores): *Educação & ensino na contemporaneidade*. Editora Metrics, 1ª edição, 2022, ISBN 978-65-5397-012-0. https://editorametrics.com.br/livro/educacao-ensino-na-contemporaneidade, acesso em 2025-06-09. 1

[4] Almeida, José Júlio Gavião de e Eline T. R. Porto: *A pessoa deficiente visual: revelações sobre a atividade motora*. Benjamin Constant, (12), 1999, ISSN 1984-6061. https://revista.ibc.gov.br/index.php/BC/article/view/614, acesso em 2025-06-09, Number: 12. 1

[5] 29/09/2023 14h01, Publicado em 28/04/2023 15h09 Atualizado em: *Base Nacional Comum Curricular (BNCC)*. https://www.gov.br/mec/pt-br/acesso-a-informacao/participacao-social/conselhos-e-orgaos-colegiados/cne/base-nacional-comum-curricular-bncc, acesso em 2025-06-09. 1

[6] *L13146*. https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm, acesso em 2025-06-09. 1

[7] *ALGORITMOS*. https://drive.google.com/drive/u/0/folders/1aE84YmBa4RBhYbWtZnXCVykB1cRAjTYj, acesso em 2025-11-09. 3

[8] *5. Data Structures*. https://docs.python.org/3/tutorial/datastructures.html, acesso em 2025-07-14. 4

[9] *NV Access | Download NVDA*. https://www.nvaccess.org/download/, acesso em 2025-11-11. 11

[10] Vieira, Anacilia, Odette Passos e Raimundo Barreto: *Um Relato de Experiência do Uso da Técnica Computação Desplugada*. Em *Workshop sobre Educação em Computação (WEI)*, páginas 671–680. SBC, julho 2013. https://sol.sbc.org.br/index.php/wei/article/view/27763, acesso em 2025-06-10, ISSN: 2595-6175. 11

[11] Ferreira, Ana Carolina C., André Melhor, Jandiaci dos S. Barreto, Luiz Fernando de Paiva e Ecivaldo Matos: *Experiência Prática Interdisciplinar do Raciocínio Computacional em Atividades de Computação Desplugada na Educação Básica*. Em *Workshop de Informática na Escola (WIE)*, páginas 256–265. SBC, outubro 2015. https://sol.sbc.org.br/index.php/wie/article/view/16525, acesso em 2025-06-10, ISSN: 0000-0000. 11, 23, 24, 26

[12] Morrison, Cecily, Villar , Nicolas, Thieme , Anja, Ashktorab , Zahra, Taysom , Eloise, Salandin , Oscar, Cletheroe , Daniel, Saul , Greg, Blackwell , Alan F, Edge , Darren, Grayson , Martin, e Haiyan Zhang: *Torino: A Tangible Programming Language Inclusive of Children with Visual Disabilities*. Human–Computer Interaction, 35(3):191–239, maio 2020, ISSN 0737-0024. https://doi.org/10.1080/07370024.2018.1512413, acesso em 2025-03-18, Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07370024.2018.1512413. 15, 31

[13] Suzuki, Hideyuki e Hiroshi Kato: *Interaction-level support for collaborative learning: AlgoBlock—an open programming language*. Em *The first international conference on Computer support for collaborative learning*, CSCL '95, páginas 349–355, USA, outubro 1995. L. Erlbaum Associates Inc., ISBN 978-0-8058-2243-4. https://dl.acm.org/doi/10.3115/222020.222828, acesso em 2025-04-23. 15, 29, 53

[14] Ferreira, Tarciana Cecília de Souza e Maria Sirlene Pereira Schlickmann: *A teoria histórico-cultural e a educação escolar numa perspectiva humanizadora*. Revista Ibero-Americana de Estudos em Educação, páginas 0643–0660, março 2022, ISSN 1982-5587. https://periodicos.fclar.unesp.br/iberoamericana/article/view/15753, acesso em 2025-07-23. 25, 54

[15] Huizinga, Johan: *HOMO LUDENS: O JOGO COMO ELEMENTO DA CULTURA*. Perspectiva, 2014, ISBN 978-85-273-1157-1. 25

[16] *What is Blockly*. https://developers.google.com/blockly/guides/get-started/what-is-blockly, acesso em 2025-04-17. 28

[17] Goolsby, Bryson, Dianne Pawluk, Hyun Woo Kim e Giovanni Fusco: *A Tangible Block Editor for the Scratch Programming Language*. Em *EXTENDED ABSTRACTS OF THE 2021 CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS (CHI'21)*, New York, 2021. Assoc Computing Machinery, ISBN 978-1-4503-8095-9. Num Pages: 6 Web of Science ID: WOS:000759178502143. 30, 57

[18] Das, Meenakshi, Daniela Marghitu, Mahender Mandala e Ayanna Howard: *Accessible Block-Based Programming for K-12 Students Who Are Blind or Low Vision*. Em Antona, M. e C. Stephanidis (editores): *UNIVERSAL ACCESS IN HUMAN-COMPUTER INTERACTION. ACCESS TO MEDIA, LEARNING AND ASSISTIVE ENVIRONMENTS, UAHCI 2021, PT II*, volume 12769, páginas 52–61, Cham, 2021. Springer International Publishing Ag, ISBN 978-3-030-78094-4 978-3-030-78095-1. Num Pages: 10 Series Title: Information Systems and Applications, incl. Internet/Web, and HCI Web of Science ID: WOS:001297846800005. 32

[19] *Tutorial LISP*. `https://www.dca.ufrn.br/~adelardo/lisp/`, acesso em 2025-07-18. 57

[20] *S-expression*, março 2025. `https://en.wikipedia.org/w/index.php?title=S-expression&oldid=1278820455`, acesso em 2025-07-18, Page Version ID: 1278820455. 57