



INSTITUTO FEDERAL DE ALAGOAS
CAMPUS ARAPIRACA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

MIGUEL MÁRCIO MAGALHÃES SILVA

**TRABALHANDO O PENSAMENTO COMPUTACIONAL EM ALUNOS COM
DEFICIÊNCIA VISUAL**

ARAPIRACA, AL
2023

MIGUEL MÁRCIO MAGALHÃES SILVA

TRABALHANDO O PENSAMENTO COMPUTACIONAL EM ALUNOS COM
DEFICIÊNCIA VISUAL

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Sistemas de Informação do Instituto Federal de Alagoas, Campus Arapiraca, como requisito parcial para obtenção de grau de Bacharel em Sistemas de Informação.

Orientadora: Dra. Cledja Karina Rolim da Silva.

ARAPIRACA, AL

2023



Dados Internacionais de Catalogação na Publicação
Instituto Federal de Alagoas
Campus Arapiraca

S586t

Silva, Miguel Márcio Magalhães.

Trabalhando o pensamento computacional em alunos com deficiência visual / Miguel Márcio Magalhães Silva. – 2023.

1 PDF: il., color. (1 arquivo : 6,9 MB).

Arquivo digital no formato PDF do trabalho acadêmico com 68 folhas.

Orientação: Prof^ª. Dr^ª. Cledja Karina Rolim da Silva
Trabalho de Conclusão de Curso (Graduação,
Bacharelado em Sistemas de Informação) – Instituto Federal
de Alagoas, *Campus Arapiraca*, Arapiraca, 2023.

1. Lógica de programação. 2. Deficientes visuais. 3.
Blocos físicos 3D. 4. Aplicativo web progressivo. I. Título.

CDD: 005.1

MIGUEL MÁRCIO MAGALHÃES SILVA

TRABALHANDO O PENSAMENTO COMPUTACIONAL EM ALUNOS COM
DEFICIÊNCIA VISUAL

Trabalho apresentado ao Curso Superior de Sistemas de Informação do Instituto Federal de Alagoas, campus Arapiraca, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado(a) em: 15/12/2023.

BANCA EXAMINADORA:


Cledja Karina Rolim da Silva

Prof. Dra. Cledja Karina Rolim da Silva (Orientador)
Instituto Federal de Alagoas - IFAL



Prof. MSc. Anderson Felinto Barbosa
Instituto Federal de Alagoas - IFAL



Prof. MSc. Diogo Sarmiento Ferreira
Instituto Federal de Alagoas - IFAL

AGRADECIMENTOS

Ao Instituto Federal de Alagoas, agradeço pela estrutura que me proporcionou trilhar meu caminho rumo ao sonho de obter o título de Bacharel.

Esta conquista, simbolizada por este trabalho, é o resultado de uma jornada repleta de aprendizado e apoio. Reconheço e sou grato por ter tido o suporte fundamental de pessoas extraordinárias, as quais tornaram possível a conclusão deste trabalho acadêmico.

Aos meus pais, Márcio e Michele, verdadeiros alicerces em minha vida. A orientação e o apoio incansável que sempre me ofereceram, foram essenciais para que eu pudesse alcançar cada etapa desta jornada acadêmica.

À minha companheira, Verônica, por ser uma fonte constante de motivação e apoio. Seu incentivo e compreensão me proporcionaram a serenidade necessária para superar desafios e seguir em frente com este trabalho.

À minha orientadora, Prof. Dra. Cledja Rolim, pela orientação dedicada e por compartilhar sua experiência e sabedoria. Seu suporte foi fundamental para o desenvolvimento deste trabalho, e sou imensamente grato por sua orientação.

Aos meus amigos e colegas, em especial Jousiclecia, Bianca, Gabriel, João e Ismael pelo apoio e amizade ao longo dessa jornada.

RESUMO

Este projeto se concentra na aplicação do pensamento computacional e da lógica de programação para pessoas com deficiência visual, visando desenvolver uma aplicação de aprendizado acessível. Com o uso de blocos físicos 3D previamente desenvolvidos e *QR codes*, a aplicação oferece uma interface acessível e interativa, permitindo que os usuários montem algoritmos de forma tangível e passo a passo. Tudo isso está em conformidade com as diretrizes de acessibilidade web, proporcionando uma experiência adaptada para dispositivos móveis e leitores de tela. Após a implementação, foi realizada uma avaliação detalhada para analisar a eficácia da ferramenta. Este trabalho destaca os resultados dessa avaliação, bem como os aprendizados obtidos durante o desenvolvimento. O foco está na aplicabilidade prática da ferramenta no ensino de lógica de programação para pessoas com deficiência visual, com o objetivo de proporcionar uma experiência de aprendizado acessível e eficaz. Os resultados demonstraram que a ferramenta traz benefícios significativos para a aprendizagem de programação por parte das pessoas com deficiência visual, não apenas garantindo a inclusão desses alunos, mas também enriquecendo a qualidade do ensino como um todo.

Palavras Chave: deficiente visual; lógica de programação; blocos físicos 3D; aplicativo web progressivo.

ABSTRACT

This project focuses on applying computational thinking and programming logic to individuals with visual impairments, aiming to develop an accessible learning application. Using pre-developed 3D physical blocks and QR codes, the application provides an accessible and interactive interface, allowing users to construct algorithms tangibly and step-by-step. All of this aligns with web accessibility guidelines, providing an adapted experience for mobile devices and screen readers. Following implementation, a detailed evaluation was conducted to assess the tool's effectiveness. This work highlights the results of this assessment, along with the lessons learned during development. The emphasis lies in the practical applicability of the tool in teaching programming logic to individuals with visual impairments, with the goal of providing an accessible and effective learning experience. The results demonstrated that the tool brings significant benefits to programming learning for individuals with visual impairments, not only ensuring their inclusion but also enhancing the overall quality of education.

Keywords: visual impairment; programming logic; 3D physical blocks; progressive web application.

LISTA DE ILUSTRAÇÕES

Figura 1 - Estrutura dos elementos da WCAG 2.....	20
Figura 2 - Utilização da role search juntamente com tag HTML.....	25
Figura 3 - Utilização da role main juntamente com tag HTML.....	26
Figura 4 - Exemplo de página HTML com os roles do WAI-ARIA.....	27
Figura 5 - Exemplo do uso do WAI-ARIA em campos de entrada do HTML.....	27
Figura 6 - Exemplo de código com informação adicional acessível.....	28
Figura 7 - Peças físicas modulares do code jumper.....	30
Figura 8 - Blocos de código a partir da montagem dos pods e hub.....	31
Figura 9 - Blocos físicos montados do BATS	31
Figura 10 - Alunos jogando o jogo de tabuleiro.....	32
Figura 11 - Representação gráfica do fluxo da metodologia proposta.....	34
Figura 12 - Tela principal do Scratch.....	35
Figura 13 - Bloco impresso na impressora 3D e utilizando adesivos com Braille.....	36
Figura 14 - Bloco impresso utilizando adesivo com Braille e o QR code.....	38
Figura 15 - Protótipo de baixa fidelidade das páginas da aplicação.....	42
Figura 16 - Protótipo de média fidelidade das páginas.....	44
Figura 18 - Apresenta a tela inicial do aplicativo.....	46
Figura 18 - Apresenta a tela de tutorial do aplicativo.....	47
Figura 19 - Página da aplicação com a explicação do bloco variável.....	48
Figura 20 - Exemplo de popup de navegadores para habilitar o uso da câmera.....	49
Figura 21 - Utilização da câmera do dispositivo do usuário.....	50
Figura 22 - Principais telas da aplicação desenvolvida.....	51
Figura 23 - Usuário testando a página de tutorial de um dos blocos.....	54
Figura 24 - Usuário testando a página de exercícios.....	54
Figura 25 - Usuário na a página de uma questão de algoritmo.....	55
Figura 26 - Usuário realizando a leitura de QR code de um bloco.....	56

LISTA DE QUADROS

Quadro 1 - Descrição das tags HTML e suas respectivas definições.....	23
Quadro 2 - Apresentação das tags HTML e sua respectiva role.....	24
Quadro 3 - Descrição dos blocos e seus respectivos QR codes.....	39
Quadro 4 - Questões elaboradas para o levantamento do estudo de caso.....	56

SUMÁRIO

1	INTRODUÇÃO.....	12
2	FUNDAMENTAÇÃO TEÓRICA.....	15
2.1	DEFICIÊNCIA VISUAL.....	15
2.1.1	Baixa Visão.....	15
2.1.2	Cegueira.....	16
2.1.3	Daltonismo.....	16
2.2	LÓGICA DE PROGRAMAÇÃO.....	17
2.3	DIRETRIZES DE ACESSIBILIDADE PARA PÁGINAS WEB.....	17
2.3.1	Diretrizes de Acessibilidade para Conteúdo Web.....	18
2.3.2	Aplicações para a Internet Ricas em Acessibilidade.....	21
2.3.2.1	Roles.....	24
2.3.2.2	Properties.....	28
2.4	CONSIDERAÇÕES FINAIS.....	28
3	TÉCNICAS DE ENSINO DE PROGRAMAÇÃO PARA CEGOS.....	30
4	Echo 3D Blocks.....	34
4.1	FLUXOGRAMA FÍSICO 3D.....	35
4.2	USO DE QR CODES.....	37
4.3	FERRAMENTAS UTILIZADAS.....	40
4.4	PROTOTIPAGEM.....	41
4.5	FUNCIONALIDADES.....	44
4.6	CONSIDERAÇÕES FINAIS.....	51
5	ESTUDO DE CASO.....	52
5.1	TAREFAS PROPOSTAS.....	52
5.2	EXECUÇÃO DA AVALIAÇÃO.....	53
5.4	ANÁLISE E DISCUSSÃO.....	59

5.4.1	Conhecimento prévio em lógica de programação.....	59
5.4.2	Uso dos blocos 3D físicos.....	59
5.4.3	Uso da aplicação.....	60
5.4.4	Leitura de QR codes e feedbacks da aplicação.....	60
5.4.5	Dificuldades encontradas.....	61
5.4.6	Sugestões de melhorias.....	61
5.5	CONSIDERAÇÕES FINAIS.....	62
6	CONCLUSÕES E TRABALHOS FUTUROS.....	63
	REFERÊNCIAS.....	65
	APÊNDICE A - TERMO DE CONSENTIMENTO.....	68

1 INTRODUÇÃO

O processo de aprendizagem de lógica de programação é uma tarefa que pode tornar-se árdua para o professor e para os alunos (GOMES *et al.*, 2008). De acordo com LIMA & MEIRELES (2015), diversas formas de tornar o conteúdo menos abstrato e mais próximo do estudante foram desenvolvidas, incluindo o uso de diagramas como o fluxograma, recursos visuais e pseudocódigo a fim de utilizar das inteligências previamente desenvolvidas pelo aluno para fomentar a inteligência lógica.

Apesar dos esforços, muitas destas formas alternativas não são criadas pensando nas pessoas com deficiências visuais, tornando estas alternativas inviáveis e fazendo com que o desenvolvimento do pensamento computacional seja inacessível para pessoas cegas. Há 2,2 bilhões de pessoas em todo o mundo com algum tipo de deficiência visual ou cegueira no relatório apresentado pela Organização Mundial da Saúde (WORLD HEALTH ORGANIZATION, 2019).

Segundo a WORLD HEALTH ORGANIZATION (2011), estima-se que 15% da população mundial tem algum tipo de deficiência, 2 a 4% deles gravemente, tornando as pessoas com deficiência o maior grupo minoritário do mundo. Porém, a tecnologia tornou possível que pessoas com diversos tipos de deficiência explorem o mundo digital e também reduza suas limitações.

Diante dessa dificuldade, algumas contribuições foram desenvolvidas para auxiliar na aprendizagem de alunos com deficiência visual. Como exemplo, o projeto de iniciação científica do Instituto Federal de Alagoas (IFAL), campus Arapiraca, que trabalhou numa abordagem inovadora para facilitar o aprendizado de programação por alunos com deficiência visual (VIEIRA *et al.*, 2023). A inovação consiste na transformação de fluxogramas, comumente utilizados no ensino de programação como representação visual de processos e algoritmos, em blocos táteis em 3D.

Esses blocos táteis oferecem uma representação física e tangível de conceitos abstratos, proporcionando uma experiência direta e prática para pessoas com deficiência visual compreenderem elementos de programação. Essa adaptação garante um ambiente de aprendizado mais inclusivo e abrangente ao aproveitar o sentido do tato para tornar os conceitos de programação acessíveis e tangíveis para

aqueles com deficiências visuais. Entretanto, o projeto carecia de uma ferramenta para orientar o uso dos blocos pelos usuários. O presente trabalho de implementação de software acessível tem como um dos principais públicos-alvo os deficientes visuais. Ademais, é essencial expandir essa abordagem para além da deficiência visual, considerando outras limitações temporárias ou comuns que possam influenciar a experiência do usuário em aplicativos de software. Por exemplo, dores nas mãos devido ao uso prolongado do mouse, como tendinite ou lesões relacionadas ao esforço repetitivo, exigem a navegação na web apenas por meio do teclado, destacando a importância de interfaces acessíveis.

O problema central abordado por este trabalho diz respeito à acessibilidade no ensino de programação para pessoas com deficiência visual, consistindo em: *é possível tornar o ensino de programação mais acessível e eficaz para pessoas cegas ou com deficiência visual, utilizando blocos 3D táteis?*

A adoção e criação de medidas para ensinar lógica de programação não apenas auxilia no processo de independência das pessoas com deficiências visuais que buscam aprender a programar, mas também desempenha um papel complementar no alcance dos Objetivos de Desenvolvimento Sustentável (ODS) estabelecidos pela ORGANIZAÇÃO DAS NAÇÕES UNIDAS (2000). Esse esforço é particularmente significativo no objetivo 4: "Educação de qualidade", contribuindo, assim, para a promoção de uma sociedade mais justa e equitativa.

Dessa forma, foi possível estabelecer os objetivos da pesquisa da seguinte forma.

1.1 OBJETIVO GERAL

O objetivo deste projeto é desenvolver um *Progressive Web App* (PWA), ou seja, um aplicativo web progressivo para auxiliar pessoas com deficiência visual a explorar e compreender a lógica de programação usando os blocos táteis 3D. O aplicativo, denominado *Echo 3D Blocks - E3B*, não se limita apenas a instruir sobre o uso dos componentes (blocos 3D), mas também introduz conceitos fundamentais de programação, proporcionando a execução de exercícios práticos. Tornando o aprendizado da programação acessível e interativo, para pessoas com deficiência visual.

1.2 OBJETIVOS ESPECÍFICOS

- a. Desenvolver uma aplicação direcionada ao ensino de lógica de programação voltada para pessoas com deficiência visual;
- b. Avaliar a eficácia e usabilidade da ferramenta desenvolvida por meio de um estudo de caso.

Este trabalho se desdobrará em algumas etapas que foram planejadas para atingir os objetivos acima. Inicialmente, no Capítulo 2 é apresentada a fundamentação teórica necessária para o entendimento da pesquisa. No Capítulo 3, são apresentados os trabalhos relacionados a partir da análise da literatura existente e sobre o ensino de programação para deficientes visuais. Já no Capítulo 4, é apresentado a metodologia empregada para o desenvolvimento da solução proposta, o *Echo 3D Blocks*. O Capítulo 5, apresenta a avaliação da solução a partir da realização de estudos de caso e os dados coletados referentes ao estudo de caso que foi realizado neste trabalho, para avaliar a eficácia e usabilidade da ferramenta criada, e responder a pergunta de pesquisa apresentada. Por fim, no Capítulo 6 são apresentados as conclusões e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos fundamentais para melhor compreensão deste trabalho. Na Seção 2.1 é apresentado o que caracteriza ser deficiência visual. Na Seção 2.2 é apresentado sobre o *QR Code*. A Seção 2.3 aborda as normas da WCAG e WAI-ARIA, que tratam de acessibilidade Web, juntamente com o HTML. Por último, na Seção 2.4 são abordadas algumas considerações sobre este capítulo.

2.1 DEFICIÊNCIA VISUAL

A Lei nº 13.146, conhecida como Lei Brasileira de Inclusão ou Estatuto da Pessoa com Deficiência (BRASIL, 2015), define que as pessoas portadoras de deficiência são aquelas que possuem limitações de longo prazo de natureza física, mental, intelectual ou sensorial, as quais, quando confrontadas com diversas barreiras, podem dificultar sua participação plena e igualitária na sociedade, em condições equiparadas às das demais pessoas.

Alguns níveis de deficiência visual são apresentados a seguir.

2.1.1 Baixa Visão

A baixa visão também conhecida como cegueira parcial é classificada como pessoas que são capazes de apenas contar dedos a curta distância e os que percebem vultos, de acordo com a pesquisa de saúde ocular da Organização Pan-Americana da Saúde (PAN AMERICAN HEALTH ORGANIZATION, 2023) o nível de acuidade visual é de 6/12 (leve) a 6/18 (moderada). Pessoas com baixa visão geralmente têm dificuldade em distinguir detalhes, perceber contrastes ou enxergar em determinadas condições de iluminação.

O grupo de pessoas com baixa visão enfrenta desafios específicos ao interagir com conteúdos online, necessitando de adaptações visuais para acessar informações de maneira mais clara e legível. Para atender às necessidades de usuários com baixa visão, é crucial garantir a acessibilidade digital em sites e aplicativos. Isso pode ser feito por meio de opções de aumento de contraste, ajustes de tamanho de fonte, escolha de paletas de cores adequadas, layouts flexíveis que permitam redimensionamento fácil do conteúdo e modo claro/escuro.

2.1.2 Cegueira

A cegueira ou amaurose (termo técnico para perda de visão) é uma condição que implica na perda total da visão de um dos olhos ou ambos, de acordo com a OPAS é considerado cegueira quando os valores de acuidade visual encontram-se abaixo de 3/60, afetando significativamente a maneira como uma pessoa interage e percebe o mundo ao seu redor.

Para as pessoas cegas, a experiência de navegar na internet e interagir com conteúdos online pode apresentar desafios consideráveis. As causas da cegueira podem ser variadas e incluem condições congênitas, como retinopatia da prematuridade, glaucoma congênito ou catarata congênita. Além disso, a cegueira também pode ser adquirida ao longo da vida devido a fatores como lesões oculares, doenças oculares degenerativas (como retinose pigmentar ou degeneração macular), infecções, diabetes não controlada, entre outras causas.

Braille é um sistema de leitura e escrita tátil para as pessoas cegas. O *Braille* foi desenvolvido por Louis Braille em 1825 na França, é baseado na utilização de seis pontos em relevo organizados em uma célula de *Braille*, constituída por um espaço retangular com duas colunas contendo três linhas cada. A combinação específica desses pontos resulta na formação de 63 símbolos distintos, os quais correspondem à escrita em tinta (LEMOS; CERQUEIRA, 2014).

2.1.3 Daltonismo

O daltonismo é uma condição genética que afeta a percepção das cores, também conhecida como cegueira cromática. As pessoas com daltonismo têm dificuldade em distinguir cores específicas, principalmente o verde e o vermelho, ou em alguns casos todas as cores. Essa condição é mais comum nos homens do que em mulheres, por razão da herança genética ligada ao cromossomo X.

Essa anomalia genética na maioria das vezes é identificada no período escolar, onde a criança ou o adolescente começa a demonstrar dificuldades no uso das cores, e em alguns casos é diagnosticado em seu próprio ambiente familiar (MORIJO *et al*, 2020).

2.2 LÓGICA DE PROGRAMAÇÃO

A lógica de programação teve suas raízes nos avanços da ciência da computação. A lógica de programação se refere a capacidade de resolver problemas de forma estruturada e sequencial, utilizando uma linguagem que é compreendida pelos computadores (TECHOPEDIA, 2020).

Ademais, a compreensão da lógica de programação é uma habilidade de suma importância para qualquer pessoa interessada em ciência da computação ou tecnologia, pois ela ajuda a desenvolver o pensamento computacional, melhora a capacidade de solucionar problemas, abstração e criação de algoritmos.

A lógica de programação é uma base fundamental para o desenvolvimento de algoritmos e softwares. Entretanto, muitos alunos apresentam dificuldades como relatado em SOUZA (2016), essa dificuldade está geralmente em entender determinados conceitos de programação, tais como ponteiros, recursão, declaração de variáveis, dentre outros conceitos mais avançados de programação.

No entanto, muitos alunos entendem os conceitos de programação, mas têm dificuldades em aplicá-los e praticar durante a construção de algoritmos. Adicionalmente, é relatado no trabalho de SOUZA (2016) que os alunos veem a programação como uma atividade tediosa e cansativa.

Por fim, alunos com algum tipo de deficiência, seja ela visual, motora, mental, ou auditiva, podem ter sua aprendizagem prejudicada por não conseguir acompanhar as atividades em sala de aula ou recursos didáticos não serem acessíveis. Nesse sentido, é essencial a criação ou utilização de recursos didáticos que viabilizem a inclusão desses estudantes, possibilitando a compreensão dos conceitos apresentados em aula (ROBE *et al.*, 2020).

2.3 DIRETRIZES DE ACESSIBILIDADE PARA PÁGINAS WEB

A acessibilidade na web refere-se às práticas a serem aplicadas na criação de *sites*, de modo que estes podem ser utilizados por qualquer pessoa, incluindo aquelas com deficiências ou que estejam em um ambiente desfavorável como: com conexões de internet lentas, hardware desatualizado ou quebrado. Por exemplo, ao adicionar legendas a um vídeo, auxilia os usuários surdos e com deficiência auditiva, assim como usuários em ambientes barulhentos que não

conseguem ouvir o áudio. Da mesma forma, garantir que o texto tenha contraste adequado ajuda usuários com baixa visão e aqueles que estão tentando usar o telefone sob luz solar intensa.

Padrões para acessibilidade web foram desenvolvidos pela W3C (*World Wide Web Consortium* - Consórcio para a Web), que é uma organização internacional que estabelece padrões para a Web (W3C, 2011). A missão central do W3C é permitir que a Web alcance todo o seu potencial, estabelecendo normas técnicas e promovendo a acessibilidade, interoperabilidade e evolução contínua da World Wide Web (GALA, 2022).

São abordados os conceitos fundamentais tanto das Diretrizes de Acessibilidade para Conteúdo Web (WCAG - *Web Content Accessibility Guidelines*) (WCAG, 2023) quanto da tecnologia Aplicações de Internet Ricas e Acessíveis (WAI-ARIA - *Accessible Rich Internet Applications*) (W3C, 2023). Essa exploração conjunta oferece uma compreensão abrangente e essencial das diretrizes fundamentais para tornar aplicativos e conteúdos web mais acessíveis e inclusivos.

2.3.1 Diretrizes de Acessibilidade para Conteúdo Web

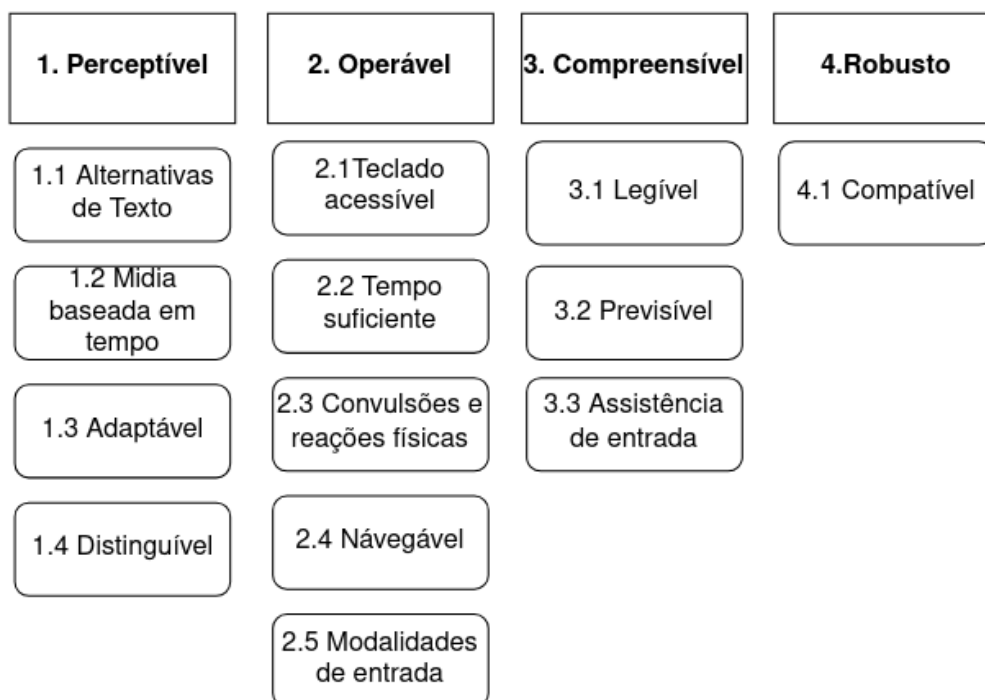
As diretrizes de acessibilidade para conteúdo web representam um conjunto de recomendações propostas pelo consórcio, e estabelece orientações para tornar o conteúdo *online* mais acessível para pessoas com diferentes tipos de deficiência e barreiras.

O WCAG é essencial na criação e design de interfaces digitais, pois garante que websites e aplicativos sejam utilizáveis por um público diversificado, incluindo pessoas com deficiências visuais, motoras, auditivas, cognitivas, entre outras. Ao decorrer dos anos, várias melhorias e refinamento das diretrizes foram lançadas, tais como: A versão 1.0 (WCAG, 1999) que forneceu orientações básicas para a criação de conteúdo acessível e 2.0 (WCAG, 2008) que aprimorou significativamente os critérios, incorporando detalhamentos mais abrangentes para aprimorar a acessibilidade, alinhando-se com os avanços tecnológicos que se expandem anualmente. Algumas das principais adições e melhorias na versão 2.0 em relação à versão anterior, foram:

- *Estrutura mais robusta* - dividindo os princípios de acessibilidade em quatro áreas principais: perceptível, operável, compreensível e robusto;
- *Níveis de conformidade* - adição de níveis de conformidade, sendo: A, AA e AAA. Para indicar o nível de acessibilidade alcançado por um site ou aplicativo, oferecendo diferentes graus de conformidade para atender às necessidades variadas de acessibilidade;
- *Crítérios mais detalhados* - fornece critérios de sucesso mais detalhados e técnicos para orientar o design e desenvolvimento acessível.
- *Alinhamento com tecnologias emergentes* - evoluiu para acompanhar as inovações tecnológicas, fornecendo diretrizes para novas tecnologias, como dispositivos móveis e aplicativos.
- *Inclusão de mais técnicas e exemplos* - ofereceu mais técnicas específicas e exemplos práticos para ajudar os desenvolvedores a implementar as diretrizes com maior eficácia e clareza.

Atualmente, a WCAG versão 2.1 (Figura 1) de 21 de setembro de 2023, estabeleceu que os critérios sejam escritos como declarações estáveis independente de tecnologias específicas. A versão atual estende às versão anterior, os seus conteúdos em conformidade com a versão 2.1 também atendem aos critérios da versão 2.0. A publicação da WCAG 2.1 não torna obsoleta ou substitui a WCAG 2.0. Porém, o W3C não só recomenda o uso da versão atual para maximizar a aplicabilidade futura dos esforços de acessibilidade, mas também incentiva o uso ao desenvolver ou atualizar políticas de acessibilidade na web (WCAG, 2023).

Figura 1 - Estrutura dos elementos da WCAG 2.



Fonte: Elaborado pelo autor.

WCAG oferecem várias camadas de orientação, visando fornecer diretrizes abrangentes e adaptáveis, sendo elas:

- *Princípios gerais* - na primeira camada da Figura 1 estão os 4 princípios que fornecem a base para a acessibilidade: perceptível, operável, compreensível e robusto. Esses princípios orientam a criação de conteúdo que seja percebido por todos os usuários, que possa ser operado de diferentes maneiras, que seja fácil de entender e que seja robusto o suficiente para ser interpretado por uma variedade de agentes de usuário, incluindo tecnologias assistivas;
- *Diretrizes gerais* - sob os princípios gerais estão as diretrizes. As Diretrizes fornecem os objetivos básicos que os autores devem trabalhar para tornar seu conteúdo mais acessível a usuários com diferentes deficiências. As diretrizes não são testáveis, mas fornecem a estrutura e os objetivos gerais para ajudar os autores a compreender os critérios de sucesso e a implementar melhor as técnicas;

- *Cr terios de sucesso* - para cada diretriz, s o fornecidos cr terios de sucesso que s o test veis para permitir que as WCAG 2.1 sejam usadas onde os requisitos e os testes de conformidade s o necess rios, como em especifica es de projeto, compras, regulamenta o e acordos contratuais. Para atender  s necessidades de diferentes grupos e diferentes situa es, s o definidos tr s n veis de conformidade: A (mais baixo), AA (intermedi rio) e AAA (mais alto). Os cr terios de sucesso s o avaliados de acordo com testes t cnicos, avalia es manuais ou por meio de ferramentas de verifica o de acessibilidade automatizadas;
- *T cnicas suficientes e consultivas* - para cada uma das diretrizes e cr terios de sucesso, o grupo de trabalho tamb m documentou uma ampla variedade de t cnicas. As t cnicas s o informativas e se enquadram em duas categorias: aquelas que s o suficientes para atender aos cr terios de sucesso e aquelas que s o consultivas. As t cnicas de consultoria v o al m do que   exigido pelos cr terios de sucesso individuais e permitem aos autores abordar melhor as diretrizes. Algumas t cnicas de aconselhamento abordam barreiras de acessibilidade que n o s o abrangidas pelos cr terios de sucesso test veis.

2.3.2 Aplica es para a Internet Ricas em Acessibilidade

A tecnologia de Aplica es de Internet Ricas e Acess veis (WAI-ARIA), oferece diretrizes sobre a cria o de conte do din mico e controles avan ados de interface do usu rio (WAI-ARIA, 2023).

Os “*widgets*” s o elementos de interface de usu rio (IU), como bot es, menus, caixas de sele o e outros elementos, que devem conter informa es sem nticas adequadas para que possam ser interpretados corretamente pelas tecnologias assistivas.

Os problemas sem nticos no desenvolvimento web se manifestam quando os elementos de uma p gina n o oferecem informa es claras e significativas para as tecnologias assistivas ou para os mecanismos de busca. A

falta de semântica adequada pode resultar em uma experiência limitada para usuários com deficiências ou dificultar a indexação correta por mecanismos de pesquisa. As diretrizes propostas pelo WAI-ARIA têm como objetivo primordial orientar os desenvolvedores na criação de aplicações acessíveis por meio do HTML (*HyperText Markup Language*, ou Linguagem de Marcação de HiperTexto) e do WAI-ARIA (*Accessible Rich Internet Applications* - Aplicações de Internet Ricas e Acessíveis).

O HTML é uma linguagem de marcação usada para estruturar e organizar o conteúdo de páginas da web. Trata-se de uma linguagem fundamental para a construção de sites, permitindo a criação de estruturas e a formatação básica de documentos na internet (MDN WEB DOCS, 2023). Alguns exemplos comuns de problemas semânticos incluem:

- *Uso inadequado de marcadores* - se os marcadores HTML, como “<header>”, “<footer>”, “<nav>” e “<main>”, etc. forem utilizados incorretamente ou ausentes, confundindo a estrutura da página para usuários com deficiência visual ou dispositivos de leitura de tela.
- *Falta de descrição em imagens do marcador HTML “”* - A ausência de atributos “alt” em elementos de imagem impede que as tecnologias assistivas forneçam descrições adequadas das imagens para usuários com deficiência visual.
- *Falta de rótulos em formulários* - Quando os campos de entrada (“<input>”, do HTML) não têm rótulos claros e associados, usuários com deficiência visual podem enfrentar dificuldades para entender o propósito de cada campo.
- *Uso impróprio de títulos e subtítulos* - Os títulos (“<h1>” até o “<h6>”, do HTML) têm uma ordem adequada e hierárquica, caso não siga o padrão pode prejudicar a compreensão da estrutura da página.

O HTML5, é a quinta versão do HyperText Markup Language (HTML), desempenha um papel fundamental no desenvolvimento de aplicações web modernas. Esta versão traz consigo uma série de recursos avançados, proporcionando maior semântica, estruturação e suporte nativo a elementos

multimídia. Ao contrário de suas versões anteriores, o HTML5 introduziu tags semânticas como: “<header>”, “<footer>”, “<section>” e “<article>”, proporcionando uma estruturação mais clara e acessível ao conteúdo web. A Tabela 1 apresenta as principais tags semânticas e suas definições.

Quadro 1 - Descrição das tags HTML e suas respectivas definições.

TAG HTML	Definição
<header>	Introdução de um documento ou seção, geralmente contendo logotipo, navegação, cabeçalhos e outros elementos de identificação para a parte superior do site ou seção.
<nav>	Usado para definir um conjunto de links de navegação. Geralmente, agrupa links que direcionam para outras páginas ou seções dentro do site.
<main>	Esta tag indica o conteúdo principal da página, excluindo cabeçalhos, rodapés, barras laterais e outras seções. Deve ser usado apenas uma vez em cada página.
<section>	Utilizado para agrupar conteúdo temático relacionado em uma página. Pode conter cabeçalhos, parágrafos, imagens, entre outros elementos.
<article>	Define um conteúdo independente e autocontido na página, como um post de blog, uma notícia, um artigo, entre outros.
<aside>	Marca o conteúdo que é tangencial ou relacionado ao conteúdo da página.
<footer>	Usado para definir o rodapé da página ou de uma seção, contendo informações como informações de contato, links de redes sociais, copyright, entre outros.
<figure>	Utilizado para encapsular imagens, gráficos, ilustrações, diagramas e outros elementos visuais.

Fonte: Elaborado pelo autor.

A fim de resolver os problemas mencionados, a incorporação da WAI-ARIA no desenvolvimento de projetos web com HTML emerge como uma solução eficaz. Esta documentação oferece semântica apropriada, capacitando os desenvolvedores a criar aplicações personalizadas que se tornam acessíveis, utilizáveis e compatíveis com tecnologias assistivas, como leitores de tela. Além disso, WAI-ARIA têm compatibilidade com dispositivos móveis e capacidade de

interação com outras tecnologias, como CSS3 (*Cascading Style Sheets* ou traduzindo para o português, Folha de Estilo em Cascatas) e *JavaScript*, tornam-no uma base sólida para o desenvolvimento de interfaces web dinâmicas e responsivas, priorizando a acessibilidade e usabilidade.

A semântica no WAI-ARIA é dividida em dois componentes principais: *roles*, que definem com que tipo de elemento o usuário está interagindo, e *states/properties* (estados e propriedades respectivamente), que, suportados pelas *roles*, descrevem o estado daquele elemento.

A WAI-ARIA identifica os tipos de aplicações e suas estruturas, fornecendo informações sobre os papéis correspondentes de cada componente para facilitar sua indexação na página e o fornecimento de conteúdo para tecnologias assistivas. Isso permite o uso de elementos com papéis diferentes dos propostos pelo HTML, interpretando-os como elementos específicos ou estruturais na aplicação desenvolvida. Os elementos *roles* e *properties* são apresentados nas próximas seções.

2.3.2.1 Roles

Roles (funções) desempenham um papel essencial na WAI-ARIA, fornecendo uma maneira de descrever semanticamente o propósito de um elemento na interface do usuário. Elas definem o tipo de componente que está presente na página web, permitindo que tecnologias assistivas entendam e interpretem corretamente o conteúdo exibido. Por exemplo, alguns *roles* comuns incluem "*button*" (botão), "*link*" (link), "*checkbox*" (caixa de seleção), "*slider*" (controle deslizante), entre outras. A Tabela 2 apresenta as principais *tags* HTML com suas respectivas funções ARIA e objetivos.

Quadro 2 - Apresentação das tags HTML e sua respectiva role.

HTML	ARIA Role	Objetivo
header	role="banner"	Exibir o título da página
nav	role="navigation"	Coleção de <i>links</i> adequados para uso ao navegar no documento ou em documentos relacionados

main	role="main"	Responsável pelo conteúdo principal ou central do documento.
footer	role="contentinfo"	Informações sobre o documento principal: notas de rodapé/direitos autorais/links para declaração de privacidade
aside	role="complementary"	Suporta o conteúdo principal, mas é separado e significativo em seu próprio conteúdo
Não possui tag HTML disponível	role="search"	Esta seção contém a funcionalidade de pesquisa do aplicativo
form	role="form"	Coleção de elementos associados a formulários
section	role="region"	Conteúdo que é relevante e para o qual os usuários provavelmente desejarão navegar. O rótulo deve ser fornecido para este elemento

Fonte: Elaborado pelo autor.

Cada *role* oferece um significado específico para o elemento, ajudando a transmitir sua função e interação esperada aos usuários com deficiência. A seguir, serão apresentadas algumas figuras exemplificando a aplicação prática das *roles* na criação de interfaces acessíveis. A Figura 2 apresenta elementos para criar um formulário HTML semântico, usando os elementos “<form>”, “<input>”, “<textarea>” e “<button>”.

Figura 2 - Utilização da role search juntamente com tag HTML.

```
<form role="search">
  <label for="search" class="hidden-visually">Search: </label>
  <input type="text" name="search" id="search" v-model="search" />
  <button type="submit">Search</button>
</form>
```

Fonte: Elaborado pelo autor.

A Figura 3 apresenta títulos descritivos para cada seção e seguindo o padrão hierárquico, aninhando os títulos em sua ordem de classificação: “<h1>” - “<h6>”. Utilizando tags de título reais em vez de estilizar para dar a aparência visual dos títulos.

Figura 3 - Utilização da role main juntamente com tag HTML.

```
<main role="main" aria-labelledby="main-title">
  <h1 id="main-title">Main title</h1>
  <section aria-labelledby="section-title-1">
    <h2 id="section-title-1"> Section Title </h2>
    <h3>Section Subtitle</h3>
    <!-- Content -->
  </section>
```

Fonte: Elaborado pelo autor.

Outro elemento importante, relacionado às *roles*, são os *roles landmarks*. Elas são elementos-chave na acessibilidade da web, oferecendo referências estruturais que ajudam os usuários a navegar eficientemente por uma página. Elas atuam como pontos de referência, identificando seções distintas de uma interface, como cabeçalhos, barras laterais, áreas de navegação, conteúdo principal e rodapés. Esses *landmarks* permitem que os usuários com deficiência visual ou cognitiva tenham uma compreensão clara da organização da página, facilitando a navegação por meio de leitores de tela e outros dispositivos assistivos. Conforme demonstrado na Figura 3, as *roles landmarks* desempenham um papel fundamental ao identificar e demarcar áreas cruciais na página, facilitando a localização pelos usuários de blocos de informações essenciais.

A utilização adequada desses *roles* contribui significativamente para uma experiência de usuário mais inclusiva e acessível. Para ilustrar melhor, serão apresentadas a seguir algumas figuras exemplificando a aplicação prática das *roles landmarks* na estruturação de interfaces acessíveis.

Figura 4 - Exemplo de página HTML com os roles do WAI-ARIA.

```
<header role="banner">
  <h1>Meu site</h1>
</header>

<nav role="navigation">
  <ul>
    <li><a href="#">Início</a></li>
    <li><a href="#">Sobre</a></li>
    <li><a href="#">Contato</a></li>
  </ul>
</nav>

<main role="main">
  <article>
    <h2>Meu conteúdo</h2>
    <p>Este é o conteúdo principal da página.</p>
  </article>
</main>
```

Fonte: Elaborado pelo autor.

Neste exemplo, os elementos *header*, *nav* e *main* estão recebendo os *roles banner*, *navigation* e *main*, respectivamente. Esses roles indicam a importância de cada elemento para a estrutura da página.

Aqui está um exemplo de como utilizar roles do WAI-ARIA para identificar widgets:

Figura 5 - Exemplo do uso do WAI-ARIA em campos de entrada do HTML.

```
<input type="text" role="textbox" aria-label="Campo para inserir o nome">
<button role="button" aria-label="Botão para pesquisar">Pesquisar</a>
<select role="combobox" aria-label="Selecione um item">
  <option value="1">Item 1</option>
  <option value="2">Item 2</option>
  <option value="3">Item 3</option>
</select>
```

Fonte: Elaborado pelo autor.

Neste exemplo, os elementos *input*, *button* e *select* estão recebendo os *roles textbox*, *button* e *combobox*, respectivamente. Esses roles indicam a função de cada elemento para o usuário.

2.3.2.2 Properties

Properties referem-se às características que descrevem os diferentes estados dos componentes presentes em uma interface, tais como *checked* para uma caixa de seleção ou *haspopup* para um menu. Essas propriedades desempenham um papel crucial não apenas para definir estados específicos dos elementos, mas também ao permitir a identificação de áreas que sofrem alterações dinâmicas na página. A integração das *properties* com tecnologias como o *JavaScript* (às vezes abreviado para *JS*), que é uma linguagem leve, interpretada e baseada em objetos com funções de primeira classe possibilita a atualização dinâmica de informações na página em tempo real (MDN WEB DOCS, 2022). Essas atualizações podem se manifestar de diversas formas, desde caixas de diálogo até modificações nos campos contidos na própria página, ampliando consideravelmente a experiência interativa do usuário.

A Figura 6 exemplifica o uso das *properties*, o atributo *role* está sendo usado para definir o papel do elemento *div* como uma caixa de diálogo. Os atributos *aria-labelledby* e *aria-describedby* estão sendo usados para fornecer informações sobre o rótulo e a descrição da caixa de diálogo, uma informação complementar acessível, demonstrando a aplicação prática e eficaz das na construção de interfaces acessíveis e dinâmicas.

Figura 6 - Exemplo de código com informação adicional acessível.

```
<div role="dialog" aria-labelledby="dialog-label" aria-describedby="dialog-description">  
  <h2 id="dialog-label">Caixa de diálogo</h2>  
  <p id="dialog-description">Esta é uma caixa de diálogo.</p>  
</div>
```

Fonte: Elaborado pelo autor.

2.4 CONSIDERAÇÕES FINAIS

Os tópicos abordados neste capítulo estão intrinsecamente relacionados com o cerne deste trabalho, desde a análise das diferentes deficiências até as tecnologias fundamentais e diretrizes para a adaptação de um aplicativo PWA voltado para auxiliar no aprendizado de lógica de programação para os deficientes visuais, especialmente por meio de blocos físicos 3D. Esta revisão se mostra

essencial, pois dela resultará um aplicativo que busca facilitar e proporcionar uma experiência de aprendizado acessível e de qualidade para usuários com necessidades específicas.

Com base na fundamentação teórica apresentada, é possível concluir que a acessibilidade web é um tema importante que deve ser considerado no desenvolvimento de sites e aplicativos. Ao seguir as diretrizes de acessibilidade, é possível garantir que pessoas com deficiência tenham acesso a esses recursos e possam participar da sociedade de forma plena e inclusiva.

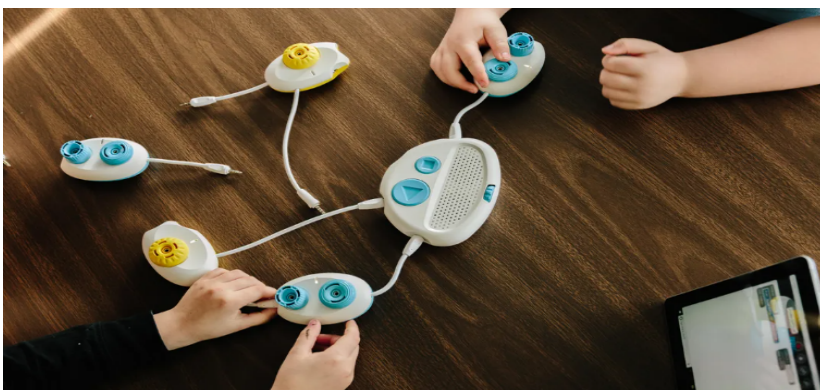
No próximo capítulo, serão apresentados estudos relacionados ao desenvolvimento de ferramentas para o auxílio do aprendizado de lógica de programação para deficientes visuais.

3 TÉCNICAS DE ENSINO DE PROGRAMAÇÃO PARA CEGOS

Neste capítulo, são apresentadas técnicas encontradas para o ensino de programação para pessoas com deficiência visual. O Code Jumper (2023) é uma ferramenta educacional desenvolvida pela *Microsoft*, projetada com peças físicas modulares que permitem que alunos criem código com ênfase na acessibilidade. O dispositivo central, denominado *Hub*, possui dois botões - um para reprodução e outro para interrupção, além de alto-falantes e controle de volume.

Os "*Pods*" são componentes plásticos conectados ao *Hub* por portas, cada um representando uma linha de código. Esses *Pods* possuem suas próprias portas, permitindo a conexão de novos *Pods*, possibilitando a construção contínua do código. A Figura 7 abaixo ilustra um fluxo das conexões entre os componentes com o uso do do *Code Jumper*.

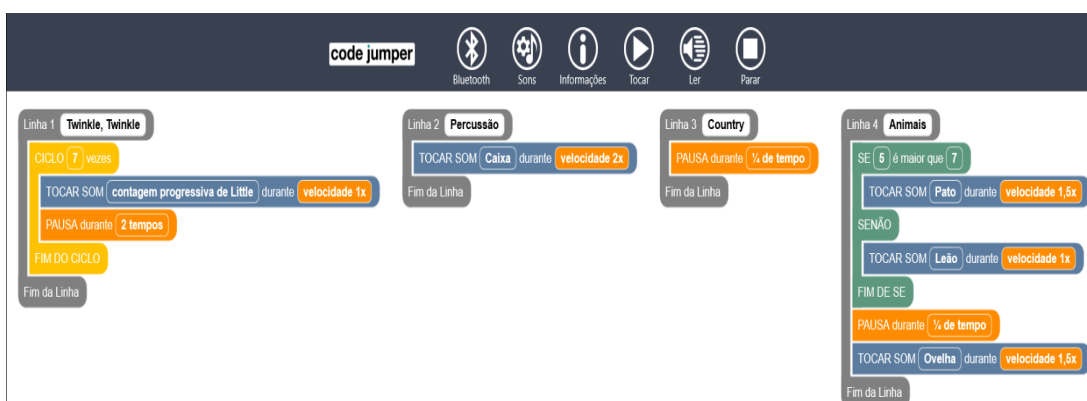
Figura 7 - Peças físicas modulares do code jumper.



Fonte: CODE JUMPER, 2023.

A Figura 8 abaixo ilustra um fluxo de código com o uso do do *Code Jumper*.

Figura 8 - Blocos de código a partir da montagem dos pods e hub.



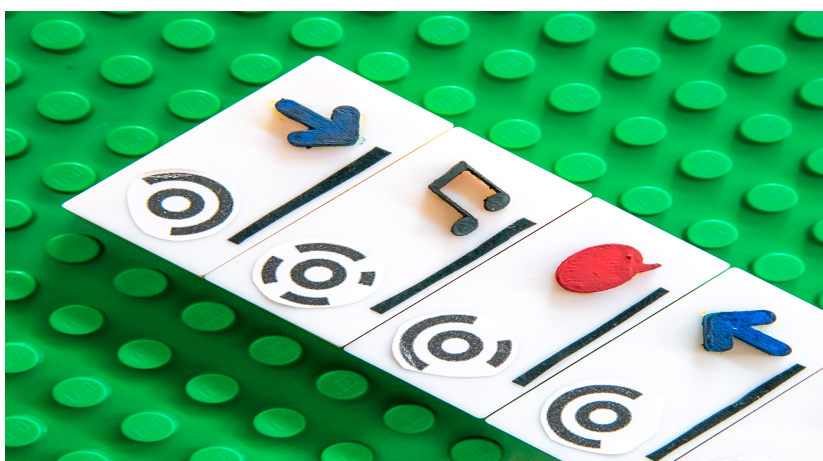
Fonte: PORVIR, 2023.

O *Block-based Accessible Tangible Programming Systems* (BATS) é um projeto inovador que combina blocos físicos e software de computador para ensinar programação a crianças com deficiência visual.

Já os blocos do BATS são pré-definidos em cores e ações, as quais são traduzidas no comportamento de personagens ou jogos no ambiente virtual. Esse sistema tangível e acessível oferece uma forma sensorial de aprender programação, facilitando o entendimento e a interação para crianças com baixa visão ou cegueira.

A Figura 9 apresenta alguns dos blocos desenvolvidos no trabalho, cada qual com uma funcionalidade específica.

Figura 9 - Blocos físicos montados do BATS .



Fonte: JUNQUEIRA, 2023.

O trabalho desenvolvido PEREIRA *et al.* (2023), teve como objetivo investigar as possíveis contribuições de um jogo didático adaptado para abordar a lógica de programação com estudantes cegos. Trata-se de um jogo de tabuleiro que tem como proposta oferecer uma abordagem divertida para o aprendizado de Lógica de Programação, composto por: um tabuleiro, setenta e duas cartas de instrução, vinte cartelas-bateria, oito cartas missão e quatro peões.

Todos os componentes do jogo foram adaptados para o público alvo, onde o tabuleiro foi construído com 4 tipos de EVA's texturizados e os peões foram adaptados com pedras decorativas, onde cada peão têm uma quantidade específica de pedras, para sua identificação a partir do tato.

Os participantes começam posicionando as cartelas-bateria ao redor do tabuleiro e escolhendo uma carta missão. Utilizando cartas de movimento, eles avançam no jogo, seguindo as instruções fornecidas em cada carta. Ao longo do jogo, aplica os princípios da lógica de programação para resolver desafios, como adquirir cartelas-bateria compatíveis com sua missão.

O jogo foi desenvolvido para crianças a partir de 7 anos, o jogo pode ser jogado por 2 a 4 jogadores, proporcionando uma experiência educativa e divertida no aprendizado de conceitos fundamentais de programação.

Figura 10 - Alunos jogando o jogo de tabuleiro.



Fonte: PEREIRA *et al.*, 2023.

Os resultados da pesquisa demonstraram que o uso de jogos de tabuleiro para o ensino de lógica de programação a estudantes com deficiência visual é uma abordagem promissora. A incorporação desses materiais adaptados e adequadamente utilizados em sala de aula não apenas promove a inclusão dessas pessoas nas atividades, mas também contribui para aprimorar a qualidade do ensino, tornando-o mais acessível e eficaz. Oferecer aos estudantes um ensino contextualizado e adaptado às suas necessidades é uma forma eficaz de melhorar a efetividade do processo de ensino-aprendizagem.

Embora a pesquisa seja robusta, a inclusão de um software na metodologia da pesquisa pode trazer melhorias específicas, pois aproxima o processo de ensino-aprendizagem do mundo computacional.

Essas abordagens representam avanços significativos no campo do ensino de programação para pessoas com deficiência visual. No entanto, há espaço para investigações adicionais e desenvolvimento de metodologias mais abrangentes e personalizadas para atender às necessidades específicas desse público-alvo

De forma geral, as pesquisas sobre o ensino de lógica de programação para estudantes com deficiência visual geralmente abordam outros assuntos, como a inclusão social e a acessibilidade. Além disso, elas geralmente têm como público alvo crianças e acadêmicos com deficiência visual que frequentam a escola ou ensino superior.

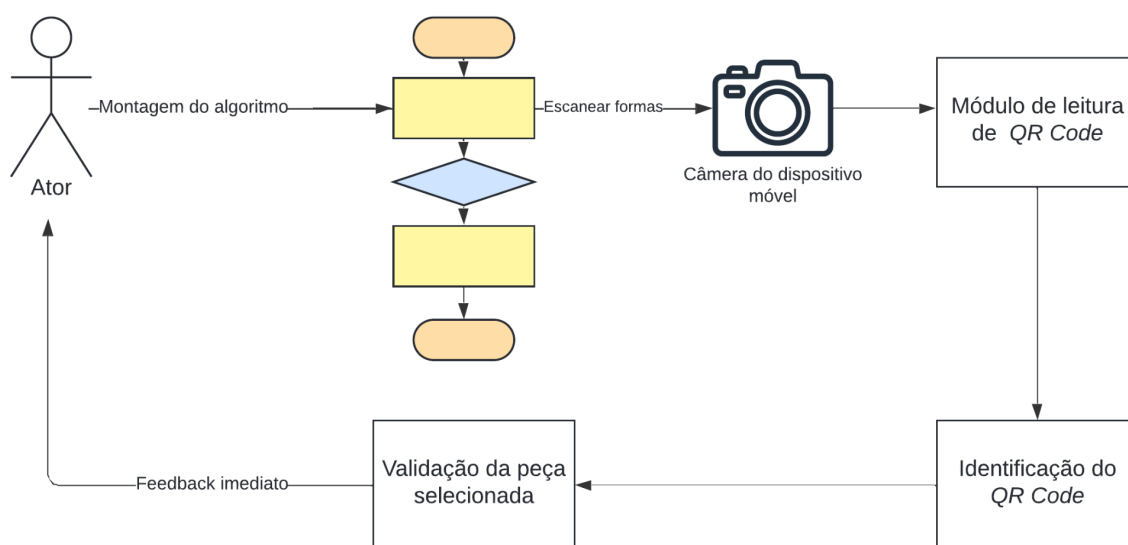
No entanto, a pesquisa em questão apresenta uma abordagem diferente, pois utiliza blocos físicos 3D que representam blocos de código adaptados para ensinar lógica de programação juntamente com um aplicativo a estudantes com deficiência visual, tanto do ensino técnico integrado quanto do curso superior do Instituto Federal de Alagoas.

4 Echo 3D Blocks

Neste capítulo, é apresentada a aplicação desenvolvida. O *Echo 3D Blocks* - E3B é um *Progressive Web App* (PWA) e tem como objetivo principal oferecer suporte interativo para o ensino de lógica de programação, direcionado especialmente a pessoas com deficiência visual. Utilizando blocos físicos 3D associados a *QR codes*, o E3B permite que os usuários montem algoritmos de maneira simples, com o objetivo de tornar a aprendizagem da programação mais intuitiva. O projeto BATS apresenta uma abordagem semelhante à proposta apresentada neste trabalho, mas utiliza diferentes tipos de blocos físicos e significados para eles. Os blocos físicos 3D apresentados nessa pesquisa se assemelham a partes de um algoritmo em um fluxograma.

O E3B pode ser acessado através do seguinte endereço: <https://echo-3d-blocks.web.app/#/>.

Figura 11 - Representação gráfica do fluxo da metodologia proposta.



Fonte: Elaborado pelo autor.

A Figura 11 ilustra as atividades envolvidas no uso da aplicação pelo usuário, para facilitar a compreensão, os passos envolvidos são apresentados a seguir:

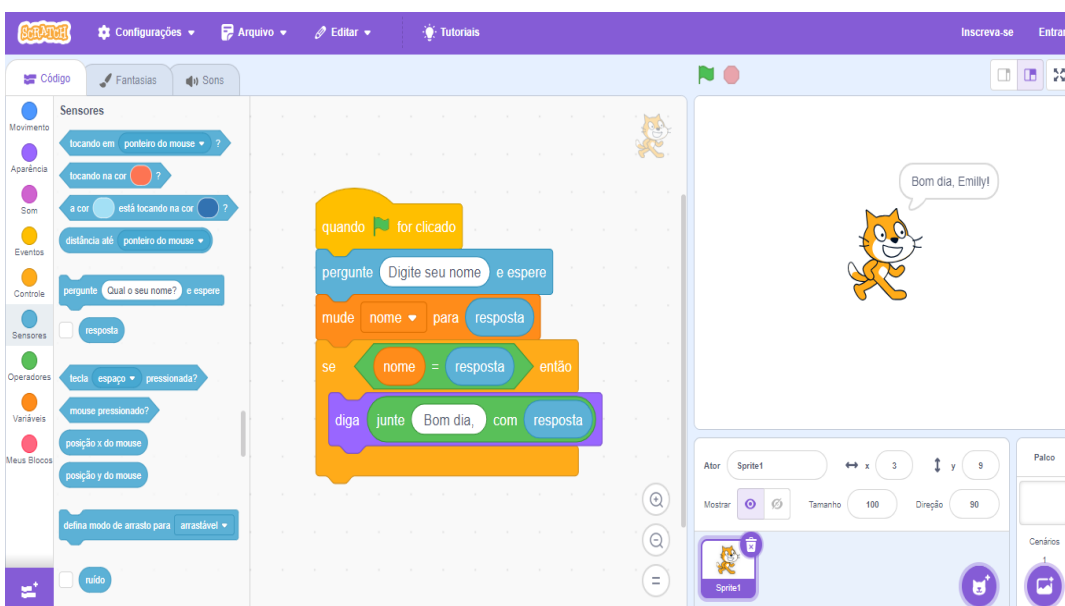
1. O usuário seleciona uma questão de lógica de programação;
2. Em seguida, o usuário monta o algoritmo a partir dos blocos de código físicos;

3. Para validar o algoritmo, neste passo, o usuário utiliza o aplicativo para escanear os *QR codes* associados aos blocos;
4. O aplicativo processa as informações dos *QR codes* e verifica a disposição correta dos blocos;
5. O aplicativo válida não apenas as posições físicas dos blocos, mas também a precisão dos tipos de blocos utilizados no algoritmo;
6. Por fim, para cada bloco, o aplicativo emite um *feedback* sonoro distinto. Esse *feedback* sonoro ajuda o usuário a identificar os blocos e a compreender a estrutura do algoritmo.

4.1 FLUXOGRAMA FÍSICO 3D

Este trabalho utilizou como base os blocos 3D criados em um projeto de iniciação científica. Este projeto fez a criação das blocos, que representam os elementos essenciais do código, tais como loops, condicionais, variáveis e comandos, com inspiração nos componentes do programa Scratch (SCRATCH, 2023). A Figura 12 apresenta a interface principal do Scratch, onde no menu lateral é disponibilizado as opções de blocos e uma lista de blocos de código para os usuários montarem seus próprios algoritmos.

Figura 12 - Tela principal do Scratch.



Fonte: SCRATCH, 2023.

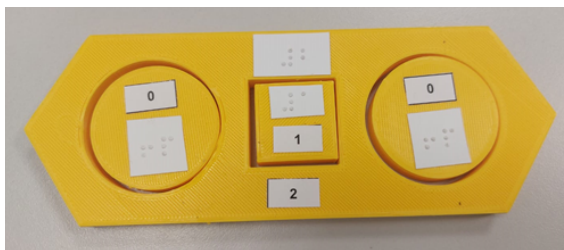
O Scratch é reconhecido como uma linguagem de programação de sintaxe simples, baseada em blocos de montar, destinada à criação de projetos compartilháveis e aprimorados por meio da comunidade de desenvolvedores. De acordo com Scaico et al. (2013), o ambiente oferecido pelo Scratch representa uma linguagem que não apenas facilita o aprendizado de programação, mas também introduz um conceito inovador ao desenvolvimento de código, focando na Computação Criativa.

O Scratch foi especificamente projetado para fomentar uma abordagem de aprendizagem interativa, experimental e lúdica. Esta metodologia encoraja os estudantes a constantemente reavaliar suas decisões, explorar novos caminhos e imaginar possibilidades criativas durante o processo de programação (SCAICO *et al*, 2013). Esta plataforma oferece um ambiente inclusivo e envolvente, incentivando o desenvolvimento do pensamento crítico, criatividade e resolução de problemas através da programação, ampliando assim as oportunidades de aprendizagem no ensino de programação.

Um exemplo de alguns dos blocos criados, é exibido na Figura 13. Os blocos representam as seguintes estruturas de um algoritmo: Encaixe de variável (apresentado na imagem com o número 0), Encaixe de operador lógico ou aritmético (apresentado na imagem com o número 1) e Comparador (apresentado na imagem com o número 2), além da numeração, em cada um dos blocos estão presentes os adesivos em *Braille* para ajudar na identificação de cada elemento. Detalhes deste projeto podem ser encontrados em Vieira et al. (2023).

Os significados e características de cada bloco será melhor abordado na próxima seção deste capítulo.

Figura 13 - Bloco impresso na impressora 3D e utilizando adesivos com *Braille*.



Fonte: Vieira et al, 2023.

Essa representação visual demonstra a estratégia adotada para garantir a identificação tátil de informações, proporcionando acessibilidade para pessoas com

deficiência visual no contexto do projeto. Os blocos proporcionam uma oportunidade única de manipulação física e tátil, oferecendo aos alunos uma experiência prática e concreta no universo da programação.

Como resultado, os estudantes adquirem uma compreensão mais sólida das estruturas fundamentais de lógica de programação. Essa abordagem não só torna os conceitos complexos mais acessíveis e compreensíveis, mas também os torna mais adequados para aqueles com deficiência visual, criando uma forma inclusiva de aprendizado.

No projeto, foram utilizadas estruturas de lógica de programação como:

- *Condicionais* - para criar caminhos específicos de execução do algoritmo, de acordo com determinadas condições.
- *Comparadores* - são usados para realizar comparações entre valores
- *Operadores lógicos* - são empregados para combinar diferentes condições.
- *Operadores matemáticos* - para realizar operações aritméticas básicas.
- *Entradas (inputs) e saídas (outputs)* - permitindo que os usuários forneçam informações ao programa e recebam resultados ou feedbacks correspondentes.

No entanto, o projeto não utiliza estruturas de repetição ou outras estruturas mais avançadas de programação.

4.2 USO DE QR CODES

A integração do fluxograma 3D físico com o aplicativo envolveu a inserção estratégica de *QR codes* em cada bloco/peça criada. Os *QR codes* fornecem identificadores únicos para cada bloco, permitindo uma identificação rápida e precisa por meio do aplicativo, proporcionando uma experiência intuitiva para os usuários.

Esses códigos foram essenciais para permitir que os usuários, que por meio de câmera, pudessem identificar logicamente os blocos no sistema. A Figura 14 ilustra uma peça com a inserção do QR Code.

Figura 14 - Bloco impresso utilizando adesivo com *Braille* e o QR code.



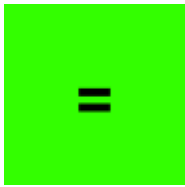





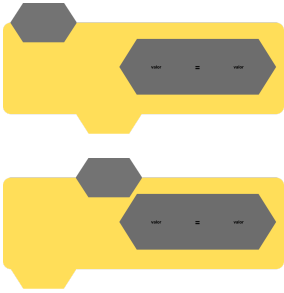





Fonte: Elaborado pelo autor.

Os códigos representam identificadores únicos atribuídos a cada bloco, sendo selecionada uma numeração específica para identificá-las. Esta numeração foi reproduzida em *Braille* em alto relevo nos blocos 3D. Essa abordagem permite que as pessoas com deficiência visual, possam facilmente identificar os blocos e executar as etapas e procedimentos para o aprendizado de lógica de programação.

Além disso, o sistema foi desenvolvido de forma a oferecer retornos sonoros imediatos ao usuário, indicando quando um bloco é identificado corretamente ou quando uma seleção inadequada é feita. A Tabela 2 contém as informações de cada bloco utilizado no projeto, com nomes, imagens representativas, descrições e os respectivos *QR codes* associados.

Quadro 3 - Descrição dos blocos e seus respectivos QR codes.

Nome das blocos	Imagem das blocos	Descrição das blocos	QR codes
Encaixe de variável		A função deste bloco é associar um valor a uma variável, ou seja, responsável por guardar um valor em uma variável.	
Encaixe de operador		A função deste bloco é guardar um comparador lógico e aritmético.	
Comparador		A função deste bloco é se encaixar em estruturas e comparar os valores por meio de um operador lógico	
Terminador		A função deste bloco é de tanto iniciar o código, como finalizar	
Condicional		A função deste bloco é equivalente ao 'if' e ao 'else', pois realiza um bloco de comandos caso a condição no comparador seja verdadeira ou falsa.	

Processo		A função deste bloco é executar uma determinada ação, manipular os dados ou fornecer algum tipo de resultado para o usuário	
----------	---	---	---

Fonte: Elaborado pelo autor.

4.3 FERRAMENTAS UTILIZADAS

As principais ferramentas utilizadas foram os frameworks VUE.JS (2023) e QUASAR FRAMEWORK (2023), a linguagem de programação JavaScript e a linguagem de marcação HTML5, reconhecidos por oferecerem uma ampla gama de recursos para o desenvolvimento ágil de interfaces web responsivas e aplicativos multiplataforma.

O JavaScript é uma linguagem de programação amplamente utilizada no desenvolvimento web, foi fundamental para a criação de interfaces interativas e dinâmicas. Sua versatilidade e potencial na construção de aplicações para a web são amplamente reconhecidos na comunidade de desenvolvimento de software (FLANAGAN; NOVAK, 1998).

Vue.js3, é um framework JavaScript progressivo para a construção de interfaces de usuário. Ele oferece uma abordagem flexível e modular para o desenvolvimento front-end, permitindo a criação de aplicativos web escaláveis e de fácil manutenção (VUE.JS, 2023). Destaca-se por sua capacidade de renderização reativa e arquitetura de componentes, o que o torna uma escolha popular entre os desenvolvedores web.

Além disso, o framework Quasar complementa o ecossistema de desenvolvimento ao oferecer um conjunto abrangente de componentes e ferramentas, permitindo a construção de aplicativos web responsivos e progressivos (QUASAR FRAMEWORK, 2023). Sua natureza de código aberto e a vasta gama de funcionalidades pré-criadas o tornam uma opção atraente para o desenvolvimento eficiente e a implementação ágil de interfaces de usuário sofisticadas.

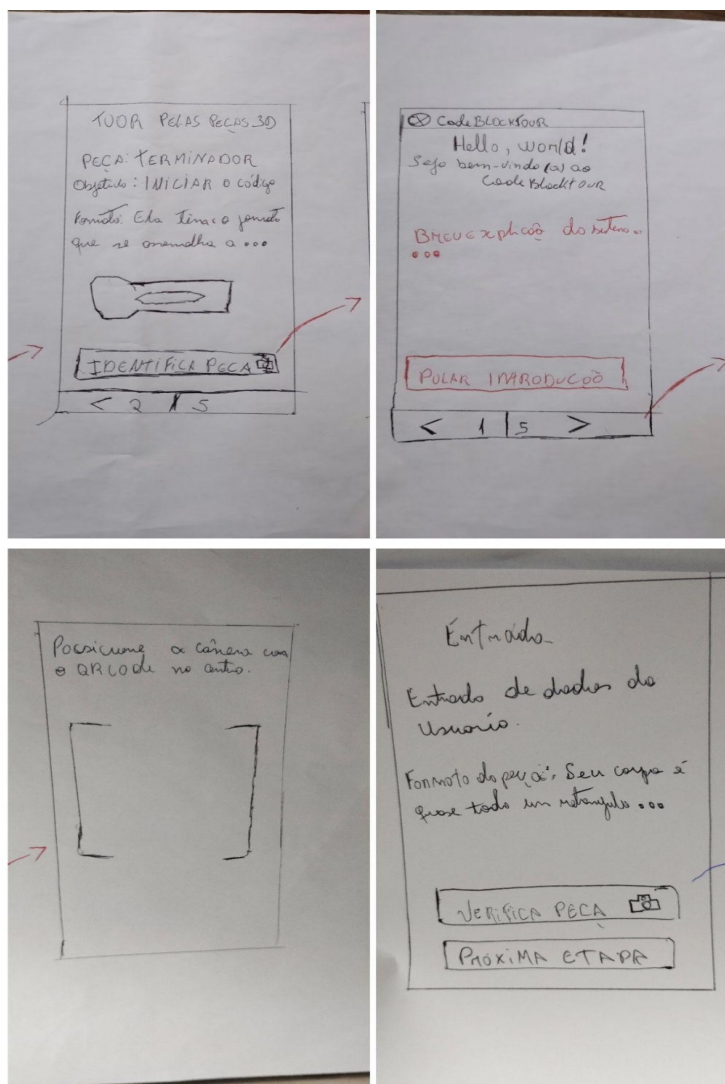
A combinação do JavaScript, Vue.js3 e Quasar oferece aos desenvolvedores uma poderosa pilha de tecnologia para a criação de aplicações

web modernas, eficazes e acessíveis. A sinergia entre essas ferramentas proporciona flexibilidade, desempenho e a capacidade de atender aos requisitos complexos do desenvolvimento do software PWA proposto.

4.4 PROTOTIPAGEM

A etapa de prototipagem é crucial no desenvolvimento de projetos, permitindo a construção de modelos iniciais que possibilitam testar ideias, funcionalidades e interações do produto ou sistema a ser desenvolvido. Neste contexto, foram explorados dois tipos de protótipos: o de baixa fidelidade e o de média fidelidade. O protótipo de baixa fidelidade foi concebido como uma representação inicial do projeto, focando na sua estrutura básica e funcionalidades principais. A Figura 15 apresenta o protótipo de baixa fidelidade das telas principais.

Figura 15 - Protótipo de baixa fidelidade das páginas da aplicação.



Fonte: Elaborado pelo autor.

Na Figura 15 é apresentado como foi desenhada a ideia de como ficaria o projeto, tanto como a disposição dos componentes de interface do usuário, quanto algumas das principais funcionalidades.

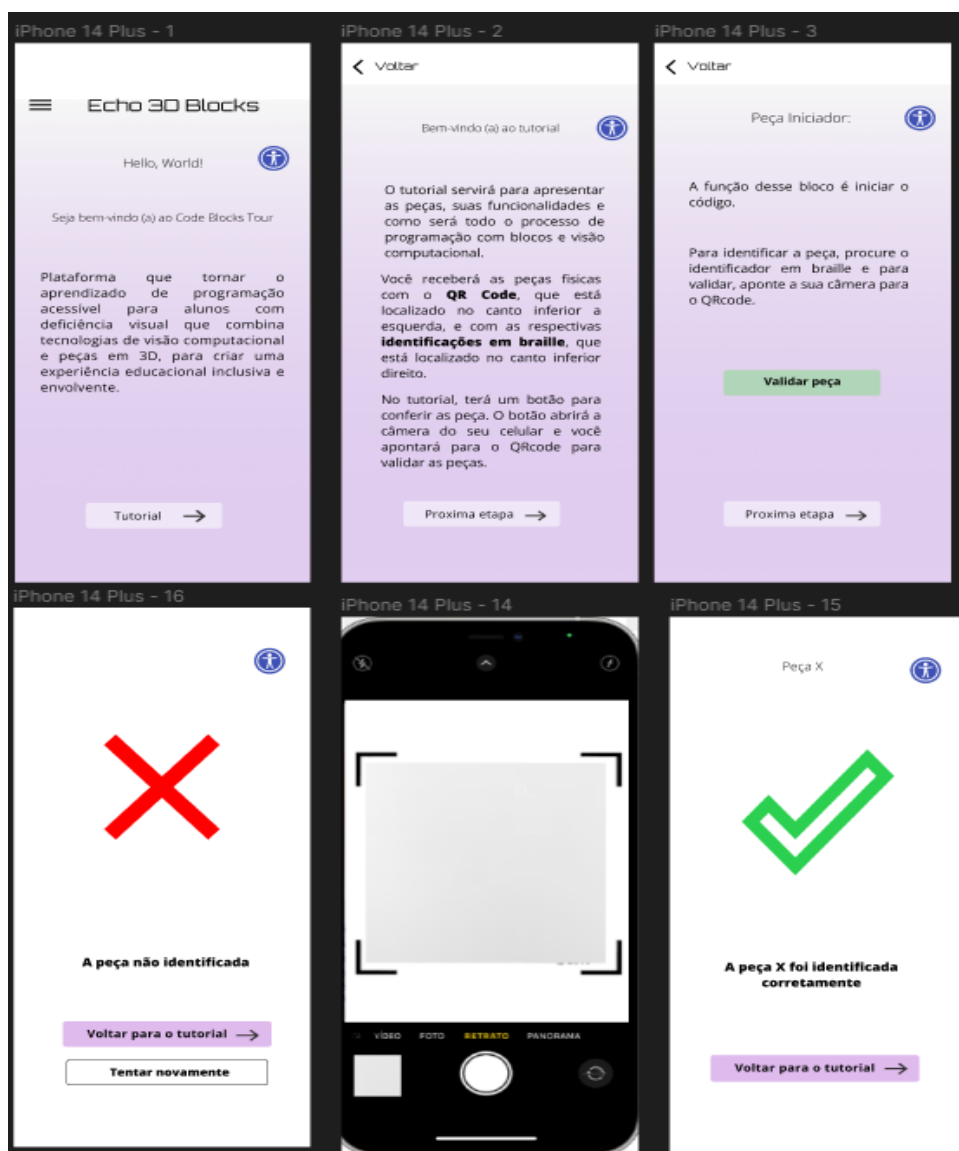
Esse protótipo é caracterizado por ser simples, rápido e de baixo custo, frequentemente desenvolvido por meio de esboços, papel, cartolina, ou utilizando ferramentas digitais como softwares de prototipagem simples. Sua principal finalidade é validar conceitos, fluxos de navegação e funcionalidades essenciais do projeto de forma rápida e interativa. Esse tipo de protótipo permite realizar ajustes e modificações de maneira ágil, sendo uma ferramenta valiosa para a coleta inicial de feedbacks e identificação de problemas conceituais.

Após a ideação e validação, foi realizada a prototipação em média fidelidade. O protótipo de média fidelidade busca uma representação mais próxima do produto final em termos de aparência e funcionalidades. Este protótipo incorpora elementos mais detalhados, cores, interações básicas e, ocasionalmente, funcionalidades limitadas.

Esses protótipos são geralmente desenvolvidos utilizando ferramentas de prototipagem digital mais avançadas, como software de design e mockups interativos como o FIGMA (2023) que foi utilizado nessa pesquisa, o protótipo de média fidelidade permite uma visualização mais realista e uma experiência mais próxima do produto final. Ele é útil para refinamento de design, testes de usabilidade mais aprofundados e para apresentação a potenciais usuários ou stakeholders.

A Figura 16 apresenta o protótipo de média fidelidade, foram feitas diversas telas utilizando a ferramenta Figma para que pudessem servir como referência na hora do desenvolvimento final da aplicação.

Figura 16 - Protótipo de média fidelidade das páginas.



Fonte: Elaborado pelo autor.

Ambos os tipos de protótipos desempenham papéis distintos e complementares no processo de desenvolvimento, contribuindo para o aprimoramento gradual do projeto, desde a validação inicial de conceitos até a representação mais próxima do produto final, permitindo assim uma evolução mais eficiente e controlada do projeto.

4.5 FUNCIONALIDADES

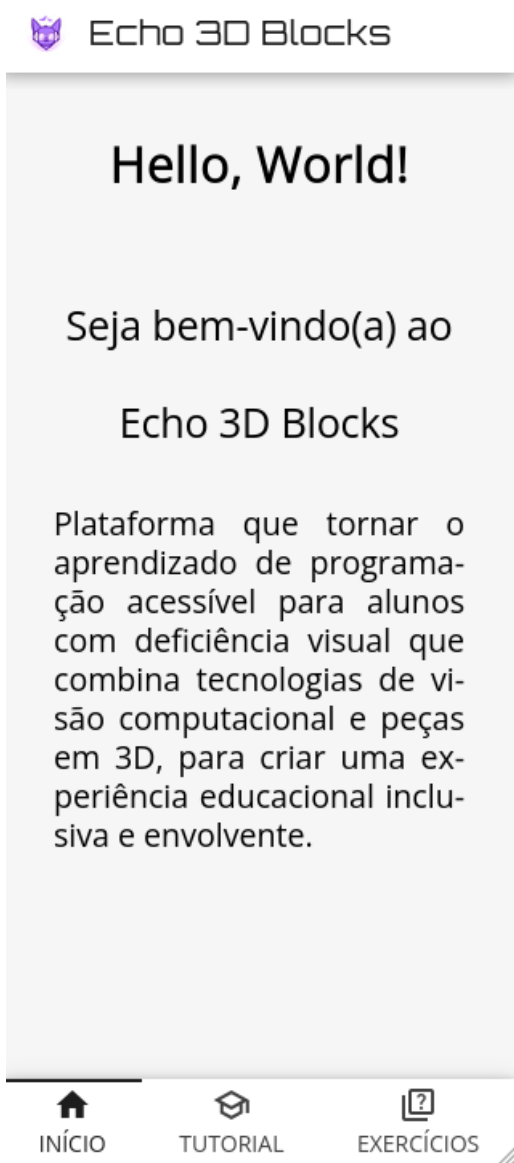
A interface do *E3B* foi projetada para ser clara e amigável para pessoas com deficiência visual. Buscando oferecer uma experiência intuitiva e acessível ao

processar as informações dos blocos físicos e fornecer feedback sonoro sobre a validação dos algoritmos. Além disso, para garantir a acessibilidade de pessoas com daltonismo no contexto do projeto, o sistema foi desenvolvido considerando essas limitações na percepção de cores. As notificações do sistema serão projetadas com ícones e mensagens descritivas que não dependem exclusivamente de cores para transmitir informações.

As mensagens serão elaboradas de forma a proporcionar clareza e entendimento, transmitindo informações sobre acertos, erros, mensagens positivas ou negativas de maneira explícita, independente da percepção cromática do usuário. Esse cuidado visa assegurar que o sistema seja igualmente acessível e compreensível para todas as pessoas, independentemente de suas limitações na percepção de cores devido ao daltonismo.

A Figura 17 apresenta a tela inicial da aplicação, com uma breve apresentação do aplicativo no centro e um menu de navegação na parte inferior, que permite que o usuário acesse as páginas: Início, Tutorial e Exercício.

Figura 18 - Apresenta a tela inicial do aplicativo.



Fonte: Elaborado pelo autor.

As funcionalidades criadas foram:

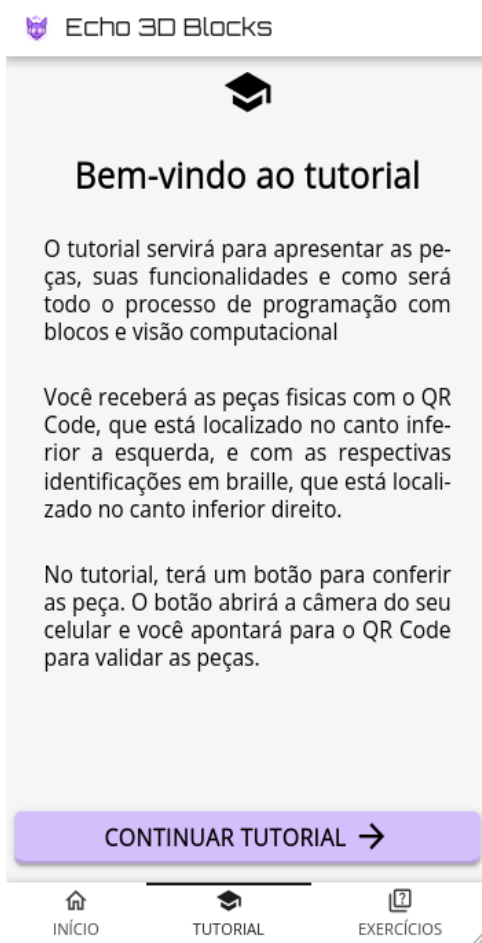
- **Processamento dos Blocos 3D:** O aplicativo é responsável por interpretar os códigos QR dos blocos 3D, processando a disposição e os tipos dos blocos para validar o algoritmo.
- **Verificação de Algoritmos:** Após o escaneamento, o software analisa as informações dos blocos para verificar se a

disposição e os tipos estão de acordo com a lógica do algoritmo proposto.

- **Feedback de Validação:** Fornece feedback imediato ao usuário, indicando a precisão e corretude do algoritmo montado com os blocos físicos.

Ao clicar na opção *Tutorial* do menu inferior o usuário é redirecionado para a primeira página do conjunto de páginas da seção de tutorial da aplicação. A Figura 18 apresenta a ilustração da primeira página do Tutorial, explicando o funcionamento deste.

Figura 18 - Apresenta a tela de tutorial do aplicativo.



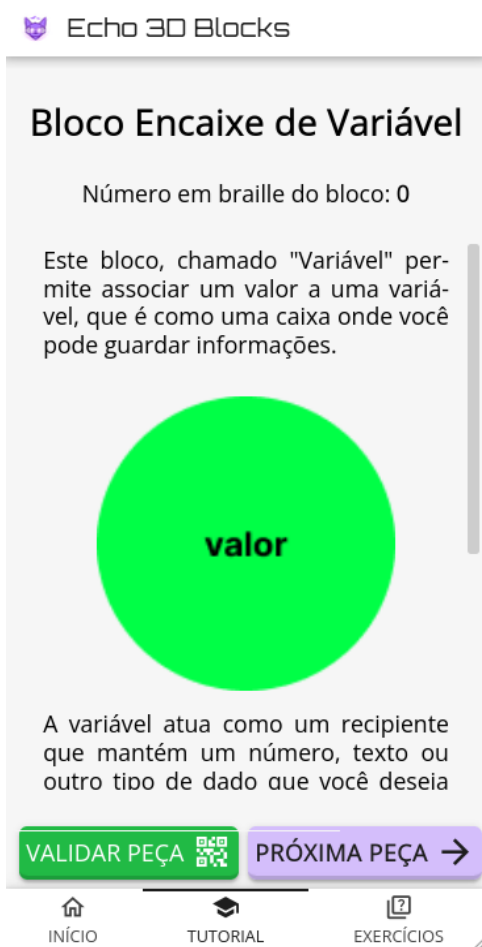
Fonte: Elaborado pelo autor.

Cada uma das páginas seguintes do tutorial será dedicada a explicar um dos blocos físicos 3D. Essas páginas não apenas oferecerão uma descrição e instruções de uso para cada bloco, mas também fornecerão a funcionalidade de

identificá-lo. Para isso, basta acessar o botão *Validar Bloco*, também localizado na parte inferior

da página. A Figura 19 apresenta o tutorial de como o bloco *Encaixe de Variável* é utilizado.

Figura 19 - Página da aplicação com a explicação do bloco variável.



Fonte: Elaborado pelo autor.

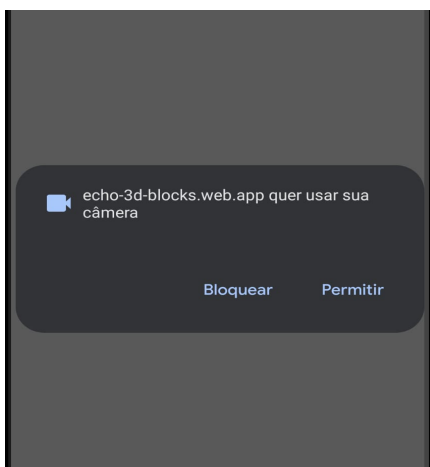
Graças à implementação das diretrizes de acessibilidade no HTML da aplicação, o usuário deve conseguir percorrer as várias páginas com facilidade usando seu leitor de tela acessível. Essa navegação possibilita a identificação clara de cada componente presente, permitindo que ele ouça uma descrição adequada do conteúdo de cada elemento da página.

Para o usuário validar um bloco físico 3D, deve selecionar o botão *Validar Peça* na parte inferior da página. Ao selecionar, a câmera do dispositivo do usuário

será ativada. A validação é realizada utilizando a câmera do dispositivo móvel do usuário.

No entanto, os navegadores realizam uma etapa de segurança antes de permitir o acesso à câmera. Na primeira vez que o usuário acessa a aplicação Echo 3D blocks, é necessário que ele confirme se deseja habilitar o uso da câmera do seu dispositivo móvel. A Figura 20 apresenta um exemplo deste questionamento.

Figura 20 - Exemplo de popup de navegadores para habilitar o uso da câmera.



Fonte: Elaborado pelo autor.

A Figura 21 ilustra a integração da câmera do dispositivo do usuário com a funcionalidade desenvolvida para a leitura de códigos QR.

Figura 21 - Utilização da câmera do dispositivo do usuário.



Fonte: Elaborado pelo autor.

A Figura 22 apresenta o conjunto de telas principais do Echo 3D Blocks.

Figura 22 - Principais telas da aplicação desenvolvida.



Fonte: Elaborado pelo autor.

4.6 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os principais aspectos necessários para o desenvolvimento e que caracterizam a validade deste trabalho. Além disso, foi apresentado tanto o link do projeto: <https://echo-3d-blocks.web.app/#/>, quanto do link do projeto na plataforma *GitHub*: <https://github.com/Miguelm-del/echo-3d-blocks>

No próximo capítulo, será abordado a avaliação deste trabalho, a coleta e a análise dos dados obtidos a partir do estudo de caso.

5 ESTUDO DE CASO

Este capítulo aborda a avaliação do projeto, detalhando as atividades realizadas, a estratégia de coleta de dados e os resultados obtidos. O objetivo deste capítulo é descrever a análise do impacto do ensino de programação para pessoas com deficiência visual por meio do conjunto de ferramentas didáticas, incluindo os blocos 3D físicos e a aplicação desenvolvida.

A estratégia avaliativa adotada para validar o sistema de ensino de lógica de programação com blocos físicos 3D e o aplicativo desenvolvido baseia-se em um estudo de caso contendo duas tarefas a serem executadas pelos usuários. A abordagem é centrada na verificação da eficácia do sistema em facilitar o aprendizado e na avaliação da experiência de usuários portadores de algum tipo de deficiência visual. O estudo não visa controlar o comportamento dos participantes, buscando, assim, observar suas interações naturais com a solução proposta.

5.1 TAREFAS PROPOSTAS

Tarefa 1: Escaneamento e Processamento pelo Aplicativo

Os participantes escanearão os códigos QR dos blocos montados para processamento pelo aplicativo. Este teste verificará a precisão na interpretação dos blocos e na validação do algoritmo montado.

Justificativa: essa atividade busca avaliar a eficácia do aplicativo em processar as informações dos blocos físicos 3D e fornecer o retorno sobre a correta disposição e tipos dos blocos no algoritmo montado.

Tarefa 2: Montagem de Algoritmo com Blocos Físicos 3D

O usuário receberá um problema para resolver utilizando os blocos físicos 3D para montar o algoritmo correspondente. Essa atividade visa avaliar a capacidade dos blocos em representar conceitos lógicos de programação de forma tangível.

Justificativa: essa tarefa permite testar a eficiência e a usabilidade dos blocos físicos 3D como ferramenta de aprendizagem, oferecendo uma representação prática e acessível dos conceitos de programação.

5.2 EXECUÇÃO DA AVALIAÇÃO

O estudo sobre o sistema PWA para auxiliar no ensino de programação foi implementado entre os alunos dos cursos de tecnologia no Instituto Federal de Alagoas - IFAL, portadores de deficiência visual, ocorrido em 11/10/2023. No IFAL apenas um dos estudantes foi identificado com o perfil definido.

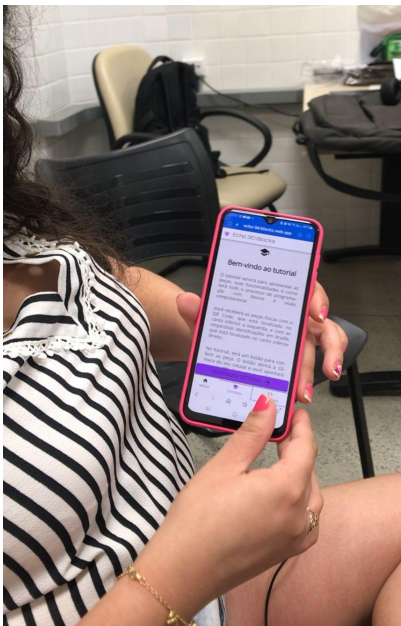
Para a execução do estudo, foram fornecidos aos alunos os recursos necessários, incluindo a documentação completa da aplicação disponível no repositório:

<https://github.com/Miguelm-del/echo-3d-blocks/wiki/Documenta%C3%A7%C3%A3o-Echo-3D-Blocks>. Além disso, para a coleta de dados, foi utilizado um questionário do Google Forms disponibilizado no endereço <https://forms.gle/n42g8skKSqyjJhN16>.

O usuário acessou a aplicação por meio de seu dispositivo móvel, com o leitor de tela ativado em seu aparelho. Foi sugerido que ele explorasse as principais páginas da aplicação, incluindo: a página inicial, o tutorial e a seção de exercícios. Posteriormente, foi pedido que realizasse as tarefas que foram propostas. Na ocasião, foram registradas fotografias para realizar a obtenção dos dados.

A Figura 23 ilustra o usuário que concordou em participar do teste da aplicação, sendo mostrado a tela de tutorial de um dos blocos disponíveis.

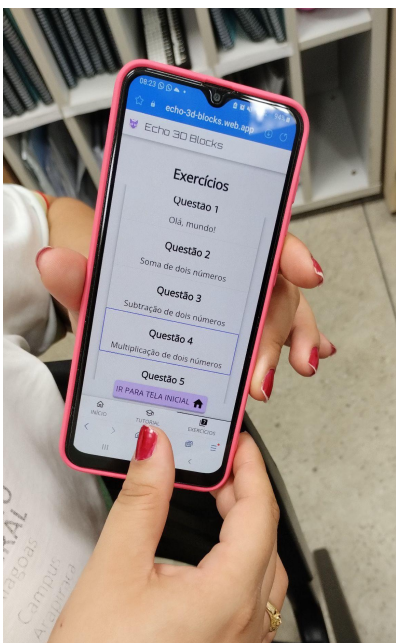
Figura 23 - Usuário testando a página de tutorial de um dos blocos.



Fonte: Elaborado pelo autor.

A Figura 24 exibe o usuário realizando testes na página de exercícios, que inclui uma lista de questões numeradas. Cada questão possui um título curto e específico. Na parte inferior da página, há um botão para retornar rapidamente à página inicial.

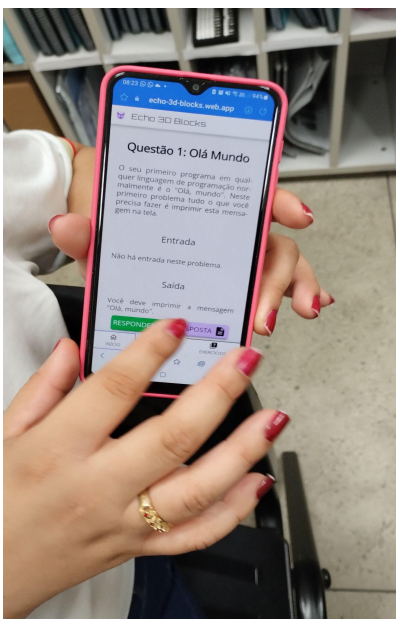
Figura 24 - Usuário testando a página de exercícios.



Fonte: Elaborado pelo autor.

A Figura 25 ilustra o usuário na página de uma das questões selecionadas por ele. Nessa página, é exibido o título da questão, uma descrição detalhada do problema e, em seguida, são apresentados exemplos de entrada e saída esperada do algoritmo. Esses exemplos auxiliam na compreensão do que é requisitado pela questão.

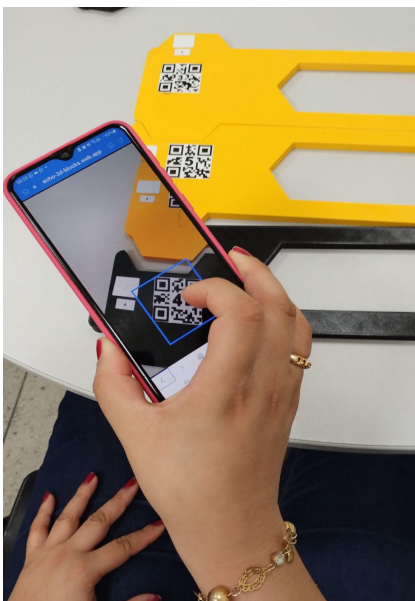
Figura 25 - Usuário na a página de uma questão de algoritmo.



Fonte: Elaborado pelo autor.

A Figura 26 ilustra um usuário que montou um algoritmo usando os blocos 3D físicos. O usuário está utilizando a câmera do dispositivo para escanear o QR code de um dos blocos que faz parte do algoritmo montado. Essa figura mostra a etapa de leitura do QR code para integrar o bloco ao algoritmo construído pelo usuário.

Figura 26 - Usuário realizando a leitura de QR code de um bloco.



Fonte: Elaborado pelo autor.

Ao final da execução das atividades, visando coletar a opinião dos usuários, bem como a efetividade da aplicação, foi solicitado que o mesmo respondesse um questionário. As perguntas aplicadas são apresentadas na Tabela 4.

O questionário é predominantemente qualitativo, pois busca opiniões, percepções e experiências dos usuários sobre o uso dos blocos físicos 3D e do aplicativo de aprendizado de lógica de programação.

Quadro 4 - Questões elaboradas para o levantamento do estudo de caso.

Questões
Email?
Nome?
Em qual faixa etária você se encaixa? (a) Menos de 18 anos (b) 18-25 anos (c) 26-35 anos (d) 36-45 anos (e) Mais de 45 anos
Qual é o seu gênero? (a) Masculino (b) Feminino (c) Não binário / Outro (d) Prefiro não informar

<p>Qual deficiência visual você possui? Por favor, selecione uma das opções abaixo:</p> <p>(a) Baixa visão (b) Cegueira total</p>
<p>Qual é o seu nível escolar?</p> <p>(a) Ensino fundamental (b) Ensino Médio (c) Ensino Médio Incompleto (d) Ensino Superior (e) Ensino Superior Incompleto</p>
<p>Você tem conhecimento prévio de lógica de programação?</p> <ul style="list-style-type: none">• Sim• Não
<p>Caso sua resposta anterior foi sim, qual nível de conhecimento você acredita ter? Assinale o grau considerando 1 como "Iniciante" e 5 como "Avançado".</p> <p>(a) 1 (b) 2 (c) 3 (d) 4 (e) 5</p>
<p>Você sentiu que os blocos físicos 3D representaram adequadamente os conceitos de lógica de programação?</p> <ul style="list-style-type: none">• Sim• Não
<p>O feedback fornecido pelo aplicativo foi claro em relação à precisão do algoritmo montado?</p> <ul style="list-style-type: none">• Sim• Não
<p>Os feedbacks sonoros e mensagens foram claros e fáceis de compreender em relação aos resultados obtidos na aplicação?</p> <ul style="list-style-type: none">• Sim• Não
<p>Como você avalia a facilidade de usar os blocos físicos 3D para montar o algoritmo? Assinale o grau considerando 1 como "muito difícil" e 5 como "muito fácil".</p> <p>(a) 1 (b) 2 (c) 3 (d) 4 (e) 5</p>
<p>Como foi sua experiência ao escanear os códigos QR dos blocos para validação pelo aplicativo? Assinale o grau considerando 1 como "Imprecisa" e 5 como "Excelente".</p> <p>(a) 1 (b) 2 (c) 3 (d) 4 (e) 5</p>

Qual foi a sua percepção geral sobre a integração dos blocos físicos 3D com o aplicativo para aprender programação (múltipla escolha)?

- (a) Intuitiva e fácil de entender
- (b) Funcional e prática
- (c) Relevante para o aprendizado
- (d) Interessante e envolvente
- (e) Precisa de melhorias
- (f) Outro (especifique)

Comente sobre qualquer desafio encontrado durante a montagem do algoritmo com os blocos físicos 3D.

Houve algum aspecto que você considerou difícil ou confuso durante o processo de escaneamento e validação pelo aplicativo?

Que melhorias você sugere para aprimorar a experiência de aprendizado com essa abordagem?

Fonte: Elaborado pelo autor.

5.4 ANÁLISE E DISCUSSÃO

Foi realizada uma análise dos resultados obtidos para identificar pontos fortes, oportunidades de melhoria e considerações importantes para aprimorar o sistema de aprendizado de programação com blocos físicos 3D e o aplicativo associado. A pessoa que participou das tarefas é uma mulher com idade entre 18 e 25 anos. Ela é portadora de deficiência visual, especificamente cegueira total, e está matriculada no ensino médio integrado de tecnologia no IFAL. Como só há uma pessoa no campus identificada com deficiência visual, não foi possível aplicar para mais pessoas. Para facilitar a discussão, definimos seções, apresentadas a seguir, referentes aos tópicos considerados mais relevantes.

5.4.1 Conhecimento prévio em lógica de programação

Para avaliar o nível de habilidade dos usuários que testaram a aplicação, esta variável foi utilizada para determinar o conhecimento prévio em lógica de programação foram aplicadas as seguintes perguntas: “*Você tem conhecimento prévio de lógica de programação?*” e “*Caso sua resposta anterior for sim, qual nível de conhecimento você acredita ter?*”. A primeira pergunta foi respondida com *sim* e o seu nível de conhecimento informado foi intermediário, com resposta 3 (resposta variando de 1 a 5), pois o mesmo está cursando o último ano do ensino médio integrado de tecnologia da informação. Essa base sólida já oferecia um conhecimento consistente em programação.

5.4.2 Uso dos blocos 3D físicos

Visando avaliar o impacto dos blocos 3D físicos no ensino de lógica de programação, a pesquisa concentrou-se na facilidade de uso e na representação fiel dos conceitos de programação. Foram aplicadas as seguintes perguntas: “*Como você avalia a facilidade de usar os blocos físicos 3D para montar o algoritmo?*” e “*Você sentiu que os blocos físicos 3D representaram adequadamente os conceitos lógicos de programação?*”.

O retorno do usuário foi positivo, indicando que os blocos representam adequadamente os conceitos de programação necessários e são fáceis e práticos

de usar, sendo uma ferramenta valiosa para aprimorar o aprendizado de pessoas com deficiência visual.

O usuário expressou que a representação tátil dos blocos de programação, juntamente com o *Braille* como identificador facilitou a assimilação dos conceitos, oferecendo uma forma concreta de interação que torna o processo de aprendizado mais intuitivo e acessível para pessoas com deficiência visual.

5.4.3 Uso da aplicação

Uma avaliação crucial deste estudo incide sobre o uso da aplicação, contemplando sua facilidade de uso e relevância. A usabilidade varia entre os usuários, pois cada um possui uma experiência diversa.

No entanto, podemos analisar qualitativamente a opinião do usuário. Onde o mesmo informou que fazia uso de leitores de tela para a utilização de computadores e dispositivos móveis. Foi relatado que a abordagem da aplicação oferece uma interface acessível e compreensível para usuários com leitores de tela, possibilitando a navegação e compreensão eficaz de cada componente do site.

Ademais, de acordo com a seguinte pergunta aplicada: “*Qual foi a sua percepção geral sobre a integração dos blocos físicos 3D com o aplicativo para aprender programação?*”, o usuário destacou que a integração dessas ferramentas didáticas é funcional e prática, sendo uma ótima opção para pessoas com deficiência visual que estão aprendendo programação. Tornando o aprendizado mais acessível e inclusivo, e pode ajudar a aumentar o engajamento dos estudantes.

5.4.4 Leitura de QR codes e feedbacks da aplicação

Um aspecto importante do projeto foi a incorporação de *QR codes* nos blocos 3D físicos. Assim, foi imprescindível avaliar se a leitura desses *QR codes* foi realizada com facilidade ou se os usuários encontraram alguma dificuldade durante esse processo. Além disso, verificou-se a eficácia dos feedbacks fornecidos para facilitar a compreensão do usuário sobre a seleção de um bloco correto ou inválido.

As perguntas “*Como foi sua experiência ao escanear os códigos QR dos blocos para validação pelo aplicativo?*” e “*Os feedbacks sonoros e mensagens foram claros e fáceis de compreender em relação aos resultados obtidos na aplicação?*” do questionário tiveram o objetivo de avaliar a opinião do usuário referente ao uso dos

QR codes e se os feedbacks sonoros e textuais auxiliaram no uso da aplicação e aprendizado. Segundo o resultado obtido das respostas do usuário, a leitura dos QR codes foi rápida e intuitiva, tendo uma experiência Excelente, com resposta 5 (resposta variando de 1 a 5). Adicionalmente, a implementação dos feedbacks sonoros e mensagens associadas a cada bloco ajudaram na compreensão do usuário, onde sua resposta foi positiva.

5.4.5 Dificuldades encontradas

Avaliar e tratar as dificuldades encontradas é um aspecto de extrema importância para o objetivo central deste projeto. A partir da seguinte pergunta foi analisado esse aspecto: “ *Houve algum aspecto que você considerou difícil ou confuso durante o processo de escaneamento e validação pelo aplicativo?*”.

De acordo com o usuário que realizou os testes e tarefas que foram propostas, a aplicação desenvolvida revelou-se de fácil utilização e intuitiva, especialmente para usuários de leitores de tela. Entretanto, durante o teste, foi identificado um problema relevante: os *QR Codes* não possuíam identificação em *Braille* ou relevo que os tornasse mais distintos para identificação tátil. Além disso, observou-se que o tamanho dos blocos 3D físicos poderia ser ligeiramente reduzido, o que facilitaria ainda mais sua manipulação e demandaria menos espaço para sua montagem. Essas melhorias são fundamentais para garantir uma experiência mais acessível e amigável para usuários com deficiência visual. Na próxima seção serão abordadas as sugestões de melhorias do projeto.

5.4.6 Sugestões de melhorias

Por fim, foi requisitado ao participante para oferecer sugestões a respeito do projeto após explorar a interação entre a aplicação e os blocos físicos 3D. Estas sugestões foram definidas na questão “*Que melhorias você sugere para aprimorar a experiência de aprendizado com essa abordagem?*”. Entre as sugestões propostas, destacaram-se alguns pontos de aprimoramento, tais como:

- No uso dos *QR codes*: inclusão de identificadores em *Braille* nos *QR codes* de cada bloco;

- Sobre os *feedbacks*: melhoria dos *feedbacks* sonoros, visando tornar os sons mais específicos e facilitar ainda mais a identificação;
- Outras funcionalidades: implementação de uma funcionalidade que acione a vibração do dispositivo móvel ao efetuar a leitura dos *QR codes*;
- Blocos físicos 3D: reduzir ligeiramente o seu tamanho e ter mais blocos para uso;
- Aperfeiçoamento dos recursos de aprendizagem: adicionar mais recursos educacionais na aplicação para abordar diferentes conceitos de programação, oferecendo explicações mais detalhadas ou exercícios interativos para promover o aprendizado.

Essas sugestões têm como objetivo enriquecer a experiência do usuário com deficiência visual, tornando a aplicação mais acessível, funcional e educativa.

5.5 CONSIDERAÇÕES FINAIS

Neste capítulo, foram detalhadas as metodologias utilizadas para a coleta de dados do estudo de caso, oferecendo uma análise e discussão aprofundadas dos resultados obtidos através da perspectiva do participante. E teve o objetivo de responder a pergunta de pesquisa “*é possível tornar o ensino de programação mais acessível e eficaz para pessoas cegas ou com deficiência visual, utilizando blocos 3D táteis?*”.

Os resultados analisados e discutidos neste capítulo demonstram avanços consideráveis na criação de ferramentas didáticas que oferecem suporte ao desenvolvimento do pensamento computacional em alunos com deficiência visual. Eles indicam a resposta afirmativa à questão de se é viável tornar o ensino de programação mais acessível e eficaz para pessoas cegas ou com outras formas de deficiência visual.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi enfrentado um desafio considerável ao compreender as realidades das pessoas com deficiência visual, sendo uma tarefa complexa para alguém sem experiência direta nesse universo.

O aprendizado da lógica de programação é desafiador para muitas pessoas, e a maioria das ferramentas existentes para essa finalidade não atende adequadamente às necessidades de pessoas com deficiência visual, carecendo de recursos e ferramentas acessíveis para esse público.

O objetivo principal foi alcançado, uma vez que o sistema proposto foi desenvolvido para ajudar a mitigar as limitações identificadas pelos usuários finais no aprendizado de lógica de programação, como evidenciado nos testes mencionados na seção de estudos de caso deste trabalho. A arquitetura de software criada neste projeto foi estruturada para garantir facilidade na manutenção e no versionamento do sistema. Essa abordagem permite atualizações e aprimoramentos contínuos, possibilitando ajustes e melhorias conforme necessário para atender às demandas específicas dos usuários com deficiência visual.

A ferramenta proposta tem o potencial de tornar o ensino de programação mais acessível e eficaz para os estudantes que possuem algum tipo de deficiência visual, contribuindo para sua independência e desenvolvimento. No entanto, há ainda alguns desafios que precisam ser superados para que a ferramenta seja cada vez mais eficaz. Um desses desafios é a necessidade de realizar testes mais abrangentes com pessoas com deficiência visual. Esses testes devem ser realizados com pessoas de diferentes níveis de experiência em programação e faixas etárias, para garantir que a ferramenta seja adequada para um público amplo.

O projeto apresenta um caminho de evolução e aprimoramento contínuos, destacando alguns pontos para trabalhos futuros, como a expansão do conjunto de blocos disponíveis, para promover a diversidade e amplitude dos conceitos de programação abordados, permitindo que a aplicação tenha uma gama mais ampla de exercícios aos usuários.

Considerando os avanços em visão computacional, uma direção futura envolveria reduzir a dependência dos *QR codes*, adotando algoritmos para identificar e analisar os blocos de acordo com forma, tamanho, cor, etc. Isso pode

incluir o uso, por exemplo, de técnicas de inteligência artificial. Essa remodelação poderia oferecer uma interação mais intuitiva e fluida na montagem de algoritmos.

A introdução de novas funcionalidades interativas, como um modo de aprendizado guiado que inclui um tutorial interativo para usuários iniciantes, juntamente com feedbacks personalizados adaptados às ações e erros ao montar algoritmos, tem como objetivo melhorar consideravelmente a experiência do usuário. Essas adições permitem uma interação mais dinâmica e enriquecedora com os recursos didáticos oferecidos pela aplicação.

REFERÊNCIAS

BRASIL. Lei N° 13.146, de 6 de julho de 2015. **Lei Brasileira de Inclusão das Pessoas com Deficiência (Estatuto da Pessoa com Deficiência)**. Diário Oficial da União 2015; 7 jul.

FIGMA, 2023. **Figma**. Disponível em: <https://www.figma.com/>. Acesso em: 10 dez. 2023.

FLANAGAN, David; NOVAK, Gregor M. 1998. **Java-Script: The Definitive Guide**. 1998.

GOMES, Anabela; HENRIQUES, Joana; MENDES, António José. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. **Educação, Formação e Tecnologias**, v. 1, n. 01, p. 93-103, 2008.

JUNQUEIRA, Ricardo. In Ciências ULisboa. **Filipa Rocha alcança o 2.º lugar do Prémio Jovens Inventores**. 2023. Disponível em: <https://ciencias.ulisboa.pt/pt/noticia/23-05-2023/filipa-rocha-nomeada-finalista-do-premio-jovens-inventores-2023>. Acesso em: 4 ago. 2023.

GALA, Ana Sofia. In hand talk. **WCAG 2.2: o que esperar da atualização de Diretrizes de Acessibilidade de Conteúdo da Web**. 2022. Disponível em: <https://www.handtalk.me/br/blog/wcag-2-2/>. Acesso em: 02 nov. 2023.

LE MOS, Edison Ribeiro; CERQUEIRA, Jonir Bechara. 2014. **O sistema Braille no Brasil**. Benjamin Constant, Rio de Janeiro, ano 20, edição especial, p. 23-28, nov. 2014.

LIMA, José Weliff Ferreira; MEIRELES, Alandson Mendonça Ribeiro. Metodologias de ensino de lógica de programação. **Revista Expressão Católica**, v. 4, n. 1, 2015.

MDN WEB DOCS. **HTML: Linguagem de Marcação de Hipertexto**, 2023. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 02 nov. 2023.

MDN WEB DOCS. **JavaScript**, 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 02 nov. 2023.

MORIJO, Daniel Kleber Santos; MARCELINO, Vitória de Oliveira; MANSANO, Naira da Silva. DALTONISMO E AS DIFERENTES PERCEPÇÕES DE CORES. **REGRAD - Revista Eletrônica de Graduação do UNIVEM - ISSN 1984-7866**, [S.l.], v. 10, n. 01, p. 433 - 439, nov. 2020. ISSN 1984-7866. Disponível em: <https://revista.univem.edu.br/REGRAD/article/view/3315>. Acesso em: 26 nov. 2023.

ORGANIZAÇÃO DAS NAÇÕES UNIDAS. **Objetivos de Desenvolvimento Sustentável (ODS)**, 2023. Disponível em: <https://brasil.un.org/pt-br>. Acesso em: 10 de set. de 2023.

PAN AMERICAN HEALTH ORGANIZATION, 2023. **Saúde ocular - OPAS/OMS**. Disponível em: <https://www.paho.org/pt/topicos/saude-ocular>. Acesso em: 8 de nov de 2023.

PEREIRA, Fernanda Veronica Fleck. **Lógica de programação: uma abordagem por meio de um jogo adaptado para alunos com deficiência visual**. 2023. Dissertação (Mestrado em Ensino de Ciências e Educação Matemática) - Universidade Estadual de Ponta Grossa, Ponta Grossa, 2023.

PORVIR. **Code Jumper leva programação a crianças com deficiência visual**. 2019. Disponível em: <https://porvir.org/code-jumper-leva-programacao-a-criancas-com-deficiencia-visual/>. Acesso em: 27 set. 2023.

QUASAR FRAMEWORK, 2023. **Quasar Framework**. Disponível em: <https://quasar.dev/>. Acesso em: 10 dez. 2023.

ROBE, Rafaela; SALTON, Bruna Poletto; BERTAGNOLLI, Silvia. **Recursos Pedagógicos para o Ensino de Programação de Estudantes com Deficiência Visual: uma revisão sistemática da literatura**. RENOTE, v. 18, n. 1, 2020.

SCAICO, Pasqueline Dantas et al. **Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch**. 2013. Revista Brasileira de Informática na Educação, v. 21, n. 02, p. 92.

SCRATCH, 2023. **Scratch**. Disponível em: <https://scratch.mit.edu>. Acesso em: 24 de set. 2023.

SOUZA, Draylson Micael; DA SILVA BATISTA, Marisa Helena; BARBOSA, Ellen Francine. **Problemas e dificuldades no ensino de programação: Um mapeamento sistemático**. Revista Brasileira de Informática na Educação, v. 24, n. 1, p. 39, 2016.

TECHOPEDIA. **What is Programming Logic?** Disponível em: <https://www.techopedia.com>. Acesso em: 30 abr. 2020.

VIEIRA, Guilherme; OLIVEIRA, Emilly Christiny; OLIVEIRA, Emilly Priscila; SANTOS, Jousiclecia; SILVA, Miguel; ROLIM, Cledja. 2023. **Uma ferramenta para auxílio do aprendizado de lógica de programação para alunos com deficiência visual**. In: Conferência Internacional sobre Informática na Educação (TISE), 2023. Disponível em: http://www.tise.cl/2023/doc/ShortPapers/TISE_2023_paper_42.pdf. Acesso em: 10 dez. 2023.

VUE.JS, 2023. **Vue.js**. Disponível em: <https://vuejs.org/>. Acesso em: 10 dez. 2023.

WAI-ARIA, 2023. **Site oficial da WAI-ARIA**. Disponível em: <https://www.w3.org/TR/wai-aria-1.2/>. Acesso em: 02 nov. 2023.

WORLD WIDE WEB CONSORTIUM (W3C). **The World Wide Web Consortium (W3C) develops standards and guidelines to help everyone build a web based on the principles of accessibility.** Disponível em: <https://www.w3.org>. Acesso em: 01 nov. 2023.

WCAG. 1999. **Web Content Accessibility Guidelines (WCAG) 1.0.** Disponível em <http://www.w3c.org/TR/WCAG10/>. Acesso em: 01 nov. 2023.

WCAG. 2008. **Web Content Accessibility Guidelines (WCAG) 2.0.** Disponível em <http://www.w3.org/TR/WCAG20/>. Acesso em: 01 nov. 2023.

WCAG. 2023. **Web Content Accessibility Guidelines (WCAG) 2.1.** Disponível em: <https://www.w3.org/TR/WCAG21/>. Acesso em: 01 nov. 2023.

WORLD HEALTH ORGANIZATION et al. 2011. **World report on disability 2011.** World Health Organization (WHO). Disponível em: <https://www.who.int/teams/noncommunicable-diseases/sensory-functions-disability-and-rehabilitation/world-report-on-disability>. Acesso em: 25 de nov. 2023.

WORLD HEALTH ORGANIZATION et al. 2019. **World report on vision.** Disponível em: <https://www.who.int/publications-detail-redirect/9789241516570>. Acesso em: 4 de nov. de 2023.

APÊNDICE A - TERMO DE CONSENTIMENTO

Você está sendo convidado(a) a participar da pesquisa “TRABALHANDO O PENSAMENTO COMPUTACIONAL EM ALUNOS COM DEFICIÊNCIA VISUAL”. A proposta visa oferecer uma ferramenta para auxiliar o aprendizado de lógica de programação. Com este projeto, esperamos facilitar a compreensão dos conceitos por pessoas com deficiência visual, proporcionando uma maneira mais acessível e eficaz de aprender. Para decidir sobre o seu consentimento, é importante que você conheça as seguintes informações sobre a pesquisa:

- a. Você foi escolhido para participar, porém, sua colaboração é totalmente opcional.
- b. Este estudo tem como propósito validar a eficácia do uso da aplicação.
- c. Sua participação nesta pesquisa é voluntária e envolverá o uso da aplicação e o preenchimento de um questionário.
- d. Você pode interromper a entrevista a qualquer momento, de acordo com sua conveniência e desejo.
- e. Caso deseje uma cópia deste termo ou necessite interromper a entrevista, por favor, comunique ao pesquisador Miguel Márcio Magalhães Silva pelo o email mmms2@aluno.ifal.edu.br de acordo com sua disponibilidade e desejo.

Entendemos que sua participação neste estudo pode levantar preocupações sobre questões éticas e morais relacionadas à privacidade. Entretanto, é importante ressaltar que a utilização da aplicação e sua participação no estudo não apresentam riscos significativos aos participantes.

As informações obtidas durante a pesquisa serão tratadas com total confidencialidade. Garantimos a segurança e o sigilo sobre sua participação, assegurando que os dados coletados não serão divulgados de maneira a permitir sua identificação.

De posse dessas informações, gostaríamos que você se pronunciasse acerca da entrevista:

- Dou meu consentimento para a sua realização.
 Não consinto com a sua realização.

Assinatura do entrevistador
Miguel Márcio Magalhães Silva
mmms2@aluno.ifal.edu.br

Assinatura do entrevistado