



ORIENTADOR:
Prof. Dr. Sylker Teles da Silva

HELLO (PAPER) WORLD: DESENVOLVIMENTO
DE SOLUÇÕES DESPLUGADAS PARA O ENSINO
COMPUTACIONAL NO CONTEXTO SOCIAL
AMAZÔNICO

Gabriel de Oliveira Araújo

DISSERTAÇÃO
DE MESTRADO
2024

Gabriel de Oliveira Araújo

HELLO (PAPER) WORLD: DESENVOLVIMENTO DE SOLUÇÕES
DESPLUGADAS PARA O ENSINO COMPUTACIONAL NO CONTEXTO
SOCIAL AMAZÔNICO

Dissertação submetida ao Programa de Pós-Graduação em
Design da Universidade Federal do Amazonas como requisito
para a obtenção do Grau de Mestre em Design.

Orientador: Prof. Dr. Sylker Teles da Silva

Manaus
2024

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

A663h Araújo, Gabriel de Oliveira
Hello (paper) world : desenvolvimento de soluções desplugadas
para o ensino computacional no contexto social amazônico /
Gabriel de Oliveira Araújo . 2024
175 f.: il. color; 31 cm.

Orientador: Sylker Teles da Silva
Dissertação (Mestrado em Design) - Universidade Federal do
Amazonas.

1. Algoritmos. 2. Design de Experiência do Usuário. 3. Design
Thinking. 4. User Experience Design. I. Silva, Sylker Teles da. II.
Universidade Federal do Amazonas III. Título

Gabriel de Oliveira Araújo

HELLO (PAPER) WORLD: DESENVOLVIMENTO DE SOLUÇÕES
DESPLUGADAS PARA O ENSINO COMPUTACIONAL NO CONTEXTO
SOCIAL AMAZONICO

Esta Dissertação foi julgada adequada para obtenção do
Título de Mestre e aprovada em sua forma final pelo
Programa de Pós-Graduação em Design.

Manaus, 28 de fevereiro de 2024

BANCA EXAMINADORA

Prof. Dr. Sylker Teles da Silva, Presidente
Universidade Federal do Amazonas

Prof. Dr. Wilson da Silva Prata, Membro
Universidade Federal do Amazonas

Prof. Dr. Andre Wilson Archer Pinto Salgado, Membro Externo
Instituto Federal do Amazonas



Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Design

FOLHA DE APROVAÇÃO

GABRIEL DE OLIVEIRA ARAÚJO

HELLO (PAPER) WORLD: DESENVOLVIMENTO DE SOLUÇÕES
DESPLUGADAS PARA O ENSINO COMPUTACIONAL NO CONTEXTO SOCIAL AMAZÔNICO

Dissertação apresentada ao Programa de Pós-Graduação em Design da Universidade Federal do Amazonas, como parte do requisito para a obtenção do título de Mestre em Design, área de concentração Design, Inovação e Desenvolvimento Tecnológico.

Aprovada em: Manaus, 28 de fevereiro de 2024

BANCA EXAMINADORA

Prof. Dr. Sylker Teles da Silva, Presidente

Universidade Federal do Amazonas

Prof. Dr. Wilson da Silva Prata, Membro Interno

Universidade Federal do Amazonas

Prof. Dr. André Wilson Archer Pinto Salgado, Membro Externo

Universidade Federal do Amazonas



Documento assinado eletronicamente por **Sylker Teles da Silva, Usuário Externo**, em 04/04/2024, às 14:56, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Wilson Silva Prata, Usuário Externo**, em 04/04/2024, às 15:51, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Andre Wilson Archer Pinto Salgado, Usuário Externo**, em 07/04/2024, às 13:38, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1991970** e o código CRC **D572830C**.

Av. Octávio Hamilton Botelho Mourão - Bairro Coroado 1 Campus Universitário Senador Arthur Virgílio Filho, Setor Norte - Telefone: (92) (92) 3305-1181 / Ramal 2600
CEP 69080-900 Manaus/AM - ppgd@ufam.edu.br

Referência: Processo nº 23105.005636/2024-51

SEI nº 1991970

Grandes designers produzem experiências agradáveis. –

Donald A. Norman, livro The Design of Everyday Things

Agradecimentos

Ao Deus Pai, infinito Dominador! E ao teu venerável, vero e único Filho, e ao Santo Espírito, Consolador. Sejam o louvor, a honra, a glória e o poder, para todo o sempre!

A minha esposa Beatriz e minha filha Chloe, a quem eu amo como a mim mesmo, por terem suportado as adversidades dessa parte de minha formação. Entre lutas e vitórias vocês sempre estiveram comigo e não me deixaram esmorecer.

Aos meus pais, Laédio e Suely, e meus irmãos Daniel e Samuel, que sempre me apoiaram. Por terem me recebido juntamente com minha família, em suas casas nos meses finais dessa formação e me deram muito auxílio. Mãe, obrigado por todos aqueles cafezinhos. Pai obrigado por ter me alimentado, a mim e minha família, quando eu não tinha.

A minha sogra, Maria Antônia, que recebeu e deu suporte a minha esposa e filha, quando não podia lhes dar atenção enquanto construía esse trabalho.

A Mara e ao meu sogro Rene, por terem me ajudado financeiramente várias vezes quando minha limitação emocional e física me induziu a sobreviver exclusivamente da bolsa da FAPEAM.

Ao meu orientador, Prof. Dr. Sylker Teles da Silva, pela orientação e condução nesta jornada de formação acadêmica mais que especial para mim.

Ao meu co-orientador, Prof. Dr. Wilson Prata, por ter me conduzido nesse processo de construção do saber, sempre me levando a acreditar em mim mesmo. Por ter sido humano em todos os momentos. Por ter me tratado como humano, respeitado minhas fraquezas e potencializado minhas habilidades.

Ao corpo docente, discente e administrativos da Manaós Tech, representados na pessoa de Glauco Aguiar como diretor, que me deram a oportunidade de aumentar meu know-how no ensino da Ciência da Computação para crianças.

A Eduardo Jorge, Ana Carolina e Maria Sophia, amigos pessoais que muito me ajudaram nos tempos disponíveis, por meio das conversas descontraídas. Me instruíram a olhar por meio de outras perspectivas a condução de minha pesquisa.

Aos alunos do Colégio Connexus e Creche Bebê Bombom, por terem se disposto a participar com todas as intervenções propostas. Esta pesquisa é por vocês e para vocês.

Aos participantes da banca de exame, Prof. Dra. Franciane da Silva Falcão, Prof. Dra. Andreza Bastos Mourão, Prof. Dr. Augusto César Barreto Rocha e Prof. Dr. Marcelo Augusto Zacarias, por terem dedicado parte de vosso tempo para apreciação de minha dissertação e arguição construtiva.

Aos meus alunos da Fundação Matias Machline, que mesmo sem saber, em nossas conversas sobre como o Design de Experiência de Usuário está em todas as coisas, me permitiram perceber que esta formação é sobre como melhorar a vida de pessoas.

E, por fim, a Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) por ter financiado esta pesquisa.

Resumo

A partir da atuação do pesquisador como professor de Robótica e Programação em uma escola de ensino fundamental, foi possível notar dificuldades de aprendizagem de alunos das séries iniciais da Educação Básica quanto aos conceitos da Ciência da Computação, quando realizavam atividades para desenvolver o Pensamento Computacional. O Pensamento Computacional é um processo racional para resolução de problemas. As dimensões do Pensamento Computacional proposto por Guarda e Pinto (2020), possuem 3 etapas das quais a segunda tem seu enfoque na Resolução do Problema, por meio de Algoritmos. Ao realizar atividades de construção de sequências lógicas no Laboratório de Inovações do Colégio Connexus, os alunos mostraram inquietações e insatisfação quanto às ferramentas empregadas no ensino dos conceitos de Algoritmos e Programação. Foram identificados inicialmente dois motivos para tal insatisfação: os alunos não viam progressão nas atividades por causa dos elementos repetitivos da interface e não reconheciam as representações gráficas dos softwares/aplicativos como objetos lúdicos de seu contexto social e cultural. Isso está relacionado ao fato de que o ensino de crianças se dá por meio de objetos lúdicos, e estes precisam ser reconhecidos com tal. Para possibilitar a formulação de novas hipóteses a respeito dessas questões, algumas entrevistas e ensaios preliminares foram realizados. As três atividades de imersão, em

formato desplugado, possibilitaram *feedbacks* pertinentes a respeito de elementos motivadores e índices de engajamentos dos alunos. Os resultados apontaram a robótica e os jogos como elementos motivadores no processo de aprendizagem dos conceitos de Algoritmos e Programação deste público-alvo. Mediante isto, sabendo que o Pensamento Computacional é a habilidade do século XXI e que a habilidade de pensar algorítmicamente para resolver problemas do dia a dia precisa ser desenvolvida desde a infância, o objetivo desta pesquisa é desenvolver uma solução centrada no usuário para ensino e aprendizagem de Algoritmos e Programação no Ensino Fundamental 1: Séries Iniciais, para apoiar atividades Desplugada (sem o uso do computador). O processo de Design para desenvolvimento da solução centrado nas crianças foi norteado por técnicas e ferramentas do Design de Experiência do Usuário – UX Design, uma vez que compreende as percepções que uma pessoa tem à medida que interage com um produto ou serviço. O *Design Thinking* foi empregado como metodologia projetual para concepção da solução. Com resultado, obteve-se um jogo de tabuleiro e cartas, que utiliza elementos lúdicos do contexto sociocultural amazônico, para ensinar conceitos relacionados a Algoritmos e Programação, tais como Sequência Lógica, Estrutura de Controle e Estrutura de Repetição.

Palavras-chave: *Algoritmos, Design de Experiência do Usuário, Design Thinking*

Abstract

From the researcher's performance as a Robotics and Programming teacher in a elementary school, it was possible to notice difficulties in the learning of students of the initial series of Basic Education regarding the concepts of Computer Science, when they carried out activities to develop Computational Thinking. Computational Thinking is a rational process for solving problems. The dimensions of Computational Thinking proposed by Guarda and Pinto (2020) have 3 stages, the second of which focuses on Problem Solving, through Algorithms. When carrying out activities of construction of logical sequences in the Laboratory of Innovations of Colégio Connexus, the students showed concerns and dissatisfaction with the tools used to teach the concepts of Algorithms and Programming. Initially, two reasons for this dissatisfaction were identified: the students did not see progress in the activities because of the repetitive elements of the interface and they did not recognize the graphic representations of the software/applications as playful objects in their social and cultural context. This is related to the fact that children are taught through playful objects, and playful objects need to be recognized as such. To enable the formulation of new hypotheses regarding these questions, some interviews and preliminary tests were carried out. The three immersion activities, in unplugged format, provided relevant feedback on motivating elements and student engagement rates. The results showed robotics and games as motivating elements in the learning process of Algorithms and Programming concepts for this target audience. Therefore, knowing that

Computational Thinking is the skill of the 21st century, that the ability to think algorithmically to solve everyday problems needs to be developed from childhood, the objective of this research is to develop a user-centered solution for teaching and learning. learning Algorithms and Programming in basic education, to support Unplugged activities (without using the computer). The Design process for the development of the solution centered on children was guided by the techniques and tools of User Experience Design – UX Design. User experience comprises the perceptions a person has as they interact with a product or service. Design Thinking was used as a design methodology for designing the solution. As a result, a board and card game was obtained, which uses playful elements from the Amazonian sociocultural context, to teach concepts related to Algorithms and Programming, such as Logical Sequence, Control Structure and Repetition Structure.

***Key word:** Algorithms, User Experience Design, Design Thinking*

Lista de Figuras

Figura 1. Esquema de conversão de dados em informações.	35
Figura 2. Novo esquema de relacionamento entre as áreas e seus contextos (adaptado de BRACKMANN (2017)).	44
Figura 3. Os Quatro Pilares do Pensamento Computacional.	45
Figura 4. Partes de uma bicicleta (Decomposição)	46
Figura 5. Similaridade entre raças de cachorros (Reconhecimento de Padrões)	47
Figura 6. Padrão e replicações	48
Figura 7. O mapa do metrô como exemplo de abstração do mundo real.	49
Figura 8. Fluxograma de um algoritmo para descobrir a idade de uma pessoa.	50
Figura 9. Novo modelo para o PC	54
Figura 10. Marcos da trajetória da Informática na Educação Brasileira.	56
Figura 11. Organização da interface do Scratch.	63
Figura 12. Interface do VisuAlg.	64
Figura 13. Comparação da Hierarquia das Necessidades Humanas e a Nova ergonomia.	68
Figura 14. Estratégias empregadas e conhecimentos adquiridos na fase de Imersão.	71
Figura 15. Alunos participando do Módulo de Mecânica de Robôs na Manaós Tech.	80
Figura 16. Interfaces das diferentes aplicações para o ensino de programação com o Robô Dash.	81
Figura 17. Alunos do 1º ano realizando a atividade na Plataforma Code.org - Fase 5, Atividade Labirinto Clássico.	82
Figura 18. Plataforma Minecraft Education.	86
Figura 19. ScratchJr Cards.	87
Figura 20. Plataforma Code.org.	88
Figura 21. Plataforma Khan Academy.	88
Figura 22. Plataforma Tynker.	89
Figura 23. Plataforma Swift Playgrounds.	90
Figura 24. Plataforma Blockly Games.	91
Figura 25. Plataforma Scratch.	92

Figura 26. Plataforma Roblox.	93
Figura 27. LEGO Mindstorms.	93
Figura 28. Cartão de Insights.	97
Figura 29. Cartão de Insight 2 preenchido.	98
Figura 30 - 1. Recursos Didáticos (Parte 1 - Diagrama de Afinidade).	99
Figura 31. Comportamentos dos Alunos (Parte 2 - Diagrama de Afinidade).	100
Figura 32. Metodologias de Ensino (Parte 3 - Diagrama de Afinidade).	101
Figura 33 - Persona Professora.	103
Figura 34. Persona Aluna.	103
Figura 35. Atividade Imersiva.	105
Figura 36. Material usado na confecção do Tabuleiro.	107
Figura 37. Execução do Workshop de Cocriação 1	108
Figura 38. Execução do Workshop de Cocriação 2	109
Figura 39. Ícone do Software Adobe Illustrator 2021	116
Figura 40. Tabuleiro em forma de matriz	120
Figura 41. Instruções e Sequência Lógica	120
Figura 42. Modelos de Sequências Lógica.	121
Figura 43. Tamanho do Tabuleiro - 1º Esboço.	122
Figura 44. Coluna adicionada – AF.	123
Figura 45. Rotas principais e secundárias.	123
Figura 46. Rotas do Tabuleiro.	124
Figura 47. Visão do tabuleiro a partir das perspectivas dos 4 jogadores.	125
Figura 48. Novas guias de linhas e colunas.	126
Figura 49. Área para design do tabuleiro.	127
Figura 50. Primeiros esboços do design do tabuleiro.	127
Figura 51. Arquitetura das informações das cartas de ação.	128
Figura 52. Exemplo de um deck de cartas.	129
Figura 53. Cartas de Repetição	129
Figura 54. Carta Especial: Saltar	130
Figura 55. Sequência lógica sem uso da carta SALTAR	130
Figura 56. Sequência lógica com a carta SALTAR	131
Figura 57. Variações de jogadas com a carta SALTAR	131
Figura 58. Casa do Tabuleiro que permite salto.	132
Figura 59. Uso da Carta de Controle em uma Sequência Lógica.	133
	10

Figura 60. Carta de Obstáculo.	133
Figura 61. Pino de Bloquei indicado no tabuleiro	134
Figura 62. Protótipo de Pino de Bloqueio.	134
Figura 63. Carta Remover Obstáculo.	135
Figura 64. Carta Teste de Mesa	135
Figura 65. Protótipo de Pino. Personagem Onça Pintada	137
Figura 66. Protótipo de Carta de Personagem.	137
Figura 67. Protótipo de Pino e Carta Personalizável.	138
Figura 68. Logo do jogo Codezônia.	140
Figura 69. Tabuleiro Codezônia	142
Figura 70. Personagens do jogo Codezônia.	144
Figura 71. Pino da personagem Onça Pintada	144
Figura 72. Cartas de Ação.	145
Figura 73. Cartas de Repetição.	146
Figura 74. Carta de Controle.	146
Figura 75. Carta de Salto.	147
Figura 76. Carta de Obstáculo.	148
Figura 77. Carta Remover Obstáculo.	148
Figura 78. Carta Teste de Mesa.	149
Figura 79. Cartas de Personagens.	149
Figura 80. Carta de Personagem Personalizável.	150
Figura 81. Exemplo de jogadas com 5 cartas.	151
Figura 82. Exemplo de uso de Carta de Repetição.	152
Figura 83. Pino posicionado na rota de um jogador.	153
Figura 84. Exemplo de uso da Carta de Controle e Cartas de Salto.	154

Lista de Quadros

Quadro 1. Classes de Amostras de Abstrações Computacionais.	40
Quadro 2. Resultado do Benchmarking.	95
Quadro 3. Quadro de Ideias.	114
Quadro 4. Matriz de Posicionamento.	115
Quadro 5. Habilidades para aprendizagem.	117
Quadro 6. Objetivos de Aprendizagem.	118
Quadro 7. Atividades chaves.	118

Sumário

1	INTRODUÇÃO	16
	1.1 Questão de Pesquisa	19
	1.2 Delimitação do Problema	20
	1.3 Objetivos	22
	1.3.1 Geral	22
	1.3.2 Específicos	22
	1.4 Justificativa	23
	1.5 Delimitação do Estudo	26
	1.6 Estrutura do Documento	26
2	REFERENCIAL TEÓRICO	28
	2.1 A natureza da Computação	29
	2.2 Pensamento Computacional	36
	2.2.1 Dimensões do Pensamento Computacional	44
	2.2.1.1 Dimensões do Pensamento Computacional conforme Brackmann (2017)	44
	2.2.1.2 Dimensões do Pensamento Computacional conforme Guarda e Pinto (2020)	50
	2.2.2 O Pensamento Computacional na Educação brasileira.....	55
	2.2.2.1 Ensino e Aprendizagem de Algoritmos e Programação no Ensino Fundamental 1: Séries Iniciais	60
	2.2.3 Recursos didáticos para o ensino e aprendizagem de Algoritmo e Programação no Ensino Fundamental 1: Séries Iniciais	61
	2.2.3.1 Metodologia Desplugada	64
	2.3 Design Centrado no Usuário	65

2.3.1 Origem e Definição	65
2.3.2 User Experience Design	67
2.3.3 UX para crianças	69
2.4 Metodologia para desenvolvimento de soluções centradas em crianças	70
2.4.1 Design Thinking	70
2.4.2 Como aplicar Design Thinking com crianças	72
3 MATERIAIS E MÉTODOS	74
3.1 Métodos, técnicas e procedimentos	75
3.1.1 Imersão	75
3.1.1.1 Pesquisas Bibliográficas	76
3.1.1.2 Observação Participativa	79
3.1.1.3 Observação Não Participativa	81
3.1.1.4 Entrevistas	84
3.1.1.5 Pesquisa Desk	85
3.1.1.6 Benchmarking	93
3.1.2 Análise e síntese	96
3.1.2.1 Cartões de Insights	96
3.1.2.2 Diagrama de Afinidade	98
3.1.2.3 Personas	101
3.1.3 Ideação	104
3.1.3.1 Atividade e Brainstorming	105
3.1.3.2 Workshops de Cocriação	106
Workshop de Cocriação 1	106
Workshop de Cocriação 2	106
3.1.3.3 Matriz de Posicionamento	110
3.1.4 Prototipação	115
3.1.4.1 Definição dos objetivos de aprendizagem	116

3.1.4.2 Definição e Design dos Elementos	
Lúdicos	118
4 RESULTADOS	139
4.1 Codezônia – Resumo do jogo	140
4.2 O Tabuleiro Codezônia	141
4.3 Os Personagens	143
4.4 As Cartas	144
4.5 Jogabilidade	150
4.5.1 Regras de Jogo	151
4.6 Pensamento Computacional e os	
conceitos de Programação no Codezônia	154
4.7 Cenários de Uso	155
4.7.1 Salas de aula	155
4.7.2 Clubes de Programação	156
4.7.3 Competições	156
4.7.4 Eventos Educacionais	156
4.8 Validação da Solução e refinamento	157
5 CONSIDERAÇÕES FINAIS	158
Sugestões para trabalhos futuros	159
REFERENCIAS BIBLIOGRÁFICAS	161
APÊNDICE A	167
APÊNDICE B	170
APÊNDICE C	172
APÊNDICE D	175

CAPÍTULO 1

Introdução

As tecnologias computacionais estão presentes no dia a dia das escolas, em maior ou menor grau. Neste contexto, as crianças são apresentadas a diversos softwares, aplicativos e plataformas online que assumem o papel de auxiliar no processo de ensino aprendizagem de conceitos da Computação, tais como a programação. Isso pode ocorrer de modo orgânico, quando os próprios alunos recorrem aos conteúdos disponíveis online, ou utilizam de recursos digitais como jogos e plataformas de ensino e aprendizagem. Porém, também pode ser estruturado e sistematizado pela instituição de ensino, onde a mesma insere problemas lógicos e recursos digitais no seu conteúdo programático.

A partir desta segunda premissa, as escolas têm investido em recursos digitais por conta de serem gratuitos, reutilizáveis, online e possuírem diferentes níveis a ser trabalhados. Por outro lado, os softwares e aplicativos não são os únicos recursos didáticos para se aprender Algoritmos e Programação, já que de forma genérica, estes conceitos estão em todos os lugares e podem ser vistos em diferentes contextos.

O processo de ensino e aprendizagem destes conceitos traz consigo uma série de oportunidades e desafios. Diante de tantas possibilidades de ferramentas com fins didáticos a serem empregadas no auxílio da aprendizagem de Algoritmos e

Programação, acaba ficando a cargo do professor variar entre as opções, oferecendo diferentes plataformas de resolução de

problemas por meio de algoritmos. Assim, os softwares e aplicativos seguem um certo padrão, no qual ofertam um problema, oferecem ferramentas que possibilitam ao aluno a construção de uma solução passo a passo.

Por outro lado, fica a cargo do aluno, construir a melhor solução baseado em seus conhecimentos, noção de lógica, modelos estruturais e demais habilidades oriundas da Ciência da Computação, habilidades estas denominadas de Pensamento Computacional - PC, do inglês *Computational Thinking*.

Porém, os softwares, aplicativos e quaisquer outras ferramentas de auxílio pedagógico precisam ser construídas centradas no usuário, isto é, no aluno, a fim de que os elementos de interface, os objetos lúdicos, as interações, os objetivos de aprendizagem, as avaliações de aprendizagem e elementos de gamificação empregados favoreçam o objetivo final que é aprendizagem dos conceitos propostos. Desta mesma forma deve-se preocupar o desenvolvedor/criador de ferramentas e objetos de aprendizagem que se propõem a auxiliar a aprendizagem de conceitos de Algoritmos.

Para esse desenvolvimento devemos ressaltar que o algoritmo é um dos pilares do Pensamento Computacional. Como processo, o PC descreve as atividades cognitivas adotadas na resolução de problemas que podem ser solucionados com apoio da computação (Wing, 2010). Para a autora, este conceito dá base para que um indivíduo consiga particionar um problema grande e complexo, analisar as partes à luz de outros problemas mais fáceis e que já foram solucionados, dar foco a detalhes mais importantes e determinar passos ordenados para resolver cada um dos problemas menores. Os passos ordenados para resolução de problemas são os próprios algoritmos.

Partindo desta contextualização, adentramos no universo das escolas que possuem a robótica e a programação em seu currículo, mais precisamente a Manaós Tech. A escola conta com um corpo docente constituído por, aproximadamente, 12 professores, entre engenheiros de softwares, programadores, desenvolvedores jogos, designers, pedagogos com pelo menos uma pós-graduação a nível de especialização e/ou mestrado. Atualmente, a escola oferece 7 cursos em sua grade de cursos, planejados para atender públicos que vão de 5 anos a 15 anos em níveis básicos, intermediários e avançados de programação. Possui em seu corpo discente cerca de 300 alunos, que por limitação de estrutura, são atendidos além da sede, nas escolas parceiras.

A Manaós Tech trabalha o ensino de conceitos de programação e robótica juntamente com as escolas parceiras. Estas são escolas regulares, registradas pelo MEC e possuem uma matriz curricular comum baseados nas Leis de Diretrizes e Bases da Educação (LDB) e na Base Nacional Comum Curricular (BNCC). Oferecem pelo menos 1 dos 3 níveis de escolaridade: Educação Básica, Ensino Fundamental e Ensino Médio, excluindo o Ensino Superior. A relação de parceria da Manaós Tech com as escolas parceiras se dá na alocação de um docente da Sede dentro da escola parceira, atuando em conformidade com o currículo da escola e em consonância com as demais disciplinas deste currículo.

Dentre as escolas parceiras encontra-se o Colégio Connexus, no qual eu sou alocado. Está situado na cidade de Manaus, possui um laboratório chamado Laboratório de Inovações, com 14 máquinas desktop, projetores, internet cabeada, lousas, LEGOs e diferentes ferramentas lúdicas que estão à disposição para atividades dos professores. O nome do laboratório também nomeia a disciplina do currículo que reserva 1 tempo de 50 min por semana para que os alunos possam ir ao laboratório realizar atividades relacionadas à Educação Digital, Programação e Robótica. As atividades são conduzidas pelo professor da Manaós Tech auxiliado por uma professora efetiva da escola. Desta forma, a atuação dos professores da Manaós Tech nos 7 cursos ministrados na sede se difere da atuação dentro das escolas parceiras, contudo, a aprendizagem de conceitos de programação e robótica estão intrínsecas em todas as atividades realizadas no Laboratório de Inovação.

O colégio Connexus, em parceria com a Manaós Tech, busca oferecer um conhecimento de vanguarda para seus alunos. Contudo, dado a amplitude dos conceitos de Programação e a variedade de recursos usados no auxílio da aprendizagem de Algoritmos por meio de computador, é possível observar uma pulverização das escolhas por parte dos alunos em relação aos recursos que querem usar. Isso se dá ao fato de que cada aluno percebe e associa os elementos de Design e de Interação desses recursos ao seu contexto particular, o que dificulta o uso de uma ferramenta comum na mediação dos conceitos Algoritmos e Programação.

As situações descritas elucidam as motivações predominantes para a presente pesquisa. A formação deste pesquisador em Licenciatura em Computação combinada ao Design, área de conhecimento em que essa pesquisa se insere, reforça a importância dos conhecimentos de Design Centrado no Usuário. Este pode ser considerado um processo iterativo de desenvolvimento de design centrado no usuário, no qual estabelece uma relação proximal e integrada entre o processo de desenvolvimento de um produto para o ensino aprendizagem e os alunos, mantendo-os no centro de todas as decisões a serem tomadas durante todo o processo.

Outro conceito a ser abordado nesta pesquisa é o Design de Experiência do Usuário, também chamado de UX Design. A Experiência do Usuário são os efeitos sentidos pelo usuário, neste caso as crianças, como resultado da interação com softwares/aplicativos que são usados no ensino-aprendizado de algoritmo. Estes efeitos podem causar impactos emocionais na usabilidade, na utilidade, durante ou após a interação (Hartson e Pyla, 2012).

Portanto, a presente pesquisa concentra-se na área de Design, Inovação e Desenvolvimento Tecnológico, compreendendo a seguinte linha de pesquisa: Design, Sistemas de Produtos e Processos, pretende desenvolver um recurso didático centrado no usuário, para auxiliar o processo de ensino e aprendizagem de conceitos de Algoritmos e Programação para crianças do Ensino Fundamental 1: Séries Iniciais.

1.1 Questão de Pesquisa

A formação em Licenciatura em Computação garante a expansão de diversas perspectivas na área da educação. O ensino de programação e robótica nas escolas têm se tornado essencial no contexto de Globalização e Informatização que a sociedade do conhecimento está inserida, mostrando-se uma escolha oportuna e relevante para esta pesquisa. Enquanto pesquisador, ter concluído a formação em Licenciatura em Computação foi um primeiro passo importante, pois nesse período que os primeiros contatos com a problemática e as vantagens decorrentes do ensino e aprendizagem destes foram realizados. Após isso, trabalhar em uma Escola de Programação e Robótica Educacional, trouxe maior clareza quanto aos temas relacionados aos desafios inerentes à escolha por estudos na área de ensino aprendizagem de Algoritmos e Programação.

Foi enquanto professor de programação para crianças que as contradições inerentes ao ensino e aprendizagem se acentuaram. Ao se conectar com os alunos desta escola, com idade entre 6 e 15 anos, foi identificada uma dificuldade na mediação da aprendizagem de conceitos de Algoritmos através plataformas online, *softwares* e aplicativos comumente usados para tal propósito. Muitos alunos demonstravam desinteresse nas ferramentas ao longo das aulas por terem elementos repetitivos na interface. Ademais, essas soluções usavam representações gráficas muitas vezes desconhecidas do contexto sociocultural das crianças, visualizando-se neste contexto uma inquietação: os alunos não veem progressão nas atividades por causa dos elementos repetitivos da interface e não reconhecem as representações gráficas dos softwares/aplicativos como objetos lúdicos de seu contexto, gerando conseqüentemente uma diminuição de engajamento em pouco tempo.

Da mesma forma, de acordo com as informações apresentadas, temos como questão da pesquisa:

Como desenvolver uma solução para ensino e aprendizagem de Algoritmos e Programação no Ensino Fundamental 1: Séries Iniciais que tenha elementos lúdicos relacionados à realidade sociocultural desses alunos no contexto Amazônico?

Assim, a função do Designer neste contexto, é realizar estudos para identificar as necessidades dos usuários e viabilizar as possíveis soluções que tenham maior impacto, com menor custo, propondo soluções que sejam úteis, fáceis de usar e prazerosa para que o aluno tenha uma melhor experiência no processo de ensino-aprendizado dos conceitos de Algoritmos e Programação. Dessa forma, verifica-se que os estudos em Design Centrado no Usuário e Design de Experiência do Usuário no desenvolvimento de soluções para o ensino e aprendizagem de Algoritmos e Programação se adequa ao contexto da Escola de Robótica e Programação que é o objeto de estudo da pesquisa e pode contribuir para apaziguar as inquietações supracitadas.

1.2 Delimitação do Problema

A educação é um direito amparado por lei. Por isso, a nossa legislação apresenta uma série de obrigações, parâmetros e recomendações para as atividades educacionais curriculares e extracurriculares. A Educação Brasileira é regulamentada a partir da Constituição Federal de 1988, na Lei de Diretrizes e Bases da Educação e no Plano Nacional de Educação, o qual propõe metas para serem alcançadas na próxima década. Para complementar a legislação, foi elaborada a Base Nacional Comum Curricular - BNCC que define os conhecimentos fundamentais que todos os alunos da Educação Básica têm direito a aprender.

A BNCC apresenta 10 competências gerais, sendo a quinta voltada para a Cultura Digital.

5. Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva. (BNCC, 2017, p. 7)

A inclusão da Cultura Digital na BNCC é um reflexo do contexto social do mundo informatizado em que vivemos. Esta competência busca não somente motivar o uso das tecnologias na educação, mas também construir um senso crítico e autônomo no estudante, a fim de que os conceitos aprendidos desenvolvam habilidades para serem usadas no dia a dia. Sendo assim, a Cultura Digital

destacada na BNCC compreende a capacidade dos estudantes de usar a tecnologia de forma crítica, com ética, para obter informação e para produzi-la.

A proposta do Colégio Connexus junto a Manaós Tech visa atender a essa necessidade. O Connexus tem engajado esforços para implementar uma infraestrutura e espaço de aprendizagem com recursos e ferramentas mais inovadoras e tecnológicas para metodologias plugadas (com o uso do computador), bem como recursos para trabalhar de forma desplugada (sem o uso do computador). Também se faz presente no Laboratório de Inovação uma docente com formação na área Tecnológica que auxilia o professor da Manaós Tech nas atividades de aprendizagem.

A escola possui um currículo com uma estruturação que dá maior liberdade para mudanças pontuais e fomentos a projetos transversais ao longo das séries. Esta iniciativa da escola visa proporcionar momentos para o desenvolvimento de habilidades e conhecimentos oriundos do Pensamento Computacional considerados como requisitos mais importantes neste século, tais como criatividade, inovação, resolução de problemas, sistematização de tomada de decisão, construção e desenvolvimento de projetos computacionais e robóticos.

Como toda prática pedagógica, o ensino de computação possui dificuldades inerentes ao seu objeto. Devemos, portanto, entender que a computação é uma ciência que tem como objeto de estudo a resolução de problemas por meio da criação e uso de algoritmos, isto é, a resolução de problemas de forma lógica e sem ambiguidades (ARAÚJO, et al., 2015). Em razão disso, ensinar conceitos de computação exige múltiplos níveis de abstração.

Desta forma, vários projetos têm se desenvolvido no Brasil para disseminar na prática o Pensamento Computacional, a fim de que estes conceitos sejam aplicados nos mais diversos campos do conhecimento com foco na resolução de problemas. Alguns destes projetos visam claramente desenvolver habilidades e atitudes consideradas fundamentais dentro do Pensamento Computacional, outros visam introduzir conceitos de Computação (BORDINI, et al., 2016).

Este desafio é ainda maior quando os agentes do conhecimento são crianças. Por não terem uma motivação extrínseca é preciso reforçar elementos de motivação intrínseca para que haja a dedicação e esforço necessário para a aprendizagem. Por isso, para ensinar conceitos de computação para crianças, tais como Algoritmos e Programação, o uso de elementos lúdicos é uma das possíveis estratégias para aumento de engajamento. Estes elementos precisam ser reconhecidos como parte do contexto dos alunos a fim de aumentar o engajamento no processo de ensino e aprendizagem.

Atualmente, diversas plataformas online são usadas na introdução de conhecimentos de Algoritmos e Programação que oferecem diferentes ambientes e elementos de cenários, personagens e ferramentas para resolução de problemas, desde os mais simples, até os mais complexos.. Entre elas, destacam-se: *Scratch*¹, *CodeSpark*², *CodeMonkey*³, *Code Combat*⁴, *Code.org*⁵ e *Run Marco!*⁶.

A partir da práxis docente deste pesquisador na realização de experimentos utilizando as supracitadas plataformas, foi possível levantar uma série de possibilidades diferentes para o ensino de computação. A riqueza dos resultados reforçou muitas das hipóteses particulares do pesquisador explicitou a necessidade de uma abordagem mais estruturada e sistematizada para a continuidade das atividades.

Assim, a proposta desta pesquisa visa construir uma solução para trabalhar conceitos de Algoritmos e Programação, com elementos comuns a todos os alunos de faixa etária de 6 a 7 anos, que proporcione uma atividade que conduza os alunos por experiências engajadas e emocionantes.

As metodologias e técnicas do Design de Experiência do Usuário irão direcionar o desenvolvimento desta solução centrada nos alunos do Colégio Connexus, a fim de que a mediação dos conceitos seja feita por meio de uma ferramenta que possua elementos lúdicos do contexto sociocultural dos alunos e estes sejam reconhecidos como tal. Desta forma, a mediação dos conceitos de Algoritmos e Programação em uma sala de aula com diferentes tipos de alunos se tornará mais eficiente.

1.3 Objetivos

1.3.1 Geral:

Desenvolver um jogo centrado no usuário para ensino e aprendizagem de Algoritmos e Programação no Ensino Fundamental 1: Séries Iniciais, por meio da computação desplugada.

1.3.2 Específicos:

¹ Scratch: <https://scratch.mit.edu>

² CodeSpark: <https://codespark.com/play>

³ Code Monkey: <https://www.codemonkey.com>

⁴ Code Combat: <https://codecombat.com/>

⁵ Cod.org: <https://code.org/learn>

⁶ Run Marco!: <https://runmarco.allcancode.com>

- Identificar as principais correntes de pensamento, conceitos e autores relacionados ao ensino e aprendizagem de algoritmo no Ensino Fundamental 1: Séries Iniciais.
- Identificar as principais correntes de pensamento, conceitos e autores relacionados ao desenvolvimento de soluções de design centrado no usuário para ensino e aprendizagem no Ensino Fundamental 1: Séries Iniciais.
- Definir métodos e técnicas para o desenvolvimento de soluções de design centrado no usuário para ensino e aprendizagem no Ensino Fundamental 1: Séries Iniciais.
- Desenvolver um modelo experimental para aplicar e validar elementos lúdicos ensino e aprendizagem de algoritmo por meio de um objeto de aprendizagem desplugado relacionados à realidade sociocultural desses alunos.
- Apresentar um Produto Viável Mínimo - MVP de um objeto de aprendizagem para o ensino aprendizagem de conceitos de Algoritmos e Programação no Ensino Fundamental 1: Séries Iniciais baseado em computação desplugada.

1.4 Justificativa

A utilização ilimitada da computação em diversos setores e principalmente no contexto escolar pode ser observada de várias maneiras. Se torna cada vez mais evidente um cenário onde diversas atividades serão automatizadas e realizadas por meios computacionais. Essa perspectiva nos leva a pensar que o domínio das tecnologias é um requisito imprescindível, tais como fala, escrita e leitura, para vivermos nos dias de hoje e atuarmos na sociedade de forma efetiva.

Contudo, essa mudança técnica não pode perder de vista os aspectos culturais. Assim, esta pesquisa dispõe-se a adentrar em reflexões sobre um futuro próximo onde são apresentadas soluções para os problemas do dia a dia cada vez mais automatizadas. Por conseguinte, é certo que é preciso contextualizar e propor novas formas de potencializar habilidades e competências básicas relacionadas a área da Computação desde os primeiros anos da educação, isto é, no Ensino Fundamental 1: Séries Iniciais, a fim de se preparar os futuros indivíduos para este cenário.

Emerge a importância do Pensamento Computacional. Afinal, neste contexto, o Pensamento Computacional tem sido apresentado como uma nova forma de pensar, de raciocinar e resolver problemas do cotidiano, baseado em conceitos da Ciência da Computação. A disseminação do Pensamento Computacional “visa promover habilidades que auxiliam tanto para a solução de problemas do dia a dia como para a solução de problemas mais complexos da sociedade” (BORDINI, 2016, p. 211), utilizando-se de técnicas de decomposição de problemas maiores em partes menores,

reconhecimento de padrões, abstração de informações importante e algoritmos com uma estruturação da resolução passível de ser repetida.

Porém, esses conceitos ainda são distantes da realidade das escolas. O currículo das Escolas Brasileiras ainda não contempla conteúdo da Ciência da Computação, mas com o aumento da abrangência do ensino destes conceitos, as reflexões acerca da disseminação do Pensamento Computacional e o ensino aprendizagem de Algoritmos e Programação são de urgente e de extrema importância.

Com relação à **justificativa científica**, observa-se um crescimento de discussões sobre o ensino de programação na educação nos últimos anos. Mesmo a passos lentos, diversas iniciativas têm tentado apontar a necessidade de reconhecermos os conceitos de Ciência da Computação como disciplina básica. Nos Estados Unidos, um currículo específico conhecido como *Model Curriculum for K-12* “defende que profissionais de diferentes áreas precisam ter conhecimentos de Computação, enquanto ciência, na busca por soluções de problemas, bem como na construção dessas soluções” (ARAUJO, *et al.* 2015, p. 131).

De acordo com Chubachi e Vitória (*apud* ARAUJO, *et al.* 2015, p. 131), “o ensino de programação é uma alternativa às práticas tradicionais”. A formação em Licenciatura em Informática deste pesquisador permite ter uma outra visão sobre o processo de ensino e aprendizagem. A desconstrução das práticas tradicionais pode se dar com a inovação dos métodos de ensino de forma a considerar o aluno como o centro de toda a práxis docente, isto é, o professor deve mediar as atitudes no ambiente da sala de aula, conduzindo os alunos por experiências práticas, para se desenvolverem como cidadãos mais críticos, autônomos, criativos e transformadores.

Têm-se notado alguns esforços na criação de novas metodologias de ensino e desenvolvimento de novas soluções para auxiliar professores e alunos no processo de ensino e aprendizagem (KENSKI, 2003, p. 4). Diferentes objetos de aprendizagens, recursos didáticos e metodologias têm surgido com o objetivo de oferecer uma experiência diferente aos usuários, ou seja, os alunos. Softwares, aplicativos, plataformas e jogos que facilitam o ensino de lógica de programação como o *Codeacademy*, *CodeSpark* e *LightBot* tem se tornado cada vez mais comum na sala de aula.

Entretanto, as soluções que visam desenvolver o Pensamento Computacional no Ensino Fundamental 1: Séries Iniciais devem ir além de atender puros conceitos da Computação. Precisam incluir elementos lúdicos reconhecidos pelos alunos dentro do seu contexto sociocultural, a fim de facilitar a compreensão dos conceitos e gerar engajamento na realização das atividades, proporcionando assim boas experiências de aprendizagem.

Este princípio está diretamente ligado ao conceito de Design de Experiência do Usuário, no qual o Designer tem um papel primordial de identificar necessidades e dificuldades e satisfazê-las com uma solução útil, agradável e que os usuários tenham prazer em usar. Desta forma, esta pesquisa visa investigar como o Design de Experiência do Usuário pode nortear o desenvolvimento de uma solução para trabalhar conceitos de Algoritmo e Programação no Ensino Fundamental 1: Séries Iniciais.

Dado o escopo da pesquisa, considera-se importante situar o espaço concreto e simbólico em que ela se desenrola. Segundo o Censo (IBGE, 2021), o estado do Amazonas ocupa um território de 1.559.167,878 km², com uma população estimada em 4.269.995 pessoas divididas em 62 municípios e na capital Manaus.

Os dados do Portal do Governo Brasileiro ⁷, mostram que cerca de 152.742 alunos estão matriculados no Ensino Infantil (Creche e Pré-escolar) e 702.763 no Ensino Fundamental 1: Séries Iniciais (1º a 9º ano). No Pré-Escolar, 108.214 são escolas Municipais, 8 são Estaduais e 9.728 são Privadas.

Em relação ao Índice de Desenvolvimento da Educação Básica, o Amazonas encontra-se em 19º lugar, com nota 5,5, enquanto o primeiro lugar, São Paulo, encontra-se com 6,7. Os Anos Iniciais em Escolas Públicas do Amazonas possuem índice 5,3, enquanto em São Paulo possuem índice 6,5. Em Escolas Privadas, o Amazonas tem índice 7,0 e as Escola Privadas de São Paulo possuem índice 7,6.

Estes dados representam um problema no ensino-aprendizagem no Estado do Amazonas. As colocações apresentadas até aqui a respeito da nova forma de racionalizar por meio do Pensamento Computacional permite aguardar resultados significativos no processo de ensino e aprendizagem de forma interdisciplinar, isto é, os alunos adquirem ferramentas mentais para construir soluções práticas em diferentes contextos da sala de aula e aplicar tais ferramentas na construção de diferentes saberes.

Por fim, a pesquisa se mostra importante para o próprio **profissional de design**, visto que cada vez mais tem se tornado importante dominar técnicas, metodologias, processos e ferramentas para desenvolver soluções baseadas no usuário, visando a resolução de problemas da sociedade, dos mais simples aos mais complexos.

⁷ Disponível em: <https://cidades.ibge.gov.br/>

1.5 Delimitação do Estudo

Situamos essa pesquisa dentro da área do Design, com foco em Design da Experiência do Usuário. Esta disciplina envolve tópicos como pesquisa com usuários, mapeamento de jornada, levantamento de necessidades, estratégia de negócio, design de interfaces, dentre outros assuntos.

Por ter como objetivo o desenvolvimento de uma solução de forma a auxiliar o processo de ensino e aprendizagem de Algoritmos e Programação no Ensino Fundamental 1: Séries Iniciais, tem-se como tópico inicial e fundamental do Design de Experiência do Usuário, a construção de soluções centradas no usuário. Esta abordagem permite direcionar todas as decisões de forma a viabilizar o desenvolvimento de uma solução que tenha maior impacto, menor esforço e menor tempo de implementação, podendo ser incrementada a partir do levantamento de novas necessidades.

A pesquisa se desenvolveu no espaço físico do Colégio Connexus, uma vez que possui a disciplina de Laboratório de Inovações que visa ensinar conceitos relacionados à programação e robótica aos alunos.

Assim, este estudo objetiva construir um produto/solução para auxiliar as atividades deste laboratório, por meio da Computação Desplugada. O produto sucede das pesquisas exploratórias, do levantamento de necessidades e dificuldades, perpassando por levantamento de requisitos, prototipação, testes e implementação a partir do processo de Design de Experiência do Usuário.

1.6 Estrutura do Documento

O segundo capítulo desta dissertação estabelece o alicerce teórico que sustenta a pesquisa, esclarecendo conceitos relacionados ao seu foco por meio de revisão bibliográfica. Inicia-se apresentando a origem e o desenvolvimento da Computação enquanto ciência, seu propósito geral até o surgimento do conceito de Pensamento Computacional. Este compreende um processo mental capaz de resolver problemas complexos atrelados ou não ao mundo computacional. Posteriormente, aborda-se sobre o ensino e aprendizagem dos conceitos do Pensamento Computacional no Brasil, examinando como as escolas tem se adequando a essa nova expertise apresentada tão quanto importante a ler e escrever. A posteriori, discute-se como os recursos didáticos de tem auxiliado no processo de ensino e aprendizagem desses conceitos para crianças em idade média de 6 a 7 anos, estudantes do Ensino Fundamental 1: Séries iniciais. Por fim, são apresentadas as Metodologias de Design centradas em usuários, que podem auxiliar no processo de concepção de recursos lúdicos didáticos para crianças.

O terceiro capítulo, discorre sobre a metodologia *Design Thinking*, perpassando pelas as fases de Imersão, Análise e Síntese, Ideação e Prototipação, descrevendo as técnicas de levantamento e análise de dados e resultados obtidos em cada etapa do processo, visando possibilitar a reprodução de outros autores em diferentes cenários.

No Capítulo 4, são apresentados os resultados preliminares do MVP, concebido a partir da metodologia de Design Centrada no Usuário, o *Design Thinking*.

No Capítulo 5, são feitas as considerações finais e por fim, são listadas todas as referências utilizadas.

CAPÍTULO 2

Referencial Teórico

Este capítulo busca apresentar um apanhado geral de temas pertinentes para melhor compreensão da pesquisa aqui proposta, de modo que foram abordados temas como a natureza e história da Computação, Design Thinking, Design Centrado em Usuário e Experiência do Usuário.

2.1 A natureza da Computação

Em uma pesquisa acadêmica, podemos abordar o problema de pesquisa de diversas maneiras, dentre elas, a perspectiva histórica. Dessa forma, uma primeira questão que emerge sobre a computação é quanto a sua origem. Nesse sentido, os estudos a partir da Matemática foram fundamentais para o desenvolvimento da Computação.

Entre discussões filosóficas e sistemas matemáticos simples e complexos, em um dado momento da história, o homem precisou de uma forma mais estruturada e analítica de expressar a exatidão e a certeza de fatos. Após longos anos de história humana, o ambiente científico finalmente se assentava sobre uma matemática que se reconhecia completa, esperando-se que “qualquer sentença expressa na linguagem formal da matemática correspondesse uma prova de sua veracidade ou falsidade” (CAFEZEIRO, et al., 2016).

Contudo, nas primeiras décadas do século XX a matemática viveu um período de grandes desafios. Apesar de que desde seu surgimento, havia diversas críticas quanto às premissas que serviram de base para o pensamento matemático, como os paradoxos de Zenão, está avançou legitimada por sua eficácia técnica. Até que no último século sua legitimidade foi colocada sistematicamente em questão.

Segundo (CAFEZEIRO, et al., 2016), os matemáticos começaram a ser instigados por questões que pareciam pôr em dúvida as bases de sustentação da matemática, calcadas no Silogismo, - sistema de pensamento baseada em 2 (duas) premissas e 1 (uma) conclusão – criado por Aristóteles e que ficou conhecida como lógica tradicional. A Lógica Tradicional foi criada para ajudar pessoas a pensar de forma mais efetiva, através do sistema de premissas e conclusão.

Ainda nesse período, surgiram publicações importantes na área. Matemáticos como Alfred North Whitehead e seu aluno Bertrand Russell publicaram nos anos de 1910, 1912 e 1913 o *Principia Mathematica*, tradução livre do latim: Princípios matemáticos, uma obra de três volumes sobre fundamentos da matemática. Em 1931, Kurt Gödel fez novos ensaios lógicos-matemáticos que culminam em Incompletude, isto é, a impossibilidade de um sistema matemático ser ao mesmo tempo completo e livre de contradições.

Gödel conclui que “não é possível que um sistema matemático expressivo o suficiente para representar sentenças como esta seja, ao mesmo tempo, completo e consistente” (CAFEZEIRO, et al., 2016, p. 123 *apud* GÖDEL, 1965). Estes autores contribuíram com o aperfeiçoamento da lógica, utilizando técnicas de símbolos lógicos para expressar ideias. Dava-se início então a transição do sistema de pensamento baseado no Silogismo de Aristóteles para um novo sistema de pensamento

que ficou conhecido como Lógica Computacional - LC, a qual usava símbolos lógicos para representar expressões que pretendiam resolver uma questão ou um problema. A partir de então, passaram a inferir esforços para definir o escopo da matemática, isto é, definir o conceito do que vem a ser "mecânico", "calculável" ou "computável".

Foi através da relação estreita com a possibilidade de realizar cálculos numéricos que a Lógica Computacional ganhou força. O matemático inglês Alan Mathison Turing (1912-1954), foi responsável por desenvolver em 1936 o que ficou popularmente conhecida como a Máquina de Turing - MT, um dispositivo teórico capaz de realizar operações simples, mas que quando combinadas, ou seja, quando usada a Lógica Computacional, permitia resolver qualquer problema computável (PETZOLD, 2008).

Em um de seus artigos mais clássicos, publicado 1936, *"On computable numbers, with an application to the Entscheidungsproblem"*, Turing definiu as bases estruturais da Computação como ela é conhecida hoje. Esta é uma prova matemática da viabilidade de realização de cálculos numéricos a partir de uma máquina conceitual que usa uma representação que hoje conhecemos como binária (0 e 1).

Peter J. Denning, Cientista da Computação e Escritor americano apresenta a origem da Computação como ciência a partir de outras Ciências. Para o autor, a Ciência da Computação é a ciência que estuda os "processos de informação e suas interações com o mundo" (Denning, 2005, p. 27). Ele afirma ainda que a Ciência da Computação é uma mistura harmoniosa e plena da ciência (ciências naturais), engenharia e matemática e tal harmonia pode ser justificado nas atividades e estudos realizados por Cientistas da Computação.

Estas atividades são principalmente ciência quando se estuda algoritmos experimentais, ciência da computação experimental e ciência computacional. São principalmente engenharia quando se estuda design, desenvolvimento e engenharia de software. E, são principalmente matemáticas quando há estudos que envolvam algum grau de complexidade computacional, software matemático e análise numérica. Contudo, a maioria das atividades da Ciência da Computação combinam os três conjuntos de atividades (Denning, 2005).

O avanço na área de engenharia e matemática auxiliaram a construção de máquinas computacionais cada vez mais robustas, com o poder de processar informações cada vez mais complexas, em maior quantidade e em menor tempo. Neste cenário, os computadores com características, mas próximas do que conhecemos hoje foram sendo introduzidas no mercado. Novas empresas de Hardwares (componentes físicos e palpáveis do computador) e Softwares (programas

executados sobre o hardware) foram emergindo, surgindo então a necessidade de categorizar esta nova área de conhecimento como ciência.

A Computação não faz ciência somente quando constrói computadores e softwares. Denning (2005) afirma que o campo da Ciência da Computação relaciona-se mais com as informações e processos do que puramente com computadores. Ao usar informações e processos para realizar experimentos, medições, verificações por observações e até mesmo previsões, percebe-se plenamente os paradigmas científicos na Computação enquanto Ciência.

Desta forma, a Ciência da Computação dá apoio a outras áreas do conhecimento usando os computadores para tratar informações de processos artificiais e naturais, não como objetos de estudos em si. Entende-se por processos artificiais os produzidos pelo homem, tais como simulações computacionais que seriam difíceis ou impossíveis de serem executadas no mundo real. Os processos naturais, por outro lado, simplesmente acontecem no mundo real e podem ser observáveis, estudados em proporções e escalas maiores e em cenários diferentes, passando a ser processos artificiais a partir de então.

A partir destas definições, a Computação se qualifica como ciência exata, visto que estuda processos de informação que ocorrem no mundo físico, isto é, os cientistas trabalham com um corpo de conhecimento sistematizado e aceito, fazendo que muito da Ciência da Computação seja aplicada⁸.

O Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq é uma entidade ligada ao Ministério da Ciência, Tecnologia e Inovações - MCTI para fomentar pesquisa e incentivar a formação de pesquisadores no Brasil (Governo Federal, 2022). O CNPq divide o conhecimento humano em grupos, chamada Árvore do Conhecimento.

A Ciência da Computação (1.03.00.00-7) faz parte do grupo Ciências Exatas e da Terra, que, segundo o CNPq, pode ser dividida em subáreas que incluem Teoria da Computação, Computabilidade e Modelos de Computação, Linguagem Formais e Autômatos, Análise de Algoritmos e Complexidade de Computação, Matemática da Computação, Modelos analíticos e de Simulação, Metodologia e técnicas da computação, Linguagens de Programação, Processamento Gráfico (graphics), Hardware, dentre outros. Todas as áreas estão disponíveis no site do Ministério da Educação.

⁸ Ciência aplicada está ligada a aplicações do conhecimento para geração de outros e/ou para a solução de problemas práticos.

Em conformidade com Denning (2005), nota-se que a Ciência da Computação é usada constantemente para apoiar pesquisas que perpassam por outras áreas do conhecimento, tais como, Linguagens, Matemática, Astronomia, Física, Design, Engenharia, entre outros.

Os conhecimentos em Computação são muito importantes para a vida na sociedade contemporânea. Assim como os conhecimentos básicos da Matemática, Física e Filosofia possibilitam o homem contar, abstrair, relacionar, medir e pensar, a Computação parece carregar consigo “mecanismos singulares de raciocínio para resolução de problemas, cujas aplicações ultrapassam as fronteiras da Computação em si” (BARCELOS & SILVEIRA, 2012, p. 3).

Considere, por exemplo, o uso de meios computacionais nas Ciências Exatas e da Terra, que tem como campo de estudo a utilização de cálculos e lógica para gerar previsões e medições precisas. Na Matemática a computação possibilita a geração de desenhos geométricos complexos, calcular o PI (π) com a precisão de 3,3 trilhões de casas decimais. Foi possível ainda encontrar o maior número primo até o momento que é de 22.338,618 dígitos (GIMPS, 2016).

Na Química, a Computação tem sido aplicada na simulação de reações químicas, sem a necessidade de arriscar as vidas dos pesquisadores durante a execução, tal como foi a simulação de uma chama de fogo que analisou 500 milhões de pontos em 120 mil incrementos de tempo, isto é, em uma fração de segundo. Desta simulação foi possível encontrar 11 tipos diferentes de moléculas e 21 reações químicas. Ao todo, foram gerados 30 Terabytes de dados, isso é equivalente a aproximadamente 40 mil CDs ou 7 mil DVDs de uma camada, ou encher os discos rígidos de 100 gigabytes em 300 computadores desktop (SANKARAN, 2007).

Nas Ciências Biológicas, cuja o objeto de estudo é a vida em diversas escalas, desde o microscópico até o meio ambiente em que o indivíduo está inserido, temos contribuições da Computação na Biologia, tornando possível o mapeamento do genoma humano, identificação de enzimas, simulações de adaptações de seres vivos em ambientes diferentes. A associação entre as áreas da Botânica, Zoologia, Genética e Computação “podem ser empregados na solução de questões ecológicas, que se referem principalmente ao significado das interações animais-plantas” (DEL-CLARO, et al., 2009, p. 16).

Nas Engenharia, cujo objeto de estudo está baseado no uso de recursos naturais para construir, criar e conceber novas ferramentas que geram benefícios à vida dos seres humanos, podemos ver o emprego da Computação na Engenharia de Software, no desenvolvimento de programas e na Mecatrônica com a construção de robôs que automatizam tarefas das mais simples como as do dia a dia, as mais complexas como em sistemas de produção de alta escala. Na Engenharia Civil diferentes

softwares possibilitam a construção de projetos arquitetônicos, simulações gráficas de espaço, ambiente, cores e estruturas antes mesmo da construção no mundo real, evitando desperdício de tempo, esforço e recursos.

Nas Ciências da Saúde, cujo foco dos estudos está na busca por diagnósticos, tratamentos e prevenção de doenças para o bem estar de pessoas e animais, a Computação tem ajudado principalmente no desenvolvimento de medicamentos, cirurgias remotas ou em simuladores, previsão de tempo de contaminações de doenças em determinados ambientes (BRACKMANN, 2017).

As Ciências Humanas estão focadas em entender os comportamentos e necessidades da sociedade. Neste contexto a Computação tem sido aplicada para cálculo e estatística de Censos, tais como as realizadas pelo IBGE no Brasil. Por meio Instituto Brasileiro de Informática (IBI), modernos sistemas de computação e de processamento eletrônico de dados foram implantados para ampliar a performance do órgão desde a coletar e manipulação, até o armazenamento de informação (CONTEL, 2014). Segundo Gonçalves (1995), a incorporação desses sistemas oriundos da Computação torna a ação do IBGE ainda mais eficiente, expandindo a capacidade efetiva no auxílio as políticas públicas em curso. Tais recursos aumentam ainda significativamente a capacidade de determinar padrões e características de indivíduos da mesma região e de regiões diferentes.

As Ciências Agrárias estudam uma mistura de agronomia com ciências exatas e biológicas. Estudos como os de Alves (et al, 2016) mostram como os aplicativos e softwares tem auxiliado os agricultores com atualizações de informações em tempo real, utilizando a *Web* e Dispositivos Móveis. Essas ferramentas visam auxiliar profissionais do agronegócio nos processos de tomada de decisão no campo.

Ao contrário da Ciências Humanas que focam no indivíduo, as Ciências Sociais Aplicadas concentram seus estudos na sociedade e coletividade. Na Administração, por exemplo a Computação tem auxiliado a Governança em TI no gerenciamento de recursos humanos e tecnológicos das empresas.

Na segurança, os sistemas de controle de acesso, de armazenamento de informações restritivas com segurança via Computação em Nuvem tem sido muito bem aceito. Para Luftman (1996), o grau de competitividade atribuído ao negócio de uma empresa está relacionado as implementação das Tecnologias de Informação e da Computação para inovar, operar produtivamente, conectar-se à rede de negócios (*e-business*), dispor de informação estruturada para nortear a tomada de decisões (*BI - Business Intelligence*), gerenciar suas operações e aplicação de seus recursos (*ERP - Enterprise*

Resources Planning) e conhecer e interagir com seus clientes e usuários (*CRM - Customer Relationship Management*).

A computação também impacta as áreas da Linguística, Letras e Artes. A Computação Gráfica tem inovado a forma de fazer Cinema, por exemplo. Antes era impossível fazer um filme sem ator e sem estúdio. Hoje é possível modelar seres humanos “virtuais” em três dimensões, que começam a assumir formas e movimentos que nos confundem e impressionam (OKAMOTO & PAIVA, 2005).

Muitos dos efeitos, conhecidos como efeitos cinematográficos são produzidos no Computador, visando além de garantir a qualidade e a realidade dos fatos para o telespectador, também a segurança dos atores e equipe cinematográfica, economia de recursos e execução de cenas em ambientes de difícil acesso ou irreais.

Diversos aplicativos e programas tem surgido como o objetivo de auxiliar o letramento e alfabetização de crianças, jovens e adultos. Neste sentido, a Computação tem contribuído fortemente com as linguagens orais e visuais, e até mesmo com a acessibilidade de Pessoas com Deficiência – PcD às Linguagens de Sinais.

Na Educação, a Computação foi tão amplamente abraçada que os impactos trazidos pela pandemia da COVID-19 foram sentidos em menores proporções se comparação com outras áreas, apesar dos desafios emergentes desses contextos. Com a modalidade de educação remota, mediada por tecnologia já amplamente utilizada, foi possível esvaziar os ambientes escolares para evitar aglomeração sem cessar completamente as interações entre professores e alunos, e sem perder por completo 2 anos de avanço educacional.

Em outras palavras, a Educação não parou em face da pandemia e foi na Computação que pode achar as maiores possibilidades para resiliência, utilizando-se dela para reinventar seus métodos, se apropriando de ferramentas oriundas de estudos das Ciências da Computação para permanecer contribuindo com o desenvolvimento humano nesses anos tão sofridos de pandemia.

Dito isto, é mais que evidente que a Ciências da Computação impacta e até transforma outras áreas do conhecimento. Esta expressiva digitalização do cotidiano encontra-se prevista inclusive na Lei de Diretrizes e Bases (LDB), no artigo 32, ao afirmar que além dos supracitados conhecimentos, é objetivo da formação básica do cidadão “a compreensão do ambiente natural e social, do sistema político, da tecnologia, das artes e dos valores em que se fundamenta a sociedade” (SOUZA & SILVA, 1997, p. 53).

A cerca de uma década, a comunidade científica houvera identificado que futuras descobertas científicas necessitariam de cientistas com inegociáveis conhecimentos de Computação. Foi a partir desta percepção que novas áreas foram surgindo, tais como Inteligência Artificial, Biologia Computacional, Simulação Molecular, Bioquímica, dentre diversas outras. Desta forma, o pensamento estruturado com os conceitos provenientes da Computação se tornou mais interessante, visto que seu alcance vai além da comunidade científica, podendo ser utilizadas por muitas outras profissões, tais como Engenheiros, Arquitetos, Músicos, Médicos, Químicos, Professores, possibilitando o estudo de objetos e a soluções de problemas em diferentes cenários (BRACKMANN, 2017).

Com o passar do tempo, a sociedade se viu diante de um mundo com uma enorme quantidade de dados que geram informações a todo instante. McGarry (1984, citado por PINHEIRO, 2014, p. 5) faz uma distinção entre dado e informação. A primeira é "matéria prima a partir da qual se pode estruturar informação" e a segunda, "mais complexa e estruturada do que dado". Em outras palavras, os dados, são pequenos fragmentos de informações (símbolos, números, códigos, entre outros) que isolados não podem ser compreendidos pelo homem. Os dados podem se originar de diversas fontes e em diferentes formatos. Estes fragmentos (dados) podem ser processados por um sistema lógico e chegar a uma conclusão logica. Já as informações são compreendidas em completudes pelo homem. São como uma reunião de dados de forma a se tornarem úteis para algum fim. As máquinas possuem maior capacidade para processar dados mais facilmente e em menor tempo do que o homem. Por esta razão, dados são inseridos pelo homem em um computador para que ao serem processados e após convertidos, saem em forma de informação para o homem.

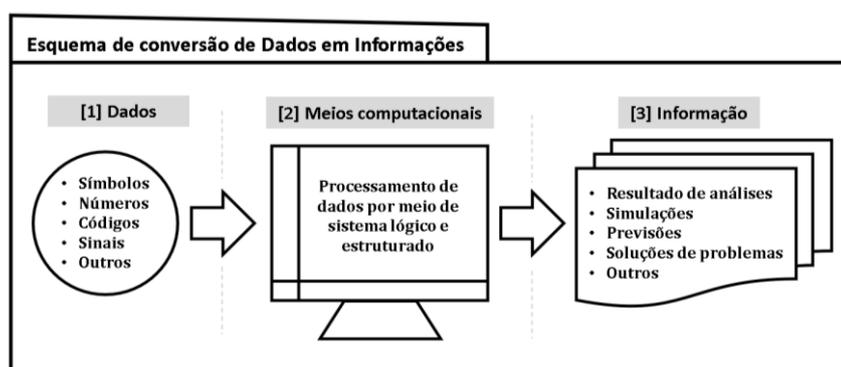


Figura 1. Esquema de conversão de dados em informações.
Fonte: O autor (2024).

Ora, para que possamos usar os meios computacionais na resolução de problemas simples e complexos, é necessário saber converter informações do mundo real em dados inteligíveis por um

computador. O segundo passo requer habilidades de construir sistemas lógicos e estruturados de pensamento que irão processar tais dados, seja ele um programa de computador, um simulador, ou um simples passo a passo de um robô. O resultado na saída do processamento é uma solução válida e útil, o que pode se tornar conhecimento. A Computação de nada adianta se não aprendermos a transformar as informações em dados compreensíveis pelas máquinas. É indispensável que os indivíduos da sociedade do conhecimento desenvolvam habilidade e competências que lhes permita realizar este processo de conversão de informações em dados e construção de sistemas de pensamento lógicos para processar esses dados para determinados fins.

2.2 Pensamento Computacional

O Pensamento Computacional possui uma árvore ancestral riquíssima que permeia desde o início da computação na década de 40. John Von Neumann, neste mesmo tempo, afirmara que os computadores seriam não apenas uma ferramenta para ajudar a ciência, mas uma maneira de fazer ciência (DENNING, 2009).

Na década de 1960, o cientista da computação Alan Perlis defendeu a necessidade de estudantes universitários de todas as disciplinas aprenderem programação e as “teorias da computação” (Guzdial, 2008).

Entre 1950 e 1960, o termo “Pensamento Algorítmico” começou a ser utilizado. Este compreendia uma orientação mental para formular problemas baseados e conversões de dados de entrada ([1], vide Figura 1) para informações de saída ([3], vide Figura 1) e procurar algoritmos para realizar tais conversões ([2], vide Figura 1). Em 1975, o Físico Kenneth Geddes Wilson promoveu a ideia de que computação e a simulação era maneiras de fazer ciência não disponíveis anteriormente. Wilson recebeu o Nobel de Física em 1982, enquanto atuava na Universidade de Cornell, Ithaca, EUA. O prêmio foi fruto de pesquisa baseados no uso do Pensamento Algorítmicos, que originou a “criação de modelos computacionais cujas simulações produziram novos entendimentos radicais de mudanças de fase em materiais” (DENNING, 2009).

No entanto, Seymour Papert foi pioneiro na ideia de crianças desenvolvendo pensamento procedimental e estruturado através da programação LOGO (Papert, 1980, 1991). Em 1972, Seymour Papert e Cynthia Solomon já haviam publicado o artigo “*Twenty things to do with a computer*” – tradução livre para “Vinte coisa para fazer com um computador” o qual já compartilhavam de ideias semelhantes ao Pensamento Algoritmo. Papert ainda chegou a mencionar o termo “Pensamento Computacional” em sua obra “*Mindstorms: Children, Computers, And Powerful Ideas*”, ao abordar a

cultura dos computadores e das tecnologias no ensino de crianças. Contudo não houve esforços para disseminação de seus princípios naquela época (PAPERT, 1980).

Foi em 2006 que Jeanette Wing popularizou o termo “Pensamento Computacional” com a publicação de um artigo intitulado “*Computational thinking*” na revista *Communications of the ACM*. Na ocasião, Wing era professora e presidente de Ciência da Computação e chefe do Departamento de Ciência da Computação da Carnegie Mellon University, Pittsburgh, USA. Wing argumenta em seu artigo que a maneira de que os Cientistas da Computação pensam sobre o mundo é útil para outros contextos. A autora segue com exemplos que dão base para sua afirmação, como o de que a Aprendizagem de Máquina revolucionou a Estatística; que na Biologia, a Ciência da Computação tem ido além da capacidade de pesquisar grandes quantidades de dados de sequência tentando reconhecer padrões; que a Biologia Computacional, ao fazer uso de abstrações e métodos computacionais, está mudando a forma dos Biólogos pensarem; De como a teoria computacional dos jogos tem mudado a forma dos economistas pensarem. Que a nanocomputação tem auxiliado pesquisa da química e computação quântica, as da física. Desta forma, o Pensamento Computacional “envolve resolver problemas, projetar sistemas e entender o comportamento humano, baseando-se nos conceitos fundamentais da Ciência da Computação” (WING, 2006). Em suma, a autora enfatiza que as dimensões do PC devem ser adicionadas à capacidade analítica de todos, independentemente de suas áreas.

A fim de delimitar melhor as fronteiras do Pensamento Computacional, Wing agrupa estas competências sob o tópico “O que é e não é [Pensamento Computacional]”, caracterizando-as em seis premissas:

- É conceituar ao invés de programar
- É uma habilidade fundamental, não mecânica
- É a maneira que os humanos pensam, não como computadores pensam
- Complementa e combina o pensamento matemático e de engenharia
- Gera ideias, não artefatos
- Para todos e em todos os lugares
- Explicando detalhadamente cada uma, temos que:

É conceituar ao invés de programar: A Ciência da Computação não é programação de computadores. Aplicar o Pensamento Computacional para resolver um problema exige pensar em vários níveis de abstração, visando reduzir problemas grandes e aparentemente insolucionáveis em problemas menores e mais simples de resolver. Não é o mero exercício de codificar um programa.

É uma habilidade fundamental, não mecânica: O Pensamento Computacional é uma forma de racionalizar, de pensar de forma estruturada e ilimitada, não uma mera execução de uma rotina. É uma habilidade de utilização de recursos ubíquos da sociedade – computadores - para resolver problemas da sociedade e por isso deveriam ser desenvolvidos por todas as pessoas.

É a maneira que os humanos pensam, não como computadores pensam: O Pensamento Computacional é a forma das pessoas pensarem, de forma inteligente e imaginativa, não da forma dos computadores pensarem de forma limitada e previsível. Neste sentido, os Computadores são meios, ferramentas pelas quais os humanos conseguem executar ações pensadas previamente para resolver problemas. Os computadores só são o que são, possuem as funções e poderes que tem hoje, porque nós os humanos pensamos nesses poderes e os criamos a fim de que possamos enfrentar problemas que não ousávamos enfrentar antes da computação.

Complementa e combina o pensamento matemático e de engenharia: A Ciência da Computação baseia-se no pensamento matemático, uma vez que tem seus fundamentos formais nela, e na engenharia, visto que os sistemas computacionais são criados para interagirem com o mundo real. Para que possamos ultrapassar as limitações dos dispositivos computacionais, os cientistas da computação necessitam pensar computacionalmente, isto é, construir mundos e cenários virtuais e simulados que vão muito além do mundo físico. Em certo sentido, a computação consegue ser razoável quando aplicado a experimentos físicos e metafísicos.

Gera ideias, não artefatos: Não são somente artefatos como softwares e hardwares que estarão presentes em todos os lugares impactando nossas vidas e sim conceitos computacionais que são empregados para abordar e resolver problemas em diferentes conceitos. Arpanet como artefato serviu para a ideia inicial que fazer a comunicação quebrar a barreira do espaço, mas foi a ideia de comunicação a distância e troca de informações em massa que motivou a continuação de estudos e estratégias para transformar a Arpanet como internet que temos hoje em dia, o qual permite-nos nos comunicar das mais diferentes formas e dos lugares mais longínquos.

Para todos e em todos os lugares: Por fim, o Pensamento Computacional será tão integrado aos processos de execução, realização, efetivação de projetos humanos que deixará de ser uma filosofia explícita.

Em um artigo publicado em 2007, Wing juntamente com outros pesquisadores afirmam que todas as pessoas pensam computacionalmente, mesmo que de forma inconsciente. Para eles, o Pensamento Computacional

“Representa um amplo espectro de raciocínio ao longo do tempo e das disciplinas. Aprender a contar é um começo do pensamento computacional humano, seguido naturalmente pela computação aritmética e níveis abstratos de pensamento baseado em símbolos, muitas vezes começando com a álgebra. Contagem, aritmética, símbolos e pensamento abstrato são fundamentais para o estudo da computação. O raciocínio computacional é o núcleo de todas as disciplinas modernas de Ciência, Tecnologia, Engenharia e Matemática (CTEM) e é intrínseco a todas as outras disciplinas de A a Z. Ele é usado em nossas vidas cotidianas desde assar um bolo, trocar um pneu ou escovar os dentes. O cérebro humano está programado para pensar computacionalmente, assim como os dispositivos de computação modernos (HENDERSON et al., 2007, p. 195).

Após esta tentativa de reafirmar o Pensamento Computacional como uma forma racional de resolver problemas, outros diversos autores se propuseram a definir o Pensamento Computacional. A própria precursora do termo, Jeannette Wing propôs diferentes definições para o termo. Em uma de suas definições, Wing (2008, p. 3717) afirma que “o pensamento computacional é um tipo de pensamento analítico” e que sua essência é a abstração. Pode-se entender por processo de abstração, a capacidade de “decidir quais detalhes precisamos destacar e quais detalhes que podemos ignorar – por trás do pensamento computacional” (WING, 2008, p. 3718). Para Wing, “na computação, abstraímos noções além das dimensões físicas de tempo e espaço” (WING, 2008, p. 3717). É nesse contexto em que Wing sustenta a ideia de que o Pensamento Computacional nos permite criar Algoritmos para resolver problemas. “Um algoritmo é uma abstração de um procedimento passo a passo para obter entradas e produzir alguma saída desejada” (WING, 2008, p. 3718).

Em seu artigo intitulado “*What and Why?*”, Wing (2011, p. 1), define o Pensamento Computacional como “o processo de pensamento envolvido na formulação de problemas e suas soluções para que as soluções sejam representadas de uma forma que possa ser efetivamente executada por um agente de processamento de informações”. Em outras palavras, o Pensamento Computacional descreve o que pode ser considerada uma atividade mental para formulação de um problema que por sua vez pode admitir uma solução computacional. Tal solução pode ser realizada por um ser humano, uma máquina, ou uma combinação de ambos (Wing, 2011).

Em uma palestra realizada na Columbia Journalism School , Wing (2014) afirma que o ato de pensar como cientista refere-se ao ato de automatizar a abstração, isto é, podemos abstrair dados do mundo real e automatizar o processo para processá-los de forma a retornar uma solução útil. Para a autora esse conceito é tão verdadeiro que é possível notar a automação da abstração na Ciência da Computação. Para isso, Wing apresenta classes de amostras de abstrações computacionais, conforme disposto no quadro abaixo:

CLASSES DE AMOSTRAS DE ABSTRAÇÕES COMPUTACIONAIS	
Classe	Amostras de Abstrações Computacionais
Algoritmos	MergeSorte, busca binária, correspondência de string, agrupamento
Estruturas de dados	Sequências, tabelas, árvores, gráficos, rede
Máquinas de estado	Autômatos finitos, Máquina de Turing
Línguas	Expressões regulares, VDM, ML, Haskell, Java, Python
Lógicas e semântica	Triplos de Hoare, lógica temporal, lógica modal, cálculo lambda
Heurísticas	A*(best-first graph search), cache
Estruturas de controle	Paralelismo/composição sequencial, iteração, recursão
Comunicação	Síncrono/assíncrono, transmissão/p2p, RPC, memória compartilhada, passagem de mensagem
Arquitetura	Camadas, pipeline, blackboard, loop de feedback, cliente-servidor, paralelo, distribuído, tolerante a falha

Quadro 1. Classes de Amostras de Abstrações Computacionais.

Fonte. Adaptado de Wing (2014). Próprio autor.

Segundo Wing (2014), isto vai além de usar o computador, de escrever um artigo no Word, de acessar a internet pelo Google Chrome. Transcende a programação de computadores. São soluções advindas de inquietações de como resolver problemas, que foram resolvidas por meio do pensar computacionalmente.

Para Bundy (2007) e Bliksstein (2008), o Pensamento Computacional é saber usar o computador como ferramenta, como instrumento para aumentar o poder cognitivo e operacional humano, aumentando nossa produtividade, nossa inventividade e criatividade.

McMaster et al. (2010) argumenta que ao longo da linha do tempo da Ciência da Computação, muitos autores têm defendido que esta ciência deveria dar mais ênfase à teoria e a problemas de design do que na programação em si (HENDERSON, 2001; WING, 2006; KRAMER, 2007). Nessa perspectiva, o Pensamento Computacional é reconhecido então como a essência da Ciência da Computação, isto é, uma de suas práticas fundamentais.

Para Nunes (2011), o Pensamento Computacional é como um “processo cognitivo utilizado pelos seres humanos para encontrar algoritmos para resolver problemas”, falando de outra forma, ao

desenvolver o Pensamento Computacional, o homem é capacitado para sistematizar e organizar a solução de problemas, ou seja, resolver de forma algorítmica (passos estruturados passíveis de reaplicação).

A Royal Society (2012), também ofereceu uma definição sucinta que captura com maestria a essência do Pensamento Computacional – “Pensamento computacional é o processo de reconhecer aspectos da computação no mundo que nos cerca e aplicar ferramentas e técnicas da Ciência da Computação para entender e raciocinar sobre sistemas e processos naturais e artificiais” (p. 29).

Segundo Liukas (2015, apud BRACKMANN, 2017, p. 28), o Pensamento Computacional é “pensar nos problemas de forma que um computador consiga solucioná-los”. Além disso, Liukas defende que o Pensamento Computacional “é executado por pessoas e não por computadores. Ele inclui o pensamento lógico, a habilidade de reconhecimento de padrões, raciocinar através de algoritmos, decompor e abstrair um problema”.

No ano de 2011, a *International Society for Technology in Education (ISTE)* e a *Computer Science Teachers Association (CSTA)* colaboraram com líderes do ensino superior, indústrias e educadores para desenvolver uma definição operacional para o Pensamento Computacional - PC. Segundo a CSTA (2011, p. 7)

- *O PC é um processo de resolução de problemas que inclui (mas não se limita a) as seguintes características:*
- *Formular problemas de uma forma que nos permita usar um computador e outras ferramentas para ajudar a resolvê-los;*
- *Organizar e analisar dados logicamente;*
- *Representar dados por meio de abstrações, como modelos e simulações;*
- *Automatizar soluções através do pensamento algorítmico (uma série de etapas ordenadas);*
- *Identificar, analisar e implementar possíveis soluções com o objetivo de alcançar a combinação mais eficiente e eficaz de etapas e recursos;*
- *Generalizar e transferir este processo de resolução de problemas para uma ampla variedade de problemas.*
- *Essas habilidades são apoiadas e aprimoradas por uma série de disposições ou atitudes que são dimensões essenciais da PC. Essas disposições ou atitudes incluem:*
- *Confiança para lidar com a complexidade*
- *Persistência em trabalhar com problemas difíceis*
- *Tolerância para ambiguidade*

- *A capacidade de lidar com problemas em aberto*
- *A capacidade de se comunicar e trabalhar com outras pessoas para alcançar um objetivo ou solução comum*

Em 2016, o *College Board* e a *National Science Foundation - NSF* lançaram um curso visando decompor o Pensamento Computacional em ideias a serem incorporados nos currículos do Ensino Médio, O curso Princípios do Pensamento Computacional tem como foco as práticas do pensamento computacional, baseando-se em sete “grandes ideias” da computação:

- *A computação é uma atividade humana criativa*
- *A abstração reduz informações e detalhes para focar em conceitos relevantes para entender e resolver problemas*
- *Dados e informações facilitam a criação de conhecimento*
- *Algoritmos são ferramentas para desenvolver e expressar soluções para problemas computacionais*
- *A programação é um processo criativo que produz artefatos computacionais*
- *Dispositivos digitais, sistemas e redes que os interligam permitem e promovem abordagens computacionais para resolver problemas*
- *A computação permite a inovação em outros campos, incluindo ciência, ciências sociais, humanidades, artes, medicina, engenharia e negócios (GROVE; PEA, 2013, p. 39).*

Por fim, Grove e Pea (2013), após a realização de uma revisão bibliográfica, apontaram nove elementos que são amplamente aceitos como componentes do Pensamento Computacional e formam a base dos currículos que visam apoiar a aprendizagem, bem como avaliar seu desenvolvimento:

- *Abstrações e generalizações de padrões (incluindo modelos e simulações)*
- *Processamento sistemático de informações*
- *Sistemas de símbolos e representações*
- *Noções algorítmicas de fluxo de controle*
- *Decomposição estruturada do problema (modularização)*
- *Pensamento iterativo, recursivo e paralelo*
- *Lógica condicional*
- *Restrições de eficiência e desempenho*
- *Depuração e detecção sistemática de erros*

É certo que não há uma definição abraçada por todos os teóricos defensores do Pensamento Computacional, bem como assume-se que ainda é desafiador definir o que de fato pode e não pode ser classificado como atividades derivadas do Pensamento Computacional, mas ao contrário do que

afirma BRACKMANN (2017), é possível sim definir claramente as fronteiras, os limites e intersecções entre o Pensamento Computacional e a Computação a partir de dois autores centrais: Wing e Denning..

Ora, o homem já pensava computacionalmente (fazia uso do Pensamento Computacional) quando era capaz de contar e em seguida, computar de forma aritmética e com níveis abstratos de pensamento baseado em símbolos, tais como (+, -, /, *), e por meio deles realizava cálculos simples como soma, subtração e divisão (HENDERSON et al., 2007). Ele também o fazia quando pensava de forma analítica e criava estruturas mentais estruturadas para resolver problemas (WING, 2006), o que conhecemos hoje como algoritmos. Desta forma, entende-se que o Pensamento Computacional se delimita aos processos de pensamento envolvidos na formulação de problemas, no qual o indivíduo reconhece aspectos da computação no mundo ao seu redor, abstrair informações e as estrutura para que “suas soluções possam ser representadas como etapas e algoritmos computacionais” (AHO, 2012, p. 832).

A Ciência da Computação, por sua vez, dá base para a representação dessas soluções previamente pensadas em etapas e algoritmos e as representa nos sistemas computacionais. Entende-se, portanto, que os sistemas computacionais não resolvem problemas. Pessoas resolvem problemas e usam as ferramentas computacionais oriundas da Ciência da Computação para expandir a capacidade do homem de computar e processar as informações (que outrora já fazia por meio do pensamento) de forma mais complexa por meio de algoritmos pré estabelecidos, a fim de solucionar um problema. O Pensamento Computacional precede a Ciência da computação pois primeiro é necessário pensarmos nos problemas de formas abstrata, transformar as incógnitas do problema variáveis computáveis para que só depois um computador consiga solucioná-los (Liukas, 2015, apud BRACKMANN, 2017, p. 28). Portanto, a “ciência da computação estuda os processos de informação tanto artificiais quanto naturais” (DENNING, 2005, p. 28).

Para exemplificar melhor o relacionamento entre as áreas, este pesquisador propõe um ajuste ao esquema de relacionamento entre as áreas apresentado por BRACKMANN (2017, p. 30), conforme Figura 2. A proposta inclui as áreas da Programação e Codificação, endossando o que o autor apresentou, com singelas adaptações.

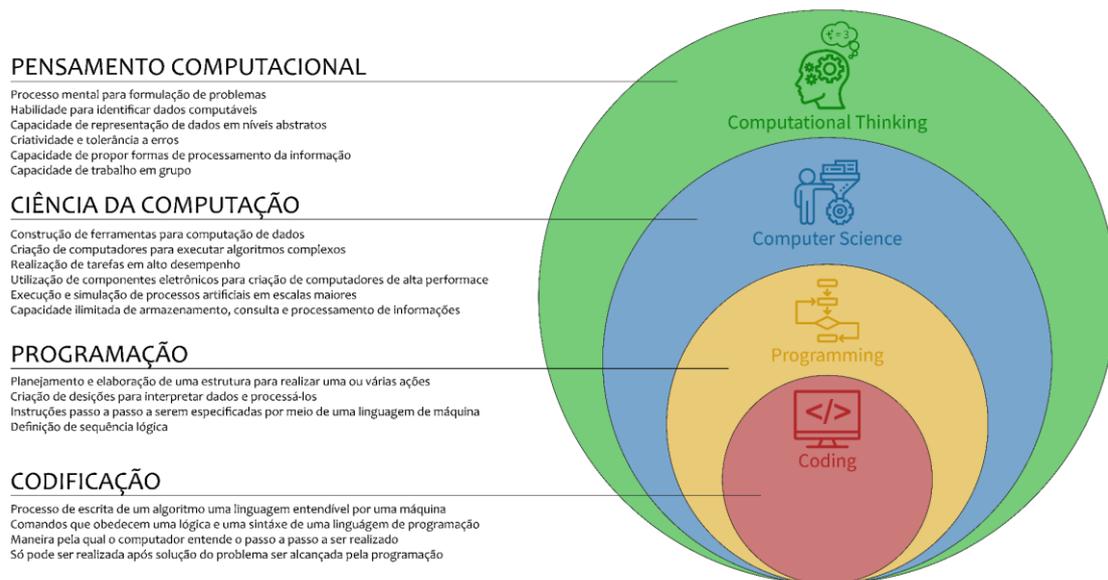


Figura 2. Novo esquema de relacionamento entre as áreas e seus contextos (adaptado de BRACKMANN (2017)).
 Fonte: O autor (2024).

Portanto este autor declara sua própria definição, a partir das supracitadas definições, afirmando que o Pensamento Computacional é: Um processo cognitivo capaz de sintetizar informações do mundo real a partir da abstração e as organizar em formato de dados computáveis, para serem processados por um sistema computacional ou não, afim de gerar uma solução útil para um problema.

2.2.1 Dimensões do Pensamento Computacional

Como foi apresentado anteriormente, diversos autores inferiram esforços para definir o Pensamento Computacional e delimitar o seu escopo. No entanto, as habilidades e dimensões que constituem o Pensamento Computacional e que devem ser desenvolvidas, ainda não são um consenso na literatura, ficando a cargo de cada estudo escolher a abordagem teórica que irá empregar (GUARDA & PINTO, 2020). Sendo assim, abaixo são apresentados dois modelos teóricos que chamaremos de Dimensões do Pensamento Computacional.

2.2.1.1 Dimensões do Pensamento Computacional conforme Brackmann (2017)

Christian Brackmann em sua tese de doutorado argumenta que Pensamento Computacional está estruturado sobre Quatro dimensões, o que ele chama de Quatro Pilares do Pensamento Computacional: Decomposição, Reconhecimento de Padrões, Abstração e Algoritmos (Figura 3).

Esses pilares ou dimensões são habilidades importantes empregadas de forma independente para atingir o objetivo principal dessa abordagem: solução de problemas computacionalmente viáveis (BRACKMANN, 2017).

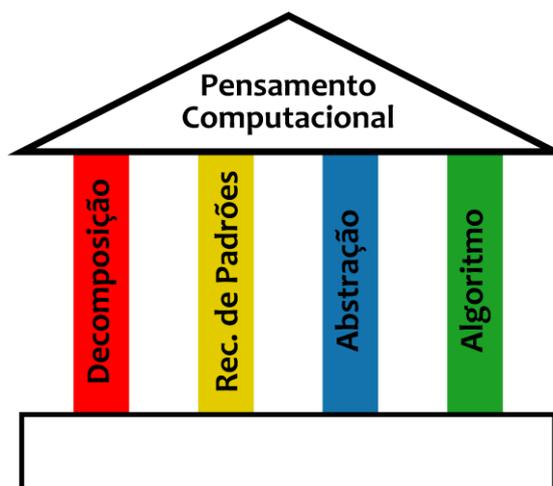


Figura 3. Os Quatro Pilares do Pensamento Computacional.
Fonte: Adaptado de BRACKMANN (2017, p. 33).

A Decomposição é o primeiro pilar e consiste no processo de quebrar um problema grande em partes maiores (Liukas, 2015). Trata-se de quebrar, particionar ou fragmentar um problema ou sistema complexo em partes menores, que são mais fáceis de entender e podem ser examinadas e resolvidas individualmente formando um todo. Brackmann ressalta ainda que quando um problema não é decomposto, sua resolução é muito mais difícil e complexa, por conta da dificuldade de gerir muitos estágios diferentes ao mesmo tempo. Desta forma, a divisão do problema em pequenas partes possibilita a resolução individual e facilitada de cada uma dessas partes como se fossem independentes.

O autor propõe que olhemos para o esquema de uma bicicleta decomposta em 3 partes: quadro, rodas e correia (Figura 4) para facilitar a compreensão de como ela funciona. Quando o esquema da bicicleta é quebrado em partes menores, fica mais fácil entender como funcionam seus mecanismos de estrutura, a tração e a transferência de energia.



Figura 4. Partes de uma bicicleta (Decomposição)
Fonte: BRACKMANN (2017, p. 35).

Brackmann, ao citar Liukas (2015, p. 35), “afirma que programadores utilizam frequentemente esta técnica para dividir um algoritmo em pedaços menores para facilitar sua compreensão e manutenção”. O autor exemplifica a Decomposição por meio de um código-fonte de um programa, que pode ser decomposto através de funções, procedimentos, objetos, módulos, entre outros. Com essa técnica é possível resolver problemas complexos de forma simples, compreender novas situações e projetar grande sistemas de mais facilmente (BRACKMANN, 2017, p. 35 apud CAS, 2014; CSIMADIA et al., 2015)

O segundo pilar é o Reconhecimento de Padrões. A literatura também associa o termo Reconhecimento de Padrões com “Generalização”. Esta dimensão consiste em “encontrar padrões entre os subproblemas gerados” (BRACKMANN, 2017, p. 35). Após realizar a decomposição de um problema complexo é possível encontrar similaridade ou características que alguns dos problemas compartilham. Desta forma, ao reconhecer um problema ou um padrão entre as partes de um problema decomposto, é possível aplicar soluções que já foram empregadas para resolver problemas semelhantes em outras situações e aumentando a eficiência da resolução.

O Reconhecimento de Padrões é uma forma de resolver problemas rapidamente fazendo uso de soluções previamente definidas em outros problemas e com base em experiências anteriores. Os questionamentos “Esse problema é similar a um outro problema que já tenha resolvido?” ou “Como ele é diferente?” são importantes nesta etapa, pois ocorre a definição dos dados, processos e estratégias que serão utilizados para resolver o problema. Algoritmos que são responsáveis pela solução de algum problema específico podem ser adaptados para resolver uma variedade de

problemas similares. Sempre que necessário, o algoritmo pode aplicar uma solução de forma generalizada (BRACKMANN, 2017, p. 35).

Para exemplificar melhor o Reconhecimento de Padrões, Brackmann aplica esta técnica através da identificação de similaridade entre raças de cães, conforme a Figura 5.



Figura 5. Similaridade entre raças de cachorros (Reconhecimento de Padrões)
Fonte: BRACKMANN (2017, p. 36).

Podemos reconhecer padrões nos cães, como por exemplo, todos possuem olhos, rabos e pelos, apesar de suas características serem diferentes. A partir do momento que conseguimos reconhecer padrões dos cães, podemos descrever outros, simplesmente seguindo o padrão e alterando algumas características, por exemplo, cão de olhos azuis, rabo curto e pelagem cinza; outro com olhos pretos, rabo longo e pelos ruivos.

Para Brackmann, o Reconhecimento de Padrões possibilita simplificar uma solução de um problema e transformá-la em uma solução genérica, de forma a replicá-la em outros cenários semelhantes. Ainda usando o exemplo dos cachorros, o autor propõe que se quisermos desenhar diversos cachorros diferentes, poderíamos montar um padrão genérico e replicá-lo para construir variações replicadas (Figura 6). Sem o Reconhecimento de Padrões, teríamos que retornar a etapa de análise das características de um cachorro toda vez que fosse necessário criar uma variação, o que seria ineficiente, principalmente em situações de resolução de problemas mais complexos. A partir do padrão, podemos resolver o problema de desenhar as variações de cachorros alterando somente algumas características.

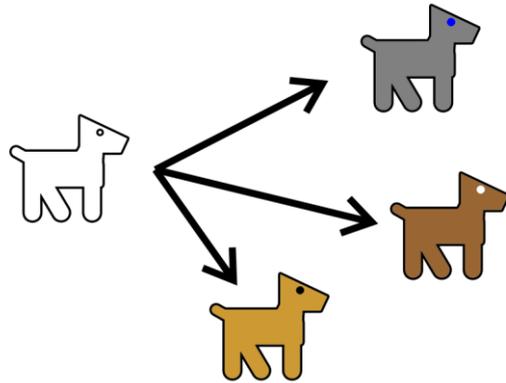


Figura 6. Padrão e replicações
Fonte: BRACKMANN (2017, p. 36).

O terceiro pilar proposto por Brackmann é a Abstração. A abstração envolve o processo de filtragem de dados e sua classificação, dentre um universo muito maior de informações. Neste pilar é fundamental se concentrar nos elementos necessários e ignorar (abstrair) os que não são relevantes. Desta forma é possível, criar uma representação (ideia) do que está se tentando resolver (BRACKMANN, 2017, p. 38).

A habilidade essencial desempenhada neste pilar é escolher o detalhe a ser ignorado para que o problema seja mais fácil de ser compreendido sem haver perda de nenhuma informação importante para tal (CAS, 2014).

Jeannette Wing (2006) afirma que a abstração é o conceito mais importante do Pensamento Computacional, visto que o processo de abstrair é utilizado em diversos momentos, tais como:

- Na escrita do algoritmo e suas iterações;
- Na seleção dos dados importantes;
- Na escrita de uma pergunta;
- Na alteridade de um indivíduo em relação a um robô;
- Na compreensão e organização de módulos em um sistema.



Figura 7. O mapa do metrô como exemplo de abstração do mundo real.
 Fonte: BRACKMANN (2017, p. 39).

O quarto e último pilar é o Algoritmo. Wing (2014) considera este pilar agrega todos os outros pilares. É uma estratégia, um conjunto de instruções claras e não ambíguas para a solução de um problema (CSIZMADIA et al., 2015). Para Brackmann (2017), em um algoritmo, as instruções são descritas e ordenadas para atingir um objetivo e podem ser escritas em forma narrativa, fluxogramas ou pseudocódigo (linguagem humana), para depois serem escritas em linguagem de programação. Para Liukas (2015), os Algoritmos são um conjunto de passos bem definidos usados para solucionar um problema. O termo se diferencia do termo “Programa”, este segundo se refere a uma sequência de instruções dadas ao computador por meio de uma linguagem que estes compreendem.

Os Algoritmos são as soluções prontas do problema, visto que para alcançá-lo, foi necessário primeiro decompor o problema, abstrair as partes principais e reconhecer os padrões para formular tal solução algorítmica que ao serem executadas, seguirão um passo a passo pré-determinado, isso significa que a solução pode ser aplicada quantas vezes forem necessárias, sem a necessidade de se criar um novo algoritmo para cada uma de suas execuções posteriores. Por isto é correto afirmar que

“a principal característica do algoritmo é a possibilidade de automação das soluções” (Brackmann, 2017, p.41).

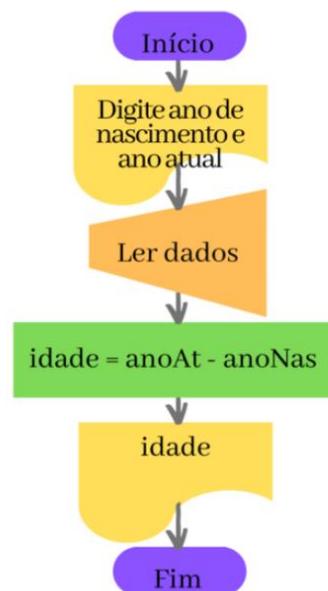


Figura 8. Fluxograma de um algoritmo para descobrir a idade de uma pessoa.
Fonte: Gaspar (2021).

Em suma, as Quatro Dimensões ou Pilares do Pensamento Computacional proposto por Brackmann se interligam da seguinte maneira:

O Pensamento Computacional envolve identificar um problema complexo e quebrá-lo em pedaços menores e mais fáceis de gerenciar (Decomposição). Cada um desses problemas menores pode ser analisado individualmente com maior profundidade, identificando problemas parecidos que já foram solucionados anteriormente (Reconhecimento de padrões), focando apenas nos detalhes que são importantes, enquanto informações irrelevantes são ignoradas (Abstração). Por último, passos ou regras simples podem ser criados para resolver cada um dos subproblemas encontrados (Algoritmos). Seguindo os passos ou regras utilizadas para criar um código, é possível também ser compreendido por sistemas computacionais e, conseqüentemente, utilizado na resolução de problemas complexos de forma eficiente, independentemente da carreira profissional que o estudante deseja seguir (BRACKMANN, et al., 2017b, p. 983).

2.2.1.2 Dimensões do Pensamento Computacional conforme Guarda e Pinto (2020)

Em um estudo bem recente, Guarda e Pinto (2020) realizaram uma revisão sistemática com o objetivo de “mapear diferentes visões conceituais e quais são as dimensões acerca do PC para servir

como referência teórica para formação de novas experiências e abordagens empregando o PC” (p. 1464). Os 19 artigos conservados após a aplicação dos procedimentos metodológicos, foram categorizados para análise futura “com base no referencial teórico, nas definições e nas dimensões do PC dispostos em cada artigo” (GUARDA & PINTO, 2020, p. 1467). Como resultado da análise, os autores sugerem 6 agrupamentos de artigos, onde cada agrupamento destaca suas dimensões do Pensamento Computacional, conforme disposto na Tabela 2.

O primeiro agrupamento apresentado por Guarda e Pinto (2020) é derivado dos artigos de Wing (2006). Este por sua vez destaca as seguintes dimensões do PC: abstração, decomposição de problemas, reformulação de problemas, automação e testes sistemáticos. Os autores ressaltam ainda que este artigo de Wing, originou uma nova onda de estudos acerca das dimensões do Pensamento Computacional.

O segundo agrupamento de artigos, também derivado de Wing, começou quando Barr e Stephenson (2011) comparam “conceitos e capacidades do PC em ciência da computação, matemática, ciências, artes da linguagem e estudos sociais”. Desta forma, nove dimensões do Pensamento Computacional foram identificadas, três estão ligados aos conceitos de manipulação de dados: coleta, análise e representação de dados; e seis, aos conceitos de solução de problemas: decomposição, abstração, algoritmos e procedimentos, automação, paralelização e simulação (GUARDA & PINTO, 2020, p. 1467).

O terceiro agrupamento de artigos, também derivado de Wing, surgiu quando CSTA e ISTE (2011) usaram as premissas de Wing para propor seis conceitos que descreviam o Pensamento Computacional: formular problemas, organizar e analisar dados, abstrações, automação através do pensamento algorítmico, avaliação de eficiência e correção e generalização.

O quarto agrupamento, derivado de Wing, surgiu com Brennan e Resnick (2012), ao descreverem quatro práticas para avaliar projetos de Pensamento Computacional: abstraindo e modularizando, reutilizando e remixando, sendo incremental e iterativo, e testando e depurando

O quinto agrupamento de artigos, derivado de Barr e Stephenson 2011 e CSTA e ISTE 2011, foi formado pelas ideias de Selby e Woollard 2013. Estes identificaram os termos mais associados ao Pensamento Computacional e propuseram uma definição das dimensões do Pensamento Computacional incluindo os termos: abstrações, decomposição, pensamento algorítmico, generalização e avaliação.

O sexto agrupamento é derivado de Brennan e Resnick 2012. É baseado na conexão de dimensões do Pensamento Computacional com análise automática de projeto. Moreno-León (2015) avalia os seguintes aspectos do Pensamento Computacional na avaliação de projetos Scratch: abstração na criação de funções e clones, paralelismo no início de vários processos ao mesmo tempo, lógica no uso de operações lógicas, sincronização no envio de mensagens, controle de fluxo no criando loops razoáveis, interatividade do usuário no uso da interação e representação de dados no uso de variáveis e listas em programas (GUARDA & PINTO, 2020, p. 1468).

Com base nos 6 supracitados agrupamentos, os autores puderam reagrupar as habilidades do Pensamento Computacional em três estágios maiores, gerando assim um novo modelo para o desenvolvimento de habilidades do PC abrangendo todas as principais dimensões extraídas dos artigos em um ciclo de solução de problemas em três etapas: 1) Definição do problema, 2) Resolução do problema e 3) Análise da solução, conforme Quadro 2 (GUARDA & PINTO, 2020, p. 1469).

	Novo Modelo:	[Wing 2006]:	[Barr e Stephenson 2011]:	[CSTA e ISTE 2011]:	[Brennan e Resnick 2012]:	[Selby e Woollard 2013]:	[Moreno-León 2015]:
Definição do problema	Formulação do problema	Formulação do problema	-	Formulação do problema	-	-	-
	Abstração	Abstração	Abstração	Abstração	Abstração e modularização	Abstração	Abstração
	Reformulação do problema	Reformulação do problema	-	-	-	-	-
	Decomposição	Problema decomposto	Problema decomposto	-	-	Decomposição	Problema decomposto
Resolução do problema	Coleta e análise de dados	-	Coleta e representação de dados	Organização e análise de dados	Reutilização e remixação de dados	-	Representação de dados
			Simulação				
	Pensamento algorítmico	-	Algoritmos e Procedimentos	Pensamento algorítmico	-	Pensamento algorítmico	Pensamento lógico
	Paralelização e iterações	-	Paralelização	-	Iterativo e incremental	-	Paralelização e iterações
	Automação	Automação	Automação	Automação	-	-	Sincronização
Análise da solução	Generalização	-	-	Generalização	-	Generalização	-
	Testes	Testes sistemáticos	-	-	Teste e depuração	-	-
	Avaliação	-	-	Avaliação e correção	-	Avaliação	-

Quadro 2. Organização dos agrupamentos das dimensões do Pensamento Computacional.
Fonte: Guarda e Pinto (2020, p. 1468).

Guarda e Pinto (2020) concluem o estudo então, definindo o ciclo de solução de problemas em três etapas: Definição do problema, Resolução do Problema e Análise da solução.

A primeira etapa, Definição do Problema, refere-se a todas as habilidades do Pensamento Computacional empregadas antes de começar a resolver o problema. Fazem parte desta etapa as seguintes habilidades:

- Formulação do problema
- Abstração
- Reformulação do problema
- Decomposição

A solução do problema começa na formulação e delimitação dele. Já que o Pensamento Computacional é um processo mental usado para resolver problemas de acordo com algoritmos, os autores ressaltam que “é essencial entender e pesquisar o problema que precisa ser resolvido” (GUARDA & PINTO, 2020, p. 1469). Após isso, usa-se a abstração para identificar e extrair informações relevantes para definir as ideias principais e modelar os aspectos principais do problema ou do sistema complexo sobre o qual está se debruçando (modularização). A Reformulação do Problema, por sua vez, pode ser empregada para reformular o problema de forma mais solucionável e familiar. Por fim, nesta primeira etapa, a Decomposição, normalmente listada após a Abstração, é proposta pelos autores após a Reformulação de problemas e consiste em dividir os problemas em unidades menores e gerenciáveis. Note que nessa primeira etapa do ciclo de soluções de problemas de Guarda e Pinto (2020) já podemos notar 3 dos pilares do Pensamento Computacional proposto por Brackmann (2017): Abstração, Reconhecimento de Padrões (Reformulação do Problema) e Decomposição.

A segunda etapa do Modelo de Guarda e Pinto (2020), Resolução do Problema, refere-se a todas as habilidades do Pensamento Computacional empregadas para criar soluções para o problema. Fazem parte desta etapa as seguintes habilidades:

- Coleta de dados
- Pensamento Algoritmo
- Paralelização e iteração
- Automação

Para resolver um problema algorítmicamente, os autores defendem que é preciso primeiro coletar e analisar dados. Após isto, é possível seguir com o design algoritmo ou pensamento algorítmico para estruturar uma série de etapas ordenadas usando a paralelização e iteração, para alcançar uma solução do problema automatizada, seja por meios computacionais ou não. Nota-se o

emprego do último pilar do Pensamento Computacional proposto por Brackmann (2017), o Algoritmo, nessa segunda etapa do Modelo de Guarda e Pinto (2020).

Fazem parte da terceira etapa do Modelo de Guarda e Pinto (2020), Análise de Solução, as seguintes habilidades:

- Generalização
- Testes
- Avaliação

Estas habilidades consistem em analisar a solução alcançada na etapa anterior de forma a aplicar o processo de solução do problema em outros diversos cenários, isto é, generalizar o processo de solução. Por fim, no processo de avaliação e teste, é possível analisar os processos e resultados sob a ótica da eficiência e utilização de recursos. Após a conclusão de todas as etapas do Pensamento Computacional, “a solução poderá ser aprimorada após a avaliação e o teste, formulando o problema novamente. Isso significa repetir a aplicação em três etapas até que o usuário esteja satisfeito com o resultado”(GUARDA & PINTO, 2020, p. 1469) . O novo modelo do Pensamento Computacional é cíclico e interativo, conforme mostra a Figura 9.

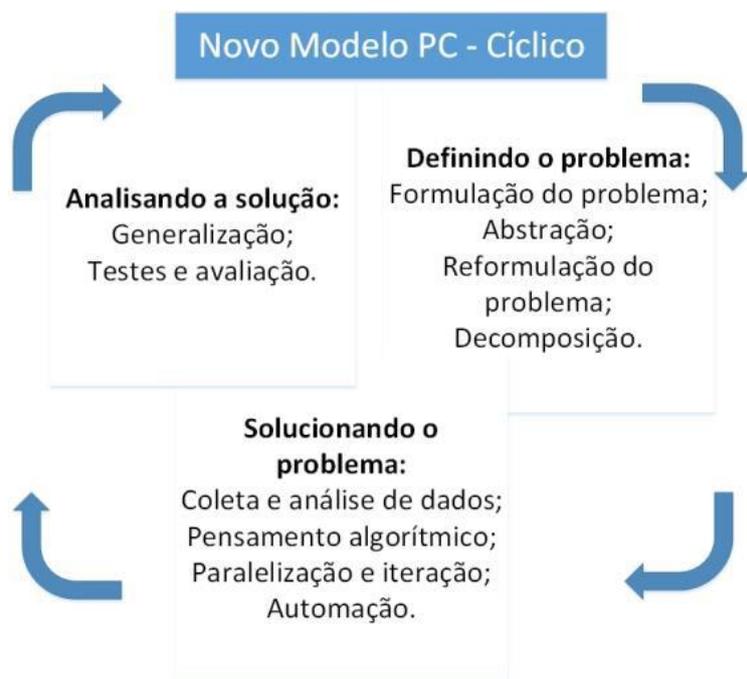


Figura 9. Novo modelo para o PC
Fonte: GUARDA & PINTO (2020, p. 1470).

Este autor conduzirá esta pesquisa utilizando o modelo teórico proposto por Guarda e Pinto (2020) ao reconhecer que tal modelo melhor delimita as dimensões do Pensamento Computacional. Suas etapas reúnem diversas habilidades e competências propostas no decorrer da história moderna do Pensamento Computacional, culminando em processo mental ágil e interativo, características fundamentais nos processos de Design.

2.2.2 O Pensamento Computacional na Educação brasileira

A Educação Brasileira é regulamentada a partir da Constituição Federal de 1988, na Lei de Diretrizes e Bases da Educação e no Plano Nacional de Educação, o qual propõe metas para serem alcançadas na próxima década. Estas leis asseguram o direito à educação, garantindo um ensino de qualidade, centrado nas aprendizagens e no desenvolvimento integral do aluno. Para complementar a legislação, foi elaborada a Base Nacional Comum Curricular - BNCC, documento democrático, coordenado pelo Ministério da Educação, com colaboração da sociedade civil, professores e gestores, que define os conhecimentos fundamentais que todos os alunos da Educação Básica têm direito a aprender.

A BNCC apresenta 10 competências gerais, sendo a quinta voltada para a Cultura Digital.

5. Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva. (BNCC, 2017)

Nessa competência há um engajamento para satisfazer a necessidade de uso das tecnologias na educação, não necessariamente abordando o Pensamento Computacional como primordial no processo de ensino e aprendizagem. A Cultura Digital compreende a capacidade dos estudantes de usar a tecnologia de forma crítica, com ética, para obter informação e para produzi-la. Segundo Seibert (2019), ela dialoga com diversos componentes curriculares, desenvolvendo a capacidade de utilizar as ferramentas digitais, isto é, o computador, softwares, as tecnologias para pesquisar informações, de apresentar o resultado de um trabalho, dentre outras possibilidades.

Embora o objetivo deste estudo não seja tratar a respeito do uso da Tecnologias na Educação, é válido mencionar que as contribuições da inserção das Tecnologias Digitais de Informação e Comunicação - TDIC foram importantes para que chegássemos às discussões sobre o ensino e aprendizagem de Algoritmos e Programação no Ensino Fundamental 1: Séries Iniciais, como parte do desenvolvimento do Pensamento Computacional em crianças. As contribuições das TDICs na educação são cada vez mais reconhecidas por boa parte dos docentes e discentes. “Esse

reconhecimento, todavia, deu-se de forma gradativa, como resultado de um processo histórico, cultural, social e político, vinculado ao desenvolvimento das tecnologias, de modo que no decorrer do processo de inserção das TDIC na Educação, várias abordagens foram experimentadas” (KAMINSKI; KLÜBER; BOSCARIOLI, 2021, p. 605). Os autores representam esse processo histórico, cultura, social e política através da linha do tempo a seguir.

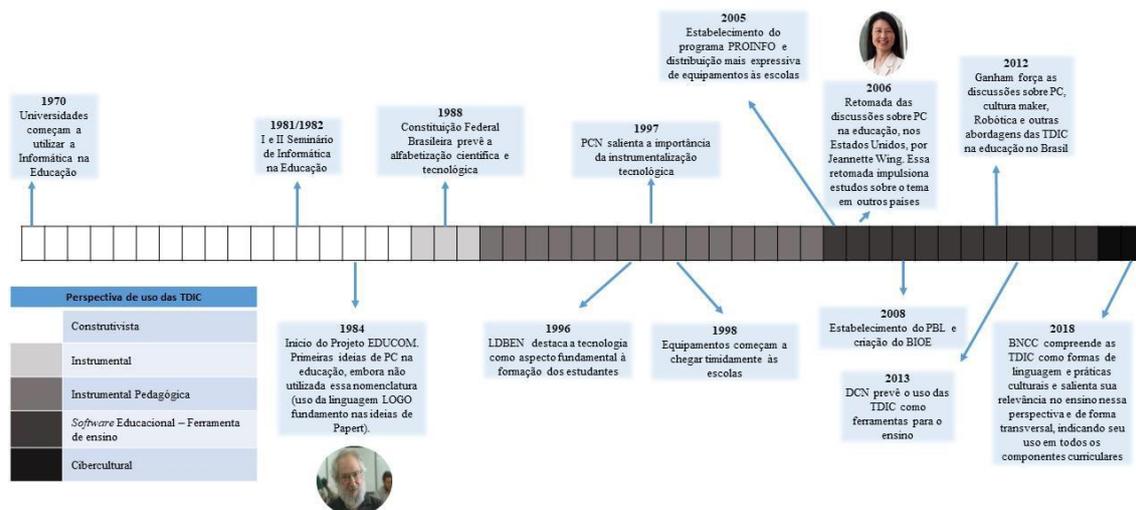


Figura 10. Marcos da trajetória da Informática na Educação Brasileira. Fonte: (KAMINSKI; KLÜBER; BOSCARIOLI, 2021, p. 619).

As diferentes maneiras de abordar a inserção das TDIC na educação são marcadas por variadas compreensões da relação das Tecnologias com o Homem. Desta forma, as ações profissionais que fazem uso das TDIC na formação dos indivíduos, suscitam discussões sobre artefatos e formas de utilizá-las pedagogicamente de forma a contribuir com os processos de ensino e aprendizagem.

A discussão sobre o uso das TICs na educação vem de muito antes da BNCC. O estudo realizado por Peixoto & Araújo (2012), que tinha por objetivo analisar os fundamentos do discurso sobre as relações entre Educação e Tecnologia, tomando por base um estado da arte sobre os usos do computador na educação escolar, no período de 1997 e 2007, pode mapear diversos autores que apresentaram as tecnologias como recursos importantes para o processo de ensino aprendizagem e influenciam este discurso. As autoras puderam identificar a presença de duas categorias no discurso pedagógico contemporâneo sobre os usos do computador na educação: o computador como recurso didático-pedagógico e o computador como recurso político-pedagógico.

Dentre os autores que alinham os discursos com a categoria que discute o computador como recurso didático-pedagógico, encontra-se Lévy (1988; 1995; 1996; 1999). Segundo as autoras, Lévy

é referenciado, principalmente, nos estudos que tratam da “Internet e Educação”, designando conceitos como “ciberespaço”, “virtual e real”, “ecologia cognitiva”. Este autor tem alicerçado o discurso pedagógico sobre o uso do computador na educação.

Piaget (1968; 1973) é referenciado em artigos para sustentar discussões que tratam do uso do computador na perspectiva da mediação pedagógica e do desenvolvimento da aprendizagem do aluno que utiliza o computador, considerando as fases de desenvolvimento, isto é, há momentos para aprender cada coisa, usando ferramentas específicas.

Quando o discurso abrange os processos de ensinar e aprender, Vygotsky (2003) é a referência mais utilizada. Este ressalta a importância do meio para o desenvolvimento da criança e para a formação da mente, por meio de atividades colaborativas, cooperativas e interativas, tal como apresentado por Ramos & Quartiero (2005, apud, PEIXOTO & ARAÚJO, 2012, p. 258), no qual trazem reflexões sobre metodologias baseadas em problemas para o desenvolvimento de processos colaborativos sustentados pelo uso de ferramentas da internet em ambiente educacional.

Ainda, segundo Peixoto & Araújo (2012), quando o objetivo era de encontrar formulações para o uso pedagógico do computador na educação escolar, o estado da arte se apoia, especialmente, em Moran (2002) e Valente (2007;2008), os quais oferecem sugestões para aplicação e uso do computador como melhoria da aprendizagem a partir do seu uso.

Assim, é necessário reconhecermos a diferença entre Pensamento Computacional, Cultura Digital (termos já definidos em nossa pesquisa) e Letramento Digital. Souza (2007) organizou um levantamento das diferentes definições de Letramento Digitais. Uma primeira definição que destacamos é proveniente de um relatório denominado “Digital Transformation” que define o pensamento digital como ““usar a tecnologia digital, ferramentas de comunicação e/ou redes para acessar, gerenciar, integrar, avaliar e criar informação para funcionar em uma sociedade de conhecimento” (SERIM, 2002 citado por SOUZA, 2007, p. 57). A segunda definição foi elaborada pela Association of College & Research Libraries, a qual define o Letramento Digital como “uma série de habilidades que requer dos indivíduos reconhecer quando a informação faz-se necessária e ter a habilidade de localizar, avaliar e usar efetivamente a informação necessária” (CESARINI, 2004, s/p citado por SOUZA, 2007, p. 57).

Embora o Pensamento Computacional também visa a utilização de sistemas computacionais e tecnologias para resolver problemas em diversos cenários, assim como a Cultura Digital, utilizando-se de habilidades para identificar, manusear e processar informações necessárias assim como o Letramento Digital, o uso exclusivo desses sistemas e tecnologias para realização de tarefas ou como

meios pedagógicos não se iguala e nem deve ser confundida como o exercício do Pensamento Computacional.

As possibilidades de abordar o Pensamento Computacional na educação têm sido bastante explorados nos últimos anos. Acreditamos que o gráfico de KAMINSKI; KLÜBER; BOSCARIOLI (2021) (vide Figura 8) que mostra a trajetória da Informática na Educação Brasileira dos anos 1970 a 2018, tende a ser complementado com o avanço das pesquisas na área de Pensamento Computacional nos últimos 4 anos, estes ainda não contemplados pelo gráfico. É possível notar inclusive o crescimento do número de publicações que abordam esse tema em vários níveis de ensino e por meio de múltiplas estratégias, utilizando-se de diferentes ferramentas digitais ou não.

Wing (2006) já havia mencionado que o Pensamento Computacional seria uma habilidade fundamental usada por todas as pessoas no século XXI, assim como ler, escrever e calcular.

Barcelos e Silveira (2012) ressalta a abrangência do ensino de conceitos de Computação, passando-se a considerá-la como ciência básica. Por esse motivo, o subconjunto de habilidades e competências da área - que chamamos de Pensamento Computacional - deveria ser desenvolvido nos alunos desde a educação básica.

França & Tesdeco (2015, p.1464 apud NUNES, 2018) ressalta que o ensino da Computação como saber necessário na formação básica dos cidadãos do século XXI “passa por noções de modelos computacionais, algoritmos, complexidade, autômatos, entre outros conteúdos”. Por essa razão, o ensino ao manuseio corretos dos periféricos do computador, como mouse e teclado, a manipulação de softwares e programas, tais como editores de texto, criadores de apresentações de slides e o uso adequado e responsável da Internet, não é suficiente na atualidade. Para as autoras, as habilidades chamadas de Pensamento Computacional, quando integradas à formação básica do indivíduo, possibilita que os futuros profissionais do mercado de trabalho, jornalistas, médicos, músicos, engenheiros, advogados, professores, dentre outros, resolvam problemas de suas respectivas áreas de atuação.

Apesar dos supracitados benefícios, o desenvolvimento do Pensamento Computacional no currículo brasileiro ainda não tem um reconhecimento formal, todavia, França & Tesdeco (2015) apontam duas possibilidades a serem consideradas para se iniciar o ensino do Pensamento Computacional no currículo escolar atual.

A primeira diz respeito à criação de uma disciplina obrigatória, assim como Matemática, que trate dos fundamentos da Computação nos diferentes anos da

educação básica. Nessa conjectura, deve-se atentar para diversos aspectos, dentre eles, os possíveis efeitos da introdução de uma nova disciplina no currículo escolar na motivação e na aprendizagem dos estudantes. A segunda possibilidade tem uma perspectiva interdisciplinar onde o pensamento computacional é trabalhado atrelado às disciplinas já existentes no currículo escolar. Nesse contexto, a promoção da aprendizagem de conceitos computacionais é feita aliada à construção de conhecimento de conteúdos curriculares dos diferentes anos da educação básica (FRANÇA & TEDESCO (2015, p. 1466).

Sobre esta questão, Barr e Stephenson (2011) propõem que não precisamos optar por um e por outra, mas que na verdade ambas as possibilidades deveriam ser implementadas. Para isto, a comunidade de educação em Ciência da Computação pode desempenhar seu papel, destacando as práticas de solução de problemas algorítmicas e as diferentes aplicações da computação em todas as disciplinas, utilizando-se da aplicação de métodos e ferramentas computacionais nas diversas áreas de aprendizagem. Concomitantemente, os educadores de Ciência da Computação devem compreender as complexidades do ambiente educacional no Ensino Fundamental 1: Séries Iniciais e incorporar os conhecimentos da Computação em atividades de extensão, isto é, não necessariamente no ambiente da sala de aula.

Para que haja o sucesso da efetiva implementação do Pensamento Computacional nas escolas, dependemos de uma reflexão sobre o que é necessário para darmos os primeiros passos rumo a formalização desses conceitos nos currículos escolares. Barr e Stephenson (2011) apontaram dois inícios para efetivas mudanças nos currículos das escolas norte-americanas, mas que são possivelmente aplicáveis a outros países como o Brasil. Os autores sustentam que

O tipo de mudança educacional sistêmica e sustentada proposta necessita de dois conjuntos de recursos. São necessários recursos para ajudar a informar os formuladores de políticas educacionais sobre a natureza e importância do pensamento computacional, suas conexões com as metas de aprendizagem que já podem ter sido estabelecidas para os alunos (por exemplo, padrões nacionais e estaduais) e formas de melhor integração dentro da estrutura mais ampla para a aprendizagem e o sucesso dos alunos. Os professores também precisam de recursos que demonstrem como integrar esses novos conceitos de maneira mais adequada e eficaz, primeiro em sua própria esfera de conteúdo e conhecimento pedagógico e, em seguida, em seu conteúdo e prática de sala de aula (BARR e STEPHENSON, 2011, p. 53).

A educação quando é bem estruturada, possui diretrizes e componentes curriculares adequados as necessidades dos discentes e docentes, tende a formar indivíduos mais qualificados e preparados para vida e para o mercado de trabalho. Adotar o Pensamento Computacional nas escolas só reforçará esta afirmativa.

Este estudo limita-se a trabalhar as habilidades de Algoritmos e Programação, apresentado na segunda etapa do Modelo de Guarda e Pinto (2020) como Resolução do Problema. Desta forma, nos tópicos a seguir, será apresentado algumas reflexões sobre o ensino e aprendizagem de Algoritmos e Programação no Ensino Fundamental 1: Séries Iniciais, quais recursos são empregados e que metodologias norteiam as práxis na escola.

2.2.2.1 Ensino e Aprendizagem de Algoritmos e Programação no Ensino Fundamental 1: Séries Iniciais

O ensino e aprendizagem de Algoritmos e Programação desde a educação básica tem sido alvo de estudos em várias partes do mundo (MARQUES, 2009; CALDER, 2010; PINTO, 2010; CORREIA, 2012; MORALLES, 2013). Embora não exista um consenso sobre uma definição para o Pensamento Computacional, seus defensores concordam em afirmar que a Educação Básica pode e deve usufruir de diversos benefícios quando articulada a ele (BARBOSA; MALTEMPI, 2020).

Visando melhorar o raciocínio lógico dos estudantes e também aumentar a criatividade para encontrar soluções para problemas reais, alguns países já realizaram modificações no currículo nacional, inserindo conteúdos de Algoritmos e Programação de forma transversal em outras disciplinas ou formal por meio de uma disciplina, a começar pela educação básica. A Inglaterra modificou o Currículo da Educação básica nacional e, a partir de 2014, as escolas começaram a lecionar computação para as crianças das escolas primárias e secundárias (BERRY, 2013). No Japão, o ensino de programação é trabalhado de forma transversal pois é inserido na disciplina de Matemática, da Educação Básica. Desde as séries iniciais, os alunos já começam suas primeiras atividades, enquanto desenvolvedores de programas para computador.

Reflexões sobre o motivo do elevado número de evasão dos cursos de graduação em Ciências da Computação e suas variações têm apontado que as dificuldades encontradas pelos alunos estão ligadas ao fato de que estes precisariam ter aprendido a pensar como cientistas da computação antes de alcançar o ensino superior, não quando alcançam. Barr e Stephenson (2011) argumentam que não é insuficiente esperar que um aluno alcance o Ensino Superior para introduzir conceitos da Computação, isto porque todos os alunos de hoje “viverão uma vida fortemente influenciada pela

computação, e muitos trabalharão em áreas que envolvem ou são influenciadas pela computação” (p. 49).

As discussões sobre a produção de um currículo contendo conteúdo da Ciência da Computação refletem um progresso no modelo de sistema educacional. Objetiva-se a partir de tais discussões, estabelecer um espaço para o desenvolvimento de novas habilidades imprescindíveis para os bons profissionais no mercado de trabalho no século XXI. As capacidades de simplificar, interpretar e resolver problemas ganham um destaque especial na sociedade, acompanhadas das capacidades de liderar, trabalhar em equipe e dominar o uso de tecnologias para quaisquer áreas do conhecimento (SCAICO, 2012).

Souza, Falcão & Mello (2021) realizaram uma investigação utilizando-se do método de revisão de literatura que tinha o objetivo além de outras questões, identificar quais as principais técnicas são utilizadas para propor o ensino de programação para crianças. Como resultado do estudo, os autores identificaram sete técnicas. A primeira e mais citada dentre os trabalhos selecionados para análise está, o desenvolvimento de jogos para ensinar programação; A segunda mais citada é a utilização de abordagens de programação através de pequenos algoritmos que precisam ser resolvidos através das fases do Pensamento Computacional, desde a decomposição e abstração das informações, até a organização das etapas a serem seguidas para resolver o problema final; A Terceira técnica mais citada faz uso do Storytelling como suporte para tornar as aulas mais atrativas, divertidas e compreensíveis. Nesta técnica, a problemática é tratada como narrativas e histórias que facilitam na elaboração da solução; A quarta técnica faz uso de robótica para chamar a atenção dos alunos, ao proporcionar um momento de criação de dispositivos eletrônicos; A quinta técnica, utiliza oficinas práticas com conteúdos lúdicos, como vídeos, histórias e outros para desenvolvimento de programação de ações e funções de personagens; A sexta, realiza o ensino interdisciplinar, no qual os conceitos de programação são introduzidos em conteúdos de outras disciplinas; E, por fim, a sétima técnica utiliza a computação desplugada, onde os alunos são apresentados a conceitos de programação sem a utilização de dispositivos eletrônicos.

2.2.3 Recursos didáticos para o ensino e aprendizagem de Algoritmo e Programação no Ensino Fundamental 1: Séries Iniciais

Outra questão importante no processo de ensino e aprendizagem de Algoritmo e Programação no Ensino Fundamental 1: Séries Iniciais são os recursos empregados no suporte dessas atividades. Existem diversas estratégias que incorporam diferentes recursos, Plugadas (com o uso do

computador) ou Desplugadas (sem uso do computador), para diferentes cenários e diferentes públicos.

Gomes, Melo e Tedesco (2016) apresentam estratégias que incorporam **jogos digitais** ao longo do processo de ensino e aprendizagem de conceitos de Algoritmos e Programação (KARADENIZ, SAMUR e OZDEN, 2014). Devido ao crescimento do mercado de jogos digitais, é possível notar um aumento na quantidade de aplicações voltadas para o ensino de programação para crianças (FALCÃO, GOMES e ALBUQUERQUE, 2015; GIBSON e BELL, 2013). Geralmente, estes jogos abrangem conceitos de sequenciamento de instruções, repetições, condicionais, funções, entre outros tópicos de programação proposto pelo currículo CSTA/ACM e *Computing at School* para a faixa etária dos infantes (GIBSON e BELL, 2013; GOMES e ALENCAR, 2015). Alguns desafios na utilização de jogos digitais também foram apontados pelos autores acima, dentre eles:

1. As interfaces de alguns jogos exigem muita leitura, limitando o uso com as crianças em fase de letramento (Falcão, Gomes e Albuquerque 2015); 2. O feedback apresentado por alguns jogos não é suficientemente claro para o público alvo (Falcão e Barbosa 2015; Gomes et al. 2015); 3. Nenhum dos jogos pesquisados é adaptativo, 4. Poucos jogos disponibilizam recursos que permitam ao professor o acompanhamento individual de cada aluno, por exemplo: mensurar a quantidade de erros, o tempo despendido em cada fase, dentre outros (Falcão e Barbosa 2015; Gomes et al. 2015) (GOMES, MELO, TEDESCO, 2016. P. 472).

Ferri e Rosa (2016), realizaram uma revisão de literatura em bases de buscas nacionais e internacionais, tais como IEEE Institute of Electrical and Electronic Engineers, ScienceDirect, ACM Digital Library, Google Acadêmico, e anais do Simpósio Brasileiro de Informática na Educação (SBIE). A pesquisa tinha por objetivo responder 5 questões de pesquisa, das quais a quinta – “Quais estratégias e ferramentas didático-pedagógicas relacionadas ao ensino de programação de computadores, estão apoiando o processo de ensino e aprendizagem no Ensino Fundamental 1: Séries Iniciais?” tem ligação direta com este tópico. Os autores puderam identificar que as **plataformas computacionais** do tipo “Micromundos”, tais como *Scratch*, *Super Logo*, *Flash CS3 Professional* e *AlgoBlock*.

A primeira plataforma mais citada dentre os artigos escolhidos para a análise é o Scratch. Desenvolvida pelo Instituto Tecnológico de Massachussets – MIT e pelo grupo KIDS da University of California, Los Angeles (SCRATCH.MIT, 2016). É uma plataforma que trabalha conceitos de programação através de uma linguagem de programação visual, por meio de blocos que representam instruções. Projetada para crianças com mais de seis anos de idade, o Scratch oferece um ambiente de desenvolvimento e um site no qual a comunidade pode hospedar os projetos, executar, estudar e reutilizar programas de outros, por meio de um ambiente altamente colaborativo. Esta plataforma

auxilia no desenvolvimento da criatividade, da racionalização sistemática e no trabalho colaborativo – competências consideradas essenciais no século XXI pela a UNESCO (2009).

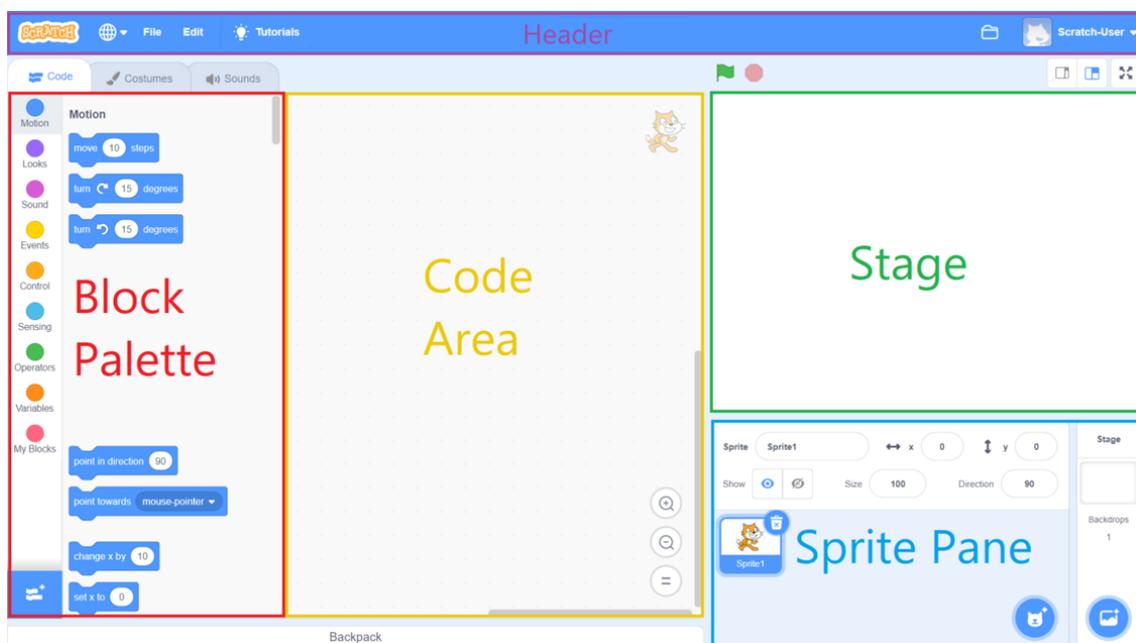


Figura 11. Organização da interface do Scratch.
Fonte: Scratch (2022)

A revisão de literatura de Souza, Falcão & Mello (2021) apresentaram resultados que confirmam o Scratch como recurso didático mais usada no ensino e aprendizagem de programação para crianças. Além do Scratch, outros três recursos foram identificados com o mesmo propósito: Arduino, Lightbot e Code.org. A relação entre os quatro recursos é apresentada no Gráfico 1.

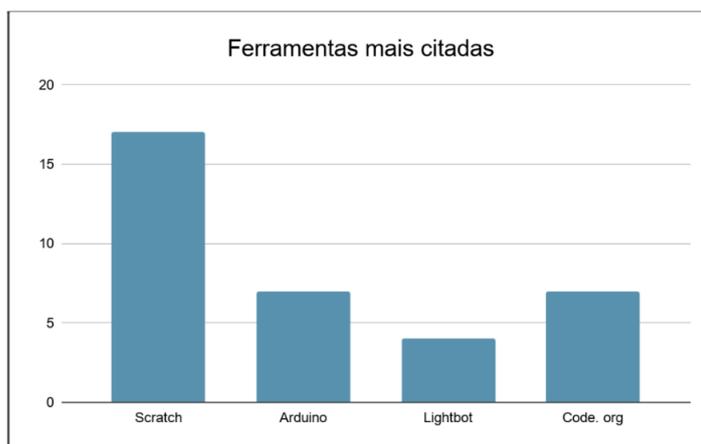


Gráfico 1. Ferramentas mais citadas no ensino de programação para crianças.
Fonte: Souza, Falcão & Mello (2021, p. 1272)

Garlet, Bigolin, & Silveira (2018) apresentam outra ferramenta para o ensino de programação, o VisuAlg. Os comandos utilizados na construção dos Algoritmos são baseados no Portugol, linguagem originada da união do Português com a Linguagem de Programação ALGOL. Apesar dos autores não destacarem a faixa etária ideal para a utilização do VisuAlg, afirmam ser muito útil para os alunos que estão iniciando os estudos na área de programação de computadores. Vale ressaltar que para utilizar a ferramenta, é necessário que o usuário domine a leitura e a escrita como pré-requisitos mínimos, bem como as operações matemáticas básicas.

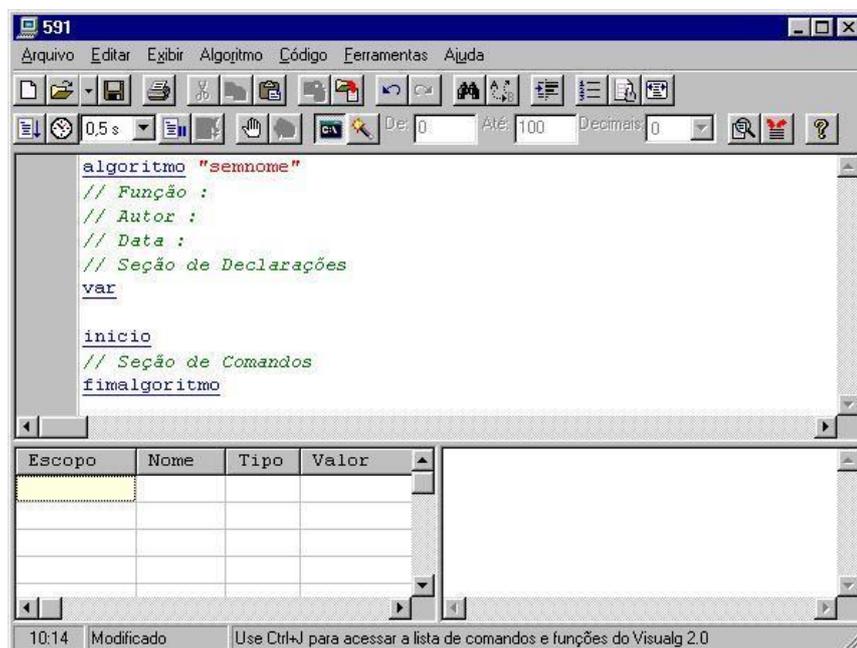


Figura 12. Interface do VisuAlg.
Fonte: Sítio VisuAlg (2022).

2.2.3.1 Metodologia Desplugada

As ferramentas digitais e tecnológicas como os computadores, celulares e a internet são excelentes meios para o desenvolvimento do Pensamento Computacional. Tais recursos tem sido fortemente empregado para o ensino e aprendizagem de Algoritmos e Programação com crianças, mas não é a única forma. É possível trabalhar conceitos de programação por meio da Computação Desplugada.

Araújo et al. (2015, p. 132), conceitua a Computação Desplugada, (no inglês *Computer Science Unplugged*) como “um método que defende o ensino de computação sem o uso do computador”. Do Santos et al. (2016, p. 12) afirma que esse método “tem como objetivo determinar os conhecimentos sobre ciência da computação, sem o uso de computadores ou qualquer meio digital.” A Computação

Desplugada oportuniza que o ensino da computação alcance lugares remotos, como pouca ou nenhuma infraestrutura (computadores, notebooks, tablets, celulares, internet, entre outros).

Além da metodologia favorecer escolas que não possuem infraestrutura tecnológica, o estado do Amazonas, localizado na Região Norte do Brasil pode se beneficiar dos resultados desta pesquisa, visto ser um estado com grande área territorial, das quais muitos alunos se encontram no cenário onde há falta de equipamentos tecnológicos, profissionais qualificados e professores especializados para realizar tarefas que se utilizam na informática.

Além disto, a Computação Desplugada proporciona uma aprendizagem dedicada, isto é, reduz drasticamente as distrações que estão associadas ao uso do computador, tablet ou celular e direciona toda a atenção dos alunos para a atividade que está sendo realizada. As atividades de computação unplugged permitem que os alunos desenvolvam o Pensamento Computacional para resolver problemas reais, sendo o ator que realiza todas as ações necessárias.

2.3 Design Centrado no Usuário

A *World Design Organization* define o Design como “um processo estratégico de resolução de problemas, que direciona a inovação, contribui para o sucesso do negócio e conduz a uma maior qualidade por todo o ciclo de vida dos produtos, dos serviços, dos sistemas e das experiências (WDO, 2022).” O Designer, como personagem chave nesse processo, resolve um problema diferente do engenheiro. O engenheiro faz de forma direta e muitas vezes linear. Já o designer se envolve em um processo fracionado e iterativo.

Quando dizemos que o Design de um produto ou serviço é Centrado no Usuário, significa que o usuário ou cliente é envolvido nos processos de desenvolvimento, participa diretamente das fases, fornece informações e ajuda a equipe de desenvolvimento na construção de alternativas de solução durante o andamento do projeto. O princípio do DCU coloca o usuário no centro das soluções de design, oportunizando que ele faça parte da equipe nas iterações que ocorrerão em cada fase do projeto (NORMAN & DRAPER, 1986).

2.3.1 Origem e Definição

Os estudos na área de interação do homem com as tecnologias começaram a ser realizados sob o nome de Interação Humano-Computador - IHC. À medida que as tecnologias foram surgindo, a tomada de decisão para adquiri-la era norteada pelo o que a tecnologia fazia e a quantidade de ferramentas que ela continha. Com o passar do tempo, o nível de exigência dos usuários aumentou e

não se é mais tolerado interfaces mal concebidas, que possuem uma usabilidade ruim e causam uma experiência de uso desagradável, difícil, cansativa, longa ou confusa.

A IHC passou a ser considerada como área do conhecimento na década de 70. Suas origens estão na Ergonomia/Fatores Humanos, na Psicologia Cognitiva, no Design e na Ciência da Computação (QUARESMA, 2018). No início dos estudos, a IHC tratava de questões da interação do homem com o hardware (equipamento físico como chaves, fechaduras, correntes e peças do próprio computador) e era nomeado de “human factors in computers” (HARTSON E PYLA, 2012), mas logo o foco mudou para a interação com os softwares (programa no computador, um aplicativo no celular ou sistema operacional de um dispositivo) e foi chamada de “fatores humanos em engenharia de software”.

Na década de 80, a psicologia cognitiva dá origem a um novo termo, a Usabilidade. A ISO 9241-11 (1998) define a Usabilidade como “a medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso”. Portanto, a Usabilidade está fundamentada em 3 pilares fundamentais da interação do usuário com sistemas computacionais: eficácia, eficiência e satisfação, sendo os dois primeiros medidos de forma objetivo e o terceiro, de forma subjetiva.

A ISO 9241 define Eficácia, Eficiência e Satisfação da seguinte maneira:

Eficácia refere-se à capacidade de um dispositivo de atingir (ou não, neste caso ineficácia) um determinado objetivo. [...] A definição de eficácia relaciona-se assim com a medição do resultado obtido (o desempenho) e não com o processo ou a actividade que conduziu ao resultado. Eficiência é a capacidade de produzir uma determinada tarefa com o mínimo de esforço: quanto menor o esforço, maior a eficiência. A satisfação refere-se ao nível de conforto sentido pelo usuário ao utilizar um produto. É a aceitação de que o objeto é um meio valioso de satisfazer os objetivos do usuário. [...] A satisfação corresponde a uma reação afetiva que diz respeito ao ato de usar um dispositivo e que pode estar associada ao prazer que o usuário recebe em troca de seu ato. A satisfação é, portanto, uma avaliação subjetiva resultante de uma comparação entre o que o ato de uso traz ao indivíduo e o que ele espera receber (ISO 9241-11 (1998) apud Brangier & Barcenilla, 2003, p. 43, 45, 50).

Os componentes de Usabilidade vinham sendo estabelecidos em IHC enquanto os pesquisadores da área debruçavam-se sobre as interações humanas a partir dos softwares, mais precisamente por meio da interface, com base nas premissas da Psicologia Cognitiva. Como resultados dessas pesquisas surgem os princípios, critérios e heurísticas de usabilidade, trazendo a evidência, autores como

Shneiderman (1987), Norman (1988), Bastien e Scapin (1993) e Nielsen (1994), sendo o último o formulador das 10 Heurísticas de Usabilidade. A partir desses estudos,

A compreensão do comportamento e desempenho humano na interação, considerando a cognição, memória (curto e longo prazo), percepção, atenção e tomada de decisão passam a ser fundamentais para o desenvolvimento de soluções adequadas na comunicação entre o humano e o computador (QUARESMA, 2018, p. 90).

Agora, o processo de concepção de um produto, um serviço ou um sistema precisa considerar minimamente esses três fatores - eficácia, eficiência e satisfação - para construir uma melhor relação do que está sendo construído com o usuário/cliente. Este processo deve se basear nas necessidades implícitas e explícitas do usuário, no seu comportamento, na sua forma de pensar, na sua forma de guardar informações, na sua forma de perceber o mundo ao seu redor, no seu nível de atenção, e não simplesmente seguir um rígido conjunto de etapas ou metodologia. Segundo Azevedo & Gibertoni (2020), os projetos que não consideram o usuário com fator central nas fases de concepção poderão ter deficiências de usabilidade e de experiência do usuário, gerando assim, erros no projeto, reclamações dos usuários, deficiência ergonômica entre outros fatores que contribui para insatisfação do usuário/cliente.

2.3.2 User Experience Design

Diversas tecnologias têm surgido dia após dia e aumentando as oportunidades de quem quer trabalhar com construção de novos produtos digitais, sistemas interativos e aplicações para mobile. Hoje o interesse dos usuários por certa tecnologia em detrimento de outra, está diretamente ligado pelas formas de interações possíveis entre usuário e tecnologia, como tais sistemas começam a fazer parte de suas vidas de forma satisfatória, como são encantados pela forma que elas resolvem seus problemas e como proporcionam uma boa experiência de uso em todos os momentos de interação.

Entre a década de 1990 e 2000, novos componentes da interação além da eficácia, eficiência e satisfação começaram a ser questionados. A partir de então, passou-se a pensar sobre os outros componentes que poderiam fazer parte da interação humana com um sistema, como prazer, emoção, afetividade, encantamento, dentre outros. A Usabilidade media a satisfação a partir do resultado satisfatório da relação eficácia e eficiência de um sistema, mas em determinado momento, este resultado passou a ser o mínimo esperado pelos usuários. Jordan (2002, apud QUARESMA, 2018, p. 90), faz uma comparação da hierarquia das necessidades humanas proposto por Maslow (1970) com

o que ele chama de “nova ergonomia” ou “novos fatores humanos”. A pirâmide da Nova Ergonomia propõe que as pessoas querem mais do que usabilidade na interação, elas querem ter prazer nessa interação.

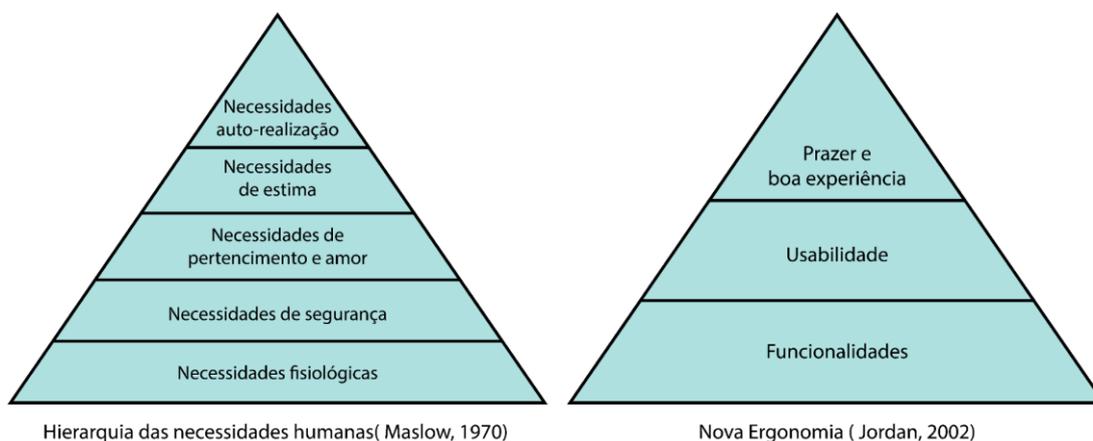


Figura 13. Comparação da Hierarquia das Necessidades Humanas e a Nova ergonomia.
Fonte: Adaptado de Jordan (2002, apud QUARESMA, 2018, p. 90)

A partir deste momento, a usabilidade começa a ser vista como requisito mínimo em uma interação. Não é suficiente que um produto ou serviço resolva o problema do usuário ou cliente. O produto/serviço precisa encantar, precisa proporcionar uma excelente experiência desde o momento que o usuário/cliente pensa dele, no período que está utilizando-se dele, até mesmo depois que a interação termina, a experiência sempre fica motivando-o ou não a voltar e comprar novamente ou nunca mais tornar a ter a ideia de utilizá-lo.

Estes conceitos deram origem ao termo User Experience - UX (Experiência do Usuário em português). O termo foi popularizado por Donald Norman no início dos anos de 1990, quando nomeou seu grupo de trabalho como “the User Experience Architect’s Office”, enquanto trabalhava na Apple (NORMAN, 2013). O conceito se relaciona a todas as situações em que um usuário se envolve com um produto/serviço, pensando nele, interagindo com ele, descartando-o ou fazendo uma nova aquisição (ALBEN, 1996; FORLIZZI & FORD, 2000).

Ao considerar a UX do usuário desde os estágios iniciais do Design em um processo de concepção de um novo produto ou serviço, diminuimos a lacuna entre a intenção do produto/serviço e a expectativa do usuário. Além disso, a melhoria em UX tem influenciado e motivado os usuários ou clientes a usarem um produto ou serviço (MAGIN et al., 2015), isto quer dizer que oferecer uma boa experiência aos usuários, contribui para uma fidelidade e evita a rotatividade. Portanto, é importante

considerar a experiência do usuário desde o estágio inicial de design até o pós-lançamento (LIN et al., 2016).

Kuniavsky (2010, p. 14) define UX como:

“a totalidade das percepções dos usuários finais à medida que eles interagem com um produto ou serviço. Essas percepções incluem eficácia (quão bom é o resultado?), eficiência (quão rápido ou barato é?), satisfação emocional (quão bom é?), e a qualidade da relação com a entidade que criou o produto ou serviço (que expectativas ele cria para interações subsequentes?).”

Apesar da experiência ser algo que simplesmente acontece na interação do usuário com um produto ou serviço, “ainda é resultado de um bom projeto que segue à risca os princípios de usabilidade, e este continua a ser um ponto de partida para uma boa experiência” (QUARESMA, 2018, p. 91). O Designer de Experiência do Usuário - UX Designer - precisa construir projetos que resultem em boas experiências, sejam projetos digitais ou não. Para tal, o Designer precisa dominar técnicas e metodologias que serão empregadas na pesquisa de mercado, nas regras de negócio, nas pesquisas com os usuários, nas pesquisas etnográficas, no mapeamento de jornada de usuário, até encontrar uma solução que seja útil, fácil de usar e prazerosa para o usuário. Desta forma, é fundamental compreender as necessidades, as dores, as dificuldades, as inquietações, os anseios, desejos e expectativas do usuário em todas as fases do design (Norman, 1986).

2.3.3 UX para crianças

Como já mencionado, a experiência do usuário não é algo que pode ser criada, projetada, construída, ela simplesmente acontece. A respeito disso, Hartson e Pyla (2012, p. 21) afirmam em seu livro, denominado “The UX Book: Process and Guidelines for Ensuring a Quality User Experience”, que a experiência pode ser apenas experimentada.

“Você não está projetando ou desenvolvendo uma boa experiência do usuário. Não há uma usabilidade ou experiência dentro do projeto; Eles são relativos ao usuário. Usabilidade ocorre dentro ou é revelado dentro do contexto de uso particular, por um usuário particular. O mesmo design, mas usado em um contexto diferente; uso diferente e/ou um usuário diferente, pode levar a uma experiência de usuário diferente.” [tradução do autor]

Desta forma, entendemos que não se pode construir um produto ou serviço esperando que todos que interajam com ele, terão a mesma e boa experiência. Contudo, delimitar corretamente o público alvo, o cliente em potencial, o usuário médio, ajudará na identificação das expectativas e norteará as

decisões projetuais, visando diminuir qualquer experiência desagradável. Por esta razão, este pesquisador compreende que é necessário identificar Metodologias e Técnicas de Design que podem ser melhor empregadas para construir uma solução para o ensino e aprendizagem de Algoritmos e Programação (objetivo da solução) centrada nas crianças (público-alvo), que compõe uma turma do ensino fundamental do Colégio Connexus (contexto específico), buscando atingir um bom nível de experiência desse público.

2.4 Metodologia para desenvolvimento de soluções centradas em crianças

2.4.1 Design Thinking

David Kelley foi um dos precursores do termo “Design Thinking” em 2003, quando já era consultor de design na IDEO, uma das maiores empresas de inovação do mundo. Na época, o Design Thinking era utilizado como estratégia de negócio, mas “passou a ganhar robustez como um processo de transformação, focado nos processos organizacionais e estratégias de pensamento criativo” (Bonini & Sbragia, 2011, p. 8). Hoje o DT é empregado como uma abordagem para solucionar problemas, inspirar a criatividade e instigar a inovação com foco no usuário.

A própria IDEO, ao conceituar a abordagem, afirma que

O design thinking é uma abordagem centrada no ser humano para a inovação que se baseia no kit de ferramentas do designer para integrar as necessidades das pessoas, as possibilidades da tecnologia e os requisitos para o sucesso do negócio (IDEO, 2022)..

Viana et al., (2012), afirma que o Design Thinking se refere a forma de pensar do Designer. Tal forma de pensar utiliza um raciocínio pouco convencional e não linear, conhecido como abdução. O Pensamento Abdução busca formular questionamentos através da apreensão ou compreensão dos fenômenos, isto é, o designer formula perguntas que serão respondidas a partir das informações coletadas durante a observação do universo que rodeia o problema.

Brown (2009 apud Bonini & Sbragia, 2011, p. 8) “propõe que o DT seja uma abordagem que utiliza sensibilidade e métodos do designer para resolver problemas e atender às necessidades das pessoas com uma tecnologia viável e comercialmente factível”.

Cardon (2010), por sua vez, descreve o Design Thinking como “uma ferramenta útil que aplica o pensamento criativo e crítico para compreender, visualizar e descrever os problemas complexos ou mal estruturados e, em seguida, desenvolver abordagens práticas para resolvê-los”.

Portanto, o Design Thinking vem como uma proposta de abordagem empática (capacidade de se colocar no lugar do outro). Para Brown (2010), o foco dessa abordagem deve ser as pessoas, isto é, uma solução criada pelo Design Thinker deve ser capaz de resolver um problema das pessoas ou ser capaz de criar uma nova experiência para pessoas. Por esse motivo, a empatia é um dos seus elementos essenciais. Sua utilização nesse projeto ajudará a enxergar as emoções geradas pelos procedimentos que prejudicam a experiência emocional, cognitiva e estética (VIANA et. al, 2012).

Segundo Brown (2008), o processo de Design Thinking é implementado a partir de 3 fases, inspiração, ideação e implementação. Viana et al. (2012), apresenta não 3 mais 4 fases: Imersão, Análise e Síntese, Ideação e Prototipação.

Neste trabalho, seremos conduzidos pelas 4 fases propostas por Viana et al. (2012). A primeira fase, chamada de Imersão, é dividida em duas etapas: Imersão Preliminar e a Imersão em profundidade. A primeira, tem o objetivo de definir o escopo do projeto e delimitar suas fronteiras. Nesta etapa também são identificados os perfis dos usuários e outros atores-chaves que deverão ser abordados. Também é levantado as áreas de interesse que serão exploradas de forma a fornecer insumos para elaboração dos temas de investigação na etapa de Imersão em profundidade. Já a etapa de Imersão em Profundidade, iniciará com a elaboração de um Plano de Pesquisa. Este plano incluirá protocolos de pesquisa primária, listagem dos perfis de usuários e atores-chaves para recrutamento e será feito o mapeamento dos contextos a serem estudado (VIANA et al., 2012).

Nesta fase, a equipe de projeto busca identificar o que as pessoas sentem, utilizando-se de técnicas de coleta de dados, a fim de gerarem diferentes tipos de conhecimento.

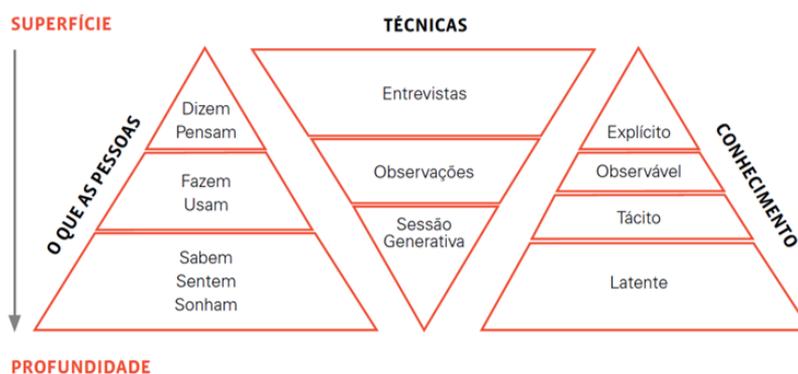


Figura 14. Estratégias empregadas e conhecimentos adquiridos na fase de Imersão.
Fonte: VIANAA et al.,(2012, p. 23).

Na fase de Análise e Síntese, as informações coletadas na fase de Imersão são analisadas e sintetizadas. Os insights são organizados de maneira a obter-se padrões e a criar desafios que auxiliem na compreensão do problema. Entende-se por insights os achados provenientes da imersão. Toda e qualquer identificação de oportunidades. As ideias são soluções geradas para atender um ou mais insights.

Na fase de ideação, é de suma importância ter uma variedade de perfis de pessoas envolvidas no processo de geração de ideias. Uma equipe multidisciplinar possibilita uma variedade de ideias a serem refinadas, por vários pontos de vista distintos. Além disso, normalmente se inclui nesse processo as pessoas que serão “servidas” pela solução, os chamados stakeholders. A fase de ideação se inicia com a equipe de projetos realizando Brainstormings ao redor do tema a ser explorado. Esta técnica de geração de ideias permite que todas as pessoas expressem seus insights com o objetivo de gerar a maior quantidade de ideias possíveis. As ideias são capturadas em Cardápios de Ideias que posteriormente serão validados ou não em reuniões com o usuário (VIANA et al., 2012).

Por fim, na fase de Prototipação é o momento de tangibilizar as ideias, transformar o abstrato em físico de forma a representar a realidade, mesmo que de forma simplificada. Esta fase é realizada para auxiliar a validação das ideias geradas na fase anterior e apesar de ser apresentada como a

2.4.2 Como aplicar Design Thinking com crianças

Apesar do Design Thinking ser utilizado como abordagem para projetar produtos incrivelmente tecnológicos e rentáveis, bem como inovar serviços prestados por empresas de grande porte, certamente pode ser aplicada com crianças a fim motivá-los à inovação. A abordagem exige organização, responsabilidade e proatividade, o que pode levar os mais descredenciados a pensar que essas habilidades são difíceis de serem construídas em crianças, o que pode tornar o emprego do Design Thinking mais trabalhoso. Tornar o aprendizado algo atrativo para as crianças é o primeiro passo para se obter toda a atenção e dedicação deles em um projeto.

O processo de Design Thinking pode ser iniciado com materiais básicos de papelarias, como post-its, canetas, lápis de cor, pincéis e papel. A facilidade de empregar essa abordagem utilizando materiais baratos e que fazem parte do dia a dia das crianças, é uma ótima chance de estimular a criatividade. Além de dispensar o domínio de ferramentas específicas.

Não há um manual pronto para utilização do Design Thinking com crianças para concepção de soluções para o ensino e aprendizagem. Existem diversas maneiras de aplicar esta abordagem e cada iniciativa irá produzir uma experiência diferente entre as crianças.

Para que haja uma organização de ideias, o mediador responsável por conduzir a abordagem junto às crianças poderá utilizar de listas, pregar post-its, colar fotos, desenhar setas, balões e outros elementos que expressam ideias, fazer colagens, construir modelos e protótipos com papel, entre outros. Desta forma, é possível incluir as crianças nas etapas de concepção de uma solução para o ensino de Algoritmos e Programação que será utilizada por elas mesmo. A proposta é que as crianças, como a parte interessada, colaborem para que junto com o pesquisador, possam construir uma solução desplugada útil, fácil de usar e prazerosa.

Ao aplicar a abordagem no Ensino Fundamental 1: Séries Iniciais, pode-se instigar as crianças a desenvolverem um senso de autonomia sobre o próprio desenvolvimento, estruturando o pensamento para alcançar as soluções, gerando empatia e criatividade.

CAPÍTULO 3

Materiais e Métodos

Neste capítulo são apresentados os materiais, métodos e ferramentas utilizadas para a realização desta pesquisa, de modo que foram listadas as técnicas de pesquisa, seu embasamento teórico, sua aplicação e os sujeitos envolvidos;

3.1 Métodos, técnicas e procedimentos

Neste capítulo será apresentado o processo metodológico para concepção do jogo centrado no usuário, a partir do Design Thinking, tendo em vista a sua adaptabilidade a diversos cenários, a ser aplicado com qualquer público, para concepção de qualquer tipo de produto, seja físico ou digital. Apesar das diversas subdivisões de fases desta abordagem que variam entre os autores, como afirmado anteriormente, esta pesquisa será conduzida sob as fases de Imersão, Análise e Síntese, ideação e Prototipação propostas por VIANA et al., (2012). Destaca-se que em cada fase, há técnicas e ferramentas a serem definidas e empregadas, baseado em necessidades específicas de cada fase. Desta forma, tais técnicas e ferramentas empregadas, são apresentadas dentro de cada fase.

3.1.1 Imersão

A Imersão é a primeira fase da abordagem do Design Thinking. Ela busca familiarizar o pesquisador com o contexto do projeto e é subdividida em duas partes: Imersão Preliminar e Imersão em Profundidade.

A Imersão Preliminar tem como objetivo proporcionar um entendimento inicial do problema e, se necessário, realizar um reenquadramento do mesmo. Já a Imersão em Profundidade está direcionada para a identificação das necessidades das partes envolvidas no projeto, bem como para a possível identificação de oportunidades que surgem a partir da compreensão das experiências delas em relação ao tema abordado. No entanto, esse processo de imersão profunda no contexto frequentemente resulta em uma grande quantidade de informações, o que pode dificultar a identificação das oportunidades e dos desafios.

Na fase de Imersão Preliminar, o pesquisador desempenha um papel crucial ao iniciar o processo de investigação. Aqui, ele busca obter uma compreensão inicial do problema, coletando dados básicos, revisando literatura relevante e conduzindo entrevistas preliminares. Essa etapa é fundamental para definir o escopo do projeto, identificar lacunas no conhecimento e, se necessário, reenquadrar o problema. Nesta fase, foram realizadas **Pesquisas Bibliográficas, Observações Participativa e Não participativa** na Manaós Tech e Colégio Connexus.

Na fase de Imersão em Profundidade, o pesquisador aprofunda sua investigação, realizando entrevistas mais detalhadas, pesquisas de campo e análise de dados qualitativos. Ele busca compreender em profundidade as necessidades, perspectivas e experiências das partes envolvidas no projeto, identificando oportunidades e desafios emergentes. Nessa fase, o pesquisador atua como um detetive, explorando as complexidades do contexto e preparando o terreno para as fases

subsequentes de análise e solução de problemas. Nesta fase foram realizados **Entrevista, Pesquisa Desk, Benchmarking**.

3.1.1.1 Pesquisas Bibliográficas

A pesquisa bibliográfica consiste na busca, seleção e análise de materiais já publicados sobre um determinado tema. Esses materiais podem incluir livros, artigos, teses, dissertações, entre outros. A pesquisa bibliográfica ajuda a embasar teoricamente o estudo, identificar lacunas no conhecimento existente e fundamentar as conclusões do trabalho.

Neste trabalho, contudo, a pesquisa foi realizada com 3 objetivos:

- Identificar na literatura a indicação da idade ideal para o exercício do Pensamento Computacional;
- Identificar as principais formas de ensinar conceitos da Ciência da Computação.
- E, identificar critérios ou pré requisitos que permeiam tal processo de ensino e aprendizagem de conceitos da Ciência da Computação, mais precisamente Algoritmo e Programação.

Alguns autores apresentam a Educação Básica como o ponto de partida para o desenvolvimento das habilidades inerentes ao Pensamento Computacional, dentre elas a habilidade de construir Algoritmos (BARBOSA; MALTEMPI, 2020). A partir de 2014, a Inglaterra alterou o currículo nacional da Educação Básica para incluir aulas de computação nas escolas primárias e secundárias (BERRY, 2013). No Japão, o ensino de programação é abordado de maneira transversal, sendo integrado à disciplina de Matemática do currículo da Educação Básica. No Brasil, a Educação Básica é formada pela Educação Infantil, Ensino Fundamental e Ensino médio (LDB, 2017, Título V, Capítulo I, Art. 21).

O Parecer CNE/CEB nº 2/2022, aprovado em 17 de fevereiro de 2022 – Normas sobre Computação na Educação Básica – Complemento à Base Nacional Comum Curricular (BNCC) e homologado pela Resolução nº 1, de 4 de outubro de 2022, define normas sobre Computação na Educação Básica, em complemento à Base Nacional Comum Curricular (BNCC). O parecer ressalta a adaptação de currículos nacionais supracitados, destacando que “o fizeram geralmente a partir dos Anos Finais do Ensino Fundamental ou Ensino Médio. Isso ocorreu há mais de uma década” (CNE/CEB, 2022, p. 15). Ainda segundo o parecer, apesar da diferença de terminologia, diversos países estruturam o currículo da Educação Básica em torno dos seguintes tópicos:

- 1) Algoritmos;
- 2) Programação;
- 3) Representação de dados;
- 4) Equipamentos digitais & Infraestrutura;
- 5) Aplicações digitais; e
- 6) Humanos e Computadores.

Ainda no referido parecer, a sugestões para implantação das políticas institucionais no que tange ao desenvolvimento de competências e habilidades acima citadas,

seria implementar a oferta em todo o segmento dos Anos Iniciais (1º ao 5º ano), mas considerando as especificidades do foco na alfabetização (1º ao 3º ano) e a ampliação de tópicos no contexto dos anos seguintes (4º e 5º ano), conforme disposto na BNCC e nas DCNs. Os Anos Finais (6º ao 9º ano) podem exigir mais, daí a sugestão de eventual implementação gradual, ano a ano (CNE/CEB, 2022, p. 17).

A educação infantil, etapa inicial da educação básica, busca promover o desenvolvimento integral da criança até os 5 anos de idade, abrangendo seus aspectos físicos, psicológicos, intelectuais e sociais, em complemento à ação da família e da comunidade (LDB, 2017, Título V, Capítulo II, Seção II, Art. 29).

O ensino fundamental, por sua vez, com duração de 9 anos e obrigatório, é gratuito nas escolas públicas e tem início aos 6 anos de idade, visando à formação básica do cidadão (LDB, Título V, Capítulo II, Seção III, Art. 32).

Conforme a Base Nacional Comum Curricular – BNCC, neste nível a criança desenvolve habilidades e competências na Língua Portuguesa, através dos eixos da Leitura, da Produção de Textos, da Oralidade, da Análise Linguística/Semiótica.

Desenvolve habilidades e competências na Artes, centradas nas linguagens Artes visuais, a Dança, a Música e o Teatro. Na Educação Física, práticas corporais ligadas ao movimento corporal, organização interna e produto cultural. Na Língua Inglesa, habilidades relacionadas aos eixos da Oralidade, Leitura, Escrita, Conhecimento linguístico e Dimensão intercultural.

Na Matemática, a BNCC orienta o desenvolvimento de competências relacionadas a Números, Álgebra, Geometria, Grandezas e Medidas, Probabilidade e Estatística.

Nas Ciências da Natureza, competências no que tange a Matéria e energia, Vida e evolução e Terra e Universo (BNCC 2017, p. 85).

Ao investigar sobre as diferentes formas de desenvolver habilidades e competências do Pensamento Computacional em crianças, principalmente relacionado aos conceitos de Algoritmos e Programação, Souza, Falcão & Mello (2021) identificaram por meio de um estudo baseado do método de Revisão Sistemática, sete técnicas para propor o ensino de programação para crianças:

1. Desenvolvimento de jogos;
2. Utilização de abordagens de programação através de pequenos algoritmos;
3. Uso do Storytelling;
4. Robótica;
5. Oficinas práticas com conteúdos lúdicos, como vídeos, histórias e outros
6. Ensino interdisciplinar;
7. Computação desplugada.

Ao analisar as diversas abordagens de ensino desses conceitos por meio das técnicas mencionadas, fica claro que certas habilidades e competências devem ser desenvolvidas previamente. Somente então o processo de abstração, decomposição, reconhecimento de padrões e algoritmos pode ser efetivamente realizado.

Portanto, neste estudo, considera-se que a idade ideal para aprender conceitos de Algoritmos e Programação é aos 6 anos, ou seja, no início do Ensino Fundamental. Para isso, o aluno deve possuir habilidades de leitura e escrita, assim como conhecimento de sequências, reconhecimento de ordem e numeral. Essa conclusão é reforçada pelo fato de que as diferentes formas de ensino desses conceitos se beneficiam dessas capacidades para o desenvolvimento de novas habilidades e competências.

3.1.1.2 Observação Participativa

A observação participante é uma abordagem qualitativa que tem suas origens na pesquisa etnográfica convencional. Segundo Gio (2017, p. 29), as observações permitem o “estabelecimento de relações entre os fatos no dia a dia que forneçam os indícios para a solução dos problemas propostos pela ciência”.

Para Batista Correa (2009), esta envolve a interação direta, frequente e prolongada do pesquisador com os membros da comunidade estudada, ocorrendo nos contextos culturais dos participantes. Nesse método, o próprio pesquisador se torna uma ferramenta de pesquisa fundamental. É essencial que o pesquisador trabalhe diligentemente para minimizar quaisquer distorções subjetivas, a fim de compreender adequadamente os eventos e interações entre os sujeitos no contexto observado, criando uma "descrição densa" das interações sociais que ocorrem em ambientes naturais (Marietto, 2018).

A Observação é uma técnica de pesquisa frequentemente usada em conjunto com entrevistas ou outras técnicas, como análise de documentos. No entanto, também pode ser usada sozinha. Para usá-la de forma científica, é importante atender a critérios como ter objetivos claros, um plano bem organizado, validação e verificação, precisão e controle.

Conforme Spradley (1980), na "Observação Participante," os objetivos vão além de apenas descrever uma situação. Ela ajuda a entender o significado, direção e dinâmica de cada momento. Ao lidar com a intersubjetividade presente, a observação no ambiente real facilita a compreensão do que está acontecendo, contanto que sejam considerados aspectos essenciais no local.

Nesta fase é importante compreender o processo de ensino e aprendizagem de Algoritmos e Programação no Ensino Fundamental 1: Séries Iniciais a partir de dois agentes principais: o professor e o aluno. Visando a observação a partir da perspectiva do professor, este pesquisador se propôs a ministrar dois cursos, o primeiro na Sede da Manaós Tech e o Segundo no Colégio Connexus.

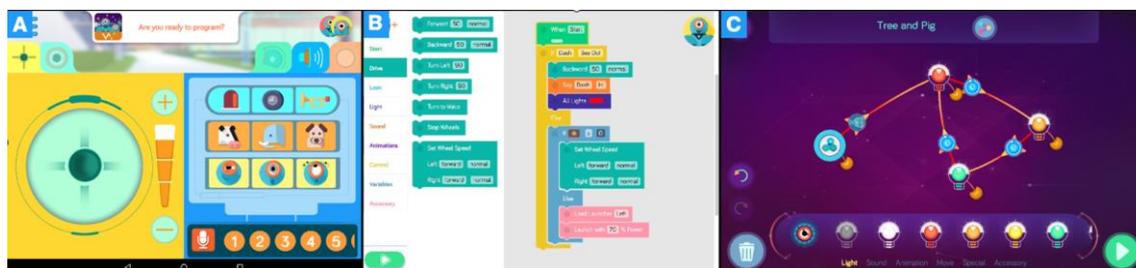
O primeiro curso foi de Introdução a Mecânica de Robôs na Sede da Manaós Tech, para crianças com faixa etária entre 5 a 7 anos, onde foram apresentados conceitos introdutórios da Robótica e Programação de Robôs. No primeiro momento do curso, houve a exploração do assunto através de recursos de multimídia como vídeos, sons e imagens em uma apresentação de slide. Esta fase teve duração de 1:30 minutos. No segundo momento, foi proposto uma atividade prática na qual os alunos utilizaram como recurso lúdico didático o *Gears Gears*, um conjunto de peças e engrenagens que podem ser acopladas umas as outras em diversos sentidos, a fim de formar diferentes figuras de

Robôs. A atividade possibilitou que os alunos reproduzissem os conceitos de integração das engrenagens para funcionamento dos robôs de forma totalmente desplugada (Figura 15). Este segundo momento teve duração de 2 horas e 30 min.



Figura 15. Alunos participando do Módulo de Mecânica de Robôs na Manaós Tech.
Fonte: O autor (2024).

O segundo curso foi ministrado no Colégio Connexus, intitulado Lógica de Programação com o *Dash*, um robô fabricado pela empresa Wonder Workshop⁹, que pode ser controlado por um controle remoto (A), por comandos estruturados em blocos (B) e através de uma interface que simulava uma linha do tempo (C), representando uma programação estruturada, conforme Figura 16. O público deste curso eram alunos do Colégio Connexus, em uma faixa etária de 06 a 07 anos, que cursavam o 1º ano do Ensino Fundamental, nos turnos Matutino e Vespertino. Nesta aula, os alunos aprenderam sobre o funcionamento, tipos e características dos robôs. Após a mediação, os alunos puderam controlar o Dash por meio dos seus 3 tipos de controles, em uma interface apresentada em um dispositivo móvel, que se conectava com o Robô via Bluetooth. Na ocasião, os alunos eram instigados a realizar algumas atividades simples de direcionamento e acionamento de leds e recursos de som.



⁹ Disponível em: <https://www.makewonder.com/robots/dash/>

Figura 16. Interfaces das diferentes aplicações para o ensino de programação com o Robô Dash.
Fonte: O autor (2024).

Ambos os experimentos foram observados a partir da perspectiva do professor. Para que a Observação Participativa assumisse um papel científico, foram definidos alguns critérios a serem observados pelo pesquisador durante as atividades de ensino e aprendizagem de ambos os cursos, a saber: 1 - Faixa etária do público de consonância com a pesquisa; 2 - Conhecimentos prévios sobre Algoritmo e Programação; 3 - Nível do conteúdo apresentado; 4 - Recurso Lúdico Adequado; 6 - Palavras de Feedbacks Positivos; 7 - Palavras de Feedbacks Negativas e 8 - Sentimentos/Sensações Percebidas, conforme apresentados no APÊNDICE A.

3.1.1.3 Observação Não Participativa

Observação Não Participante, Não Participativa ou Observação Completa, o observador não participa de nenhuma forma do contexto social do grupo e deve utilizar instrumentos que possibilitem estudos posteriores à observação, como instrumentos de registro por imagem, áudio, vídeo e notas (Marietto, 2018).

A Observação Não Participativa foi realizada nas salas de aulas da Manaós Tech e do Colégio Connexus, com o objetivo de identificar aspectos em relação a motivações dos alunos no processo de ensino e aprendizagem dos conceitos de Algoritmo e Programação em sala de aula. Também esperava-se identificar dificuldades, necessidades não atendidas e visualizar possíveis contextos onde a aprendizagem poderia ser afetada por algum aspecto proveniente da metodologia de ensino ou recursos didáticos empregados.

Neste contexto, o pesquisador adentrou ao espaço, na condição única e exclusivamente de Observador, onde se eximiu de contatos diretos com o público observado, de forma não-intrusiva, comportando-se de maneira discreta e posicionando-se em espaços menos evidentes para os alunos, de forma a diminuir qualquer possibilidade de interferência no contexto da pesquisa. Na ocasião, as aulas eram mediadas por professores da Sede da Manaós Tech e quando realizadas no Colégio Connexus, pela professora titular do Laboratório de Inovações. As observações nos dois espaços se perpetuaram por mais de 80 horas, divididas em 4 meses. O público alvo da Observação Não Participante é o mesmo Observação Participante.

A maior parte da Observação Não Participativa ocorreu no Colégio Connexus, tendo em vista a maior quantidade de amostras observáveis. Os procedimentos aconteceram no Laboratório de Inovações, durante as aulas da disciplina de mesmo nome, nos turnos da manhã e da tarde (Figura 17).



Figura 17. Alunos do 1º ano realizando a atividade na Plataforma Code.org - Fase 5, Atividade Labirinto Clássico.
Fonte: O autor (2024).

Nesta etapa de Observação Não Participativa, utilizou-se o instrumento de Diário de Campo, uma ferramenta de registro que envolve a anotação da realidade observada para auxiliar na descrição, análise e interpretação do objeto de estudo (Tonin et al., 2018). Considerando a complexidade inerente ao trabalho de observação direta, as notas de campo destacam-se como ferramentas valiosas para registrar as observações e reflexões decorrentes do processo de pesquisa. Elas constituem um instrumento elaborado que possibilita aprofundar as informações, observações e reflexões que surgem ao longo da investigação ou durante o momento de observação (Lopes, 1993). Assemelha-se a um “diário de bordo” onde anotam-se, durante cada sessão de observação, os eventos observados e a progressão da investigação (Beaud & Weber, 1998).

Alguns autores divergem sobre os tipos de notas que devem compor o diário de campo.

Para Schatzman e Strauss (1973), geralmente, a redação das notas de campo é realizada de maneira flexível, sem uma estrutura rígida, registrando os eventos à medida que ocorrem. Isso permite compilar uma variedade de "notas substanciais", "notas metodológicas" e "notas analíticas" que desempenham um triplo papel:

- **Notas substanciais:** Descrever os atores, os cenários físicos e simbólicos investigados.

- **Notas metodológicas:** Controlar epistemologicamente e avaliar os procedimentos técnicos utilizados.
- **Notas analíticas:** Aprofundar reflexões teórico-metodológicas relacionadas ao objeto de estudo, muitas vezes de forma pessoal e menos estruturada, mas com o objetivo de promover uma interconexão entre teoria e pesquisa.

As "notas substantivas" ou "notas de observação" buscam, sobretudo, reconstituir uma descrição da organização social, com o propósito de caracterizar situações, atores e comportamentos.

Para Gerhard, et al. (2009), o diário de campo é composto de duas notas, "descritivas" e "reflexivas".

Notas Descritivas: É dedicada à captação das características das pessoas, ações e conversas observadas, alinhadas com o contexto do local de estudo (BOGDAN & BIKLEN, 1994). Envolve:

- Descrição dos sujeitos
- Reconstrução dos diálogos
- Descrição do espaço físico
- Relatos de acontecimentos particulares
- Descrição da atividade
- Comportamento, postura do observador

Notas Reflexivas: Engloba principalmente a perspectiva do observador, suas ideias e inquietações. De acordo com Bogdan & Biklen (1994), esta etapa de registro é mais subjetiva. Engloba:

- a análise
- o método
- os conflitos e dilemas éticos
- o ponto de vista do observador
- pontos de clarificação

Para construção do Diário de Campo desta pesquisa, utilizou-se um modelo adaptado de Gerhard, et al. (2009) (ver APÊNDICE B) e para registro, utilizou-se um Notebook com Editor de Texto instalado. As notas eram preenchidas durante ou após cada sessão de observação nos supracitados locais.

Por meio da Observação Não Participativa, foi possível levantar os primeiros insights sobre a motivação dos alunos no contexto de ensino e aprendizagem de conceitos de Algoritmos e Programação. Por compreender que a observação, por si só, tem poucas probabilidades de conduzir a um conhecimento suficientemente geral e explicativo, foi necessária a utilização de outros métodos para refinar os insumos, os quais serão referidos a seguir.

3.1.1.4 Entrevistas

A entrevista desempenha um papel fundamental na coleta de dados qualitativos para a investigação em questão. As entrevistas foram conduzidas com o objetivo de explorar e compreender as percepções dos professores sobre o engajamento dos alunos em atividades de programação, preferências em relação a ferramentas de ensino específicas, atitudes em relação à introdução de novas ferramentas educacionais e métodos para medir o nível de engajamento dos alunos.

Conforme elucida Prodanov e Freitas (2013), por meio da entrevista, "as informações colhidas sobre fatos e opiniões devem constituir-se em indicadores de variáveis que se pretende explicar. É, pois, um diálogo preparado com objetivos definidos e uma estratégia de trabalho".

Complementando esta ideia, Marconi e Lakatos (2003), afirmam que as entrevistas são procedimentos utilizados na investigação social e tem como objetivo principal, obter do entrevistado informações sobre um determinado assunto ou problema. Caracteriza-se por "um encontro entre duas pessoas para que uma delas obtenha informações sobre determinado assunto, mediante uma conversação de natureza profissional" (Marconi e Lakatos, 2003).

A entrevista ou inquirição contextual, segundo Holtzblatt (2003), acontece em encontros individuais realizados no espaço de trabalho do usuário, isto é, em seu contexto, onde o pesquisador se concentra em observar um fenômeno em andamento.

Para o pesquisador conduzir esta inquirição, é necessário não só observar, mas também registrar o contexto de tarefas dos entrevistados. O pesquisador analisa o que está acontecendo no exato momento da interação. O objetivo é entender o entrevistado e gerar novas ideias para novas direções de um sistema.

Nesta etapa foram realizadas três entrevistas. A primeira contou com a participação do professor Eduardo Jorge Lira Antunes da Silva, professor efetivo da Manaós Tech que atua como professor de Programação e Robótica para crianças a 3 anos. A segunda, com a Professora e Coordenadora da Manaós Tech, Ana Caroline Nunes. A terceira, contou com a professora efetiva do Laboratório de Inovações, no Colégio Connexus, professora Erika da Silva. As entrevistas ocorreram de forma

remota, utilizando o Meet como recurso de contato. Para isto, foi elaborado um roteiro de entrevista, visando explorar diversos aspectos do ensino de programação para crianças. Com perguntas que vão desde a definição do Pensamento Computacional até questões práticas de usabilidade de ferramentas. O roteiro busca capturar a experiência e perspectivas dos professores de forma abrangente, buscando entender 3 aspectos: Quem, para quem o professor ensina, perfil do público; Como; metodologias e recursos didáticos empregados na praxis; e, Quanto, qual o grau de engajamento e satisfação dos alunos na perspectiva do professor. Ao conduzir as entrevistas com base nesse roteiro, espera-se obter insights valiosos que contribuirão para uma compreensão mais profunda do processo de ensino de Algoritmos e Programação no contexto escolar, bem como para o desenvolvimento de estratégias educacionais mais eficazes. O roteiro está disponível no APÊNDICE C.

Após a realização da entrevista online, a mesma foi transcrita para um formato de formulário. Ressalta-se ainda que interjeições como “Ahh...”, “E...” e palavras isoladas que não representam dados a serem utilizados para esta pesquisa, tais como “Deixa eu pensar”, “Deixa eu ver”, “Hmmm... espera!” foram removidos a partir de uma escuta minuciosa das gravações, afim de facilitar a visualização e análise dos dados e eliminar conteúdo não pertinentes da conversa.

3.1.1.5 Pesquisa Desk

A Pesquisa Desk ou Desk Research é uma busca de informações sobre um assunto pré-definido em diversas fontes, como websites, revistas, blogs, livros, artigos, entre outros (Vianna, 2012).

A Pesquisa Desk foi realizada principalmente em sites da internet, utilizando o motor de busca da Google Inc., com o objetivo de identificar recursos didáticos digitais e não digitais usados para o ensino dos conceitos de Algoritmo e Programação para crianças do Ensino Fundamental 1: Séries Iniciais.

Os principais recursos didáticos utilizados para o ensino dos conceitos de Algoritmo e Programação para crianças do Ensino Fundamental 1: Séries Iniciais puderam ser classificados em 3 tipos: **Jogos Educacionais, Plataformas Educacionais e Plataforma de Desenvolvimento.**

Para esta pesquisa, **Jogos Educacionais** são qualquer atividade de formato instrucional ou de aprendizagem que envolva competição e que seja regulada por regras e restrições (Dempsey, Rasmussen e Luccassen (1996) citados por Botelho (2024)), sejam digitais ou não.

O Minecraft: Education Edition ¹⁰ é uma versão do popular jogo Minecraft desenvolvida especificamente para uso em ambientes educacionais. Lançada pela Mojang e pela Microsoft, essa edição tem como objetivo aproveitar a popularidade e a versatilidade do Minecraft para proporcionar experiências educacionais inovadoras. Além de jogar explorando recursos de mundo aberto, também é possível utilizar comandos em blocos para controlar as ações dos jogadores.



Figura 18. Plataforma Minecraft Education.
Fonte: O autor (2024).

Baseado no Blocos de Código do Scratch, os cartões de codificação do ScratchJr¹¹ constituem uma abordagem lúdica para introduzir a linguagem de programação visual ScratchJr a crianças de 5 a 7 anos, utilizando cartões físicos. Cada cartão contém instruções passo a passo, guiando as crianças na utilização de blocos de programação para conceber suas próprias histórias, animações e jogos interativos. Enquanto avançam por cada cartão, as crianças também aprimoram suas habilidades de pensamento computacional.

¹⁰ Disponível em: <https://education.minecraft.net/pt-pt>

¹¹ Disponível em: <https://scratchjr.fun.com/coding-blocks/>



Figura 19. ScratchJr Cards.
Fonte: scratchjrfun.com.

Foram classificadas **Plataformas Educacionais**, sites e sistemas web que reúnem atividades educativa envolvendo Algoritmos e Programação, mesmo que o conteúdo da plataforma sejam jogos, cursos ou materiais didáticos pedagógicos quaisquer.

Code.org¹² é uma plataforma que incentiva o aprendizado de conceitos de Ciência da Computação por meio de vários cursos e jogos educacionais envolvendo Algoritmo e Programação. Os cursos possuem diversos níveis, oferecendo aos usuários progredirem continuamente, iniciando por conceitos básicos e indo até os avançados, sempre utilizando elementos lúdicos popularmente reconhecidos.

¹² Disponível em: <https://studio.code.org/>



Figura 20. Plataforma Code.org.
Fonte: O autor (2024).

Khan Academy¹³ é uma plataforma online que oferece cursos de várias áreas do conhecimento, para alunos do Ensino Fundamental, Médio e até Superior. Os cursos são organizados em Áreas, conteúdos são divididos em Unidades. As lições são apresentadas em formato expositivo, em textos com imagens e também em forma de vídeo narrado.

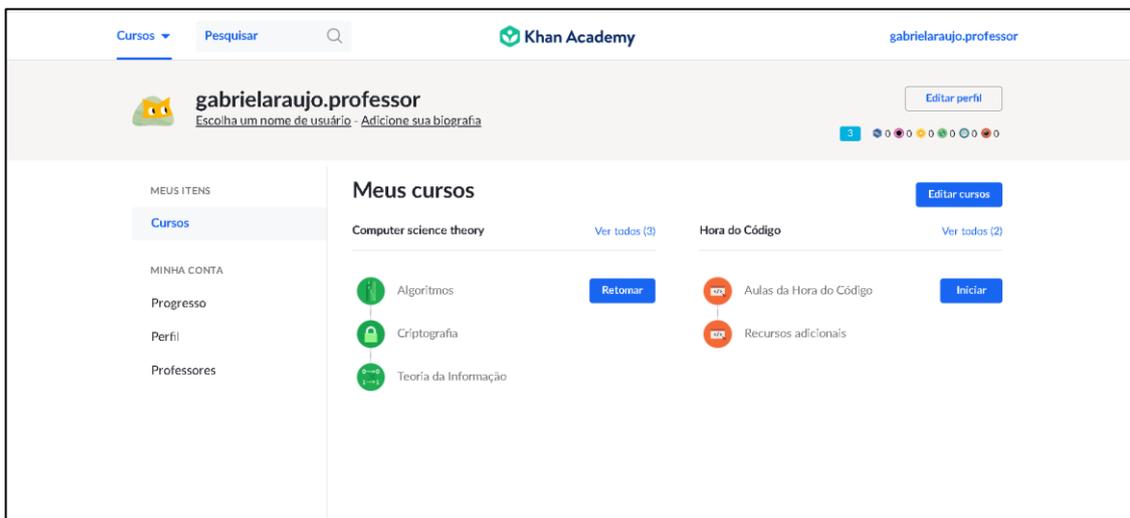


Figura 21. Plataforma Khan Academy.
Fonte: O autor (2024).

¹³ Disponível em: <https://pt.khanacademy.org/>

Tynker¹⁴ é uma plataforma de aprendizado de programação destinada a crianças e jovens. Ela foi projetada para tornar o ensino da programação mais acessível e envolvente, utilizando uma abordagem visual e interativa. A plataforma Tynker oferece uma variedade de recursos para ajudar os alunos a desenvolverem habilidades de programação e raciocínio lógico, além de cobrir uma ampla gama de tópicos, desde conceitos básicos de programação até temas mais avançados, como desenvolvimento de jogos e programação de drones.

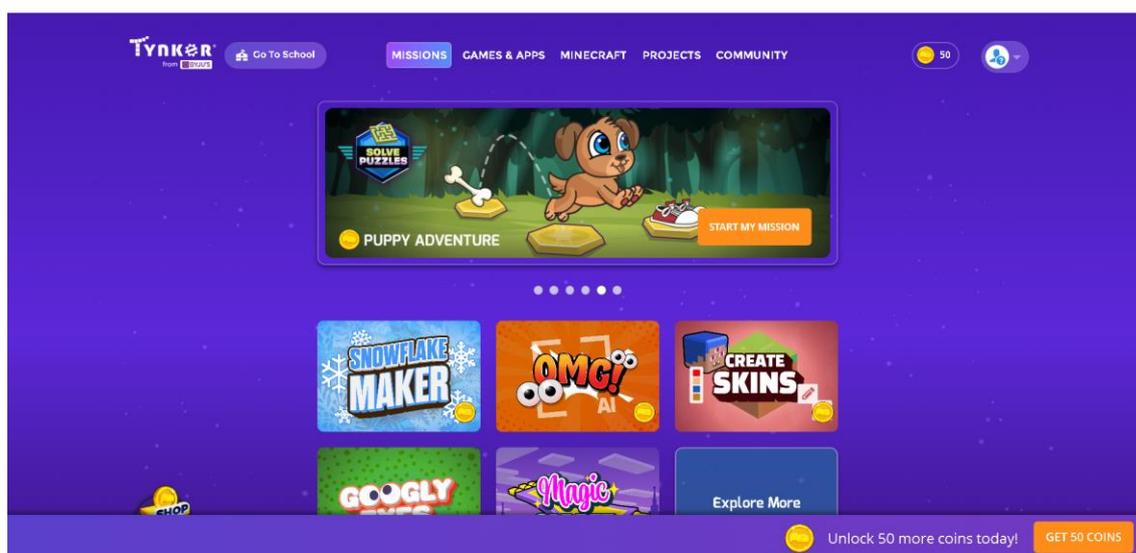


Figura 22. Plataforma Tynker.
Fonte: O autor (2024).

O Swift Playgrounds ¹⁵ é uma plataforma educacional desenvolvida pela Apple, projetada para ensinar programação usando a linguagem de programação Swift. Lançado em 2016, o Swift Playgrounds é destinado a iniciantes em programação, especialmente aqueles que desejam aprender a programar para iOS e macOS. Swift Playgrounds oferece uma série de desafios e lições estruturadas para ensinar conceitos fundamentais de programação, como loops, condicionais, funções e estruturas de dados. Para trabalhar nesta plataforma, é necessário um conhecimento um pouco mais avançado de codificação por linhas de comandos.

¹⁴ Disponível em: <https://www.tynker.com/>

¹⁵ Disponível em: <https://developer.apple.com/swift-playgrounds/>

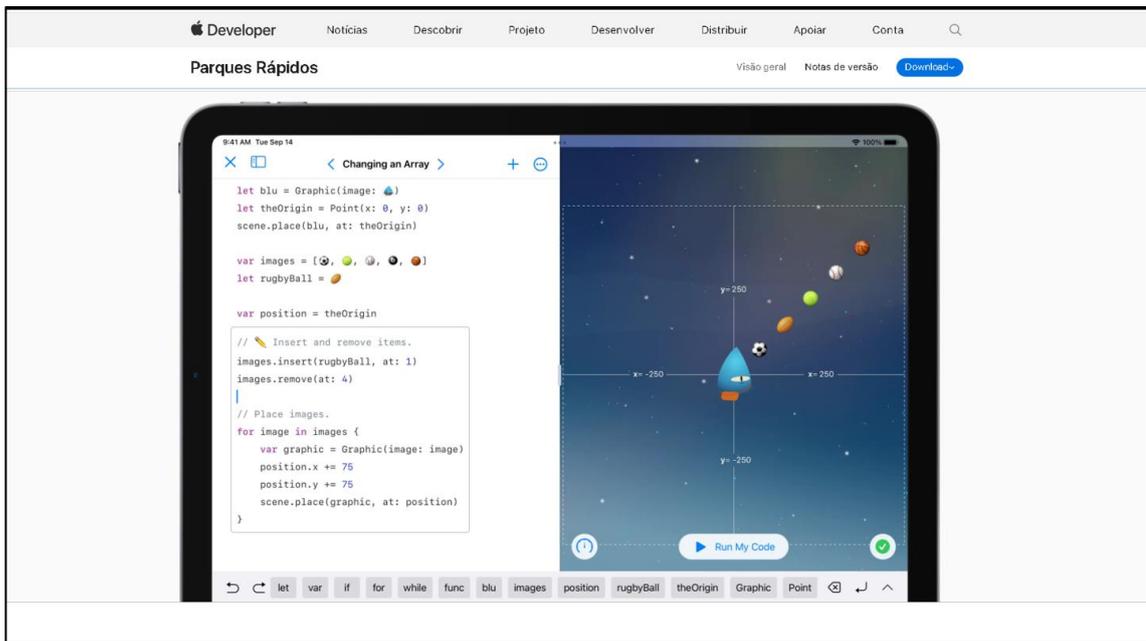


Figura 23. Plataforma Swift Playgrounds.
Fonte: O autor (2024).

Blockly Games¹⁶ é uma plataforma educacional e lúdica que utiliza a linguagem de programação visual Blockly para ensinar conceitos de programação de maneira interativa e divertida. O Blockly é uma linguagem de programação visual de blocos desenvolvida pelo Google que permite aos usuários criar código por meio da manipulação de blocos gráficos.

¹⁶ Disponível em: <https://blockly.games/>

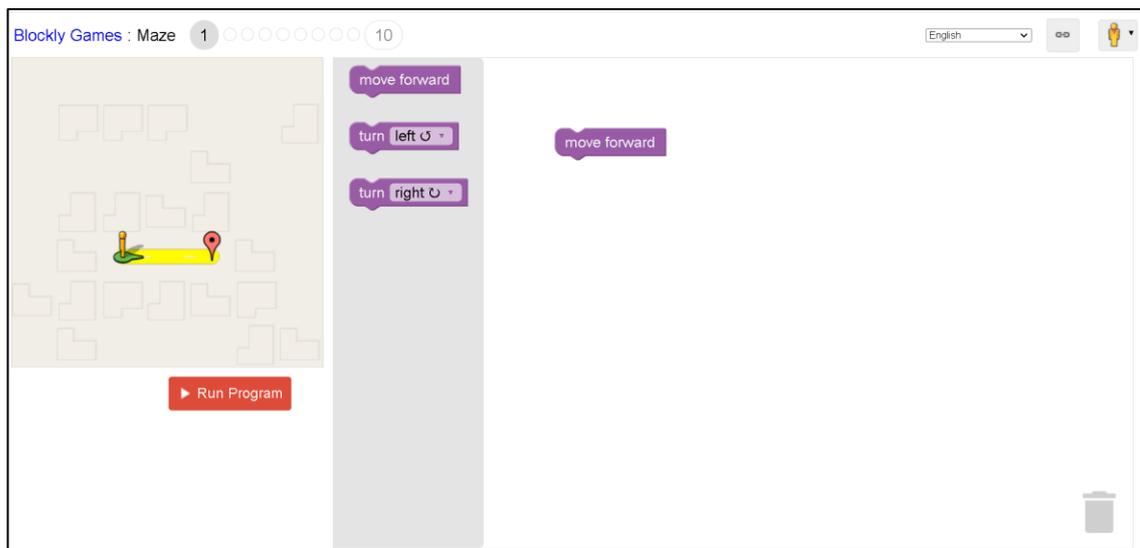


Figura 24. Plataforma Blockly Games.
Fonte: O autor (2024).

Por fim, foram classificados como **Plataformas de Desenvolvimento** os recursos didáticos digitais que possibilitavam que o aluno desenvolvesse uma solução, sejam jogos, animações ou outros, utilizando Algoritmo e Programação.

Scratch¹⁷ é uma linguagem de programação visual desenvolvida pelo MIT (Instituto de Tecnologia de Massachusetts) para facilitar a aprendizagem de programação, especialmente para iniciantes, crianças e jovens. Ele possui uma plataforma própria onde o aluno pode utilizar uma abordagem de blocos de arrastar e soltar, o que elimina a necessidade de escrever código manualmente. É uma ferramenta educacional poderosa que permite que usuários de todas as idades desenvolvam habilidades de pensamento lógico e criativo enquanto se divertem programando, podem desenvolver seu próprios jogos e animações.

¹⁷ Disponível em: <https://scratch.mit.edu/>

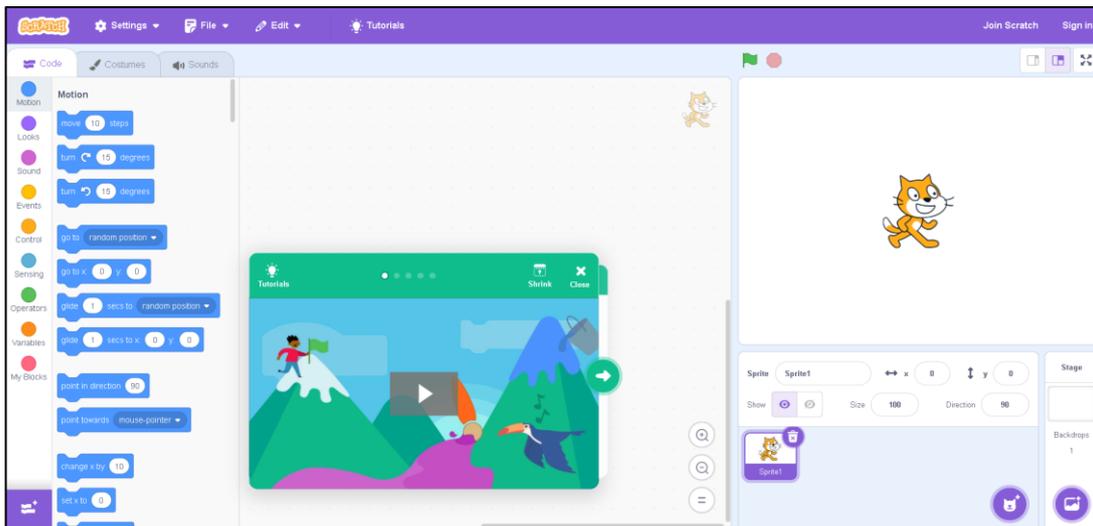


Figura 25. Plataforma Scratch.
Fonte: O autor (2024).

O Roblox¹⁸ é uma plataforma de criação e hospedagem de jogos online que permite aos usuários criar, compartilhar e jogar uma ampla variedade de jogos interativos. Lançado em 2006, o ROBLOX se destaca por sua abordagem única de permitir que os usuários construam seus próprios mundos virtuais usando a linguagem de programação Lua.



¹⁸ Disponível em: <https://create.roblox.com/>

Figura 26. Plataforma Roblox.
Fonte: O autor (2024).

O LEGO Mindstorms¹⁹ é uma linha de produtos da LEGO que combina peças de LEGO tradicionais com componentes eletrônicos e software programável, permitindo a criação e programação de robôs personalizados. O kit inclui tijolos especiais que contêm componentes eletrônicos, como processadores, portas de entrada/saída e conexões Bluetooth. Além disso, o kit possui sensores, como sensores de toque, sensores de cor, giroscópios e acelerômetros.

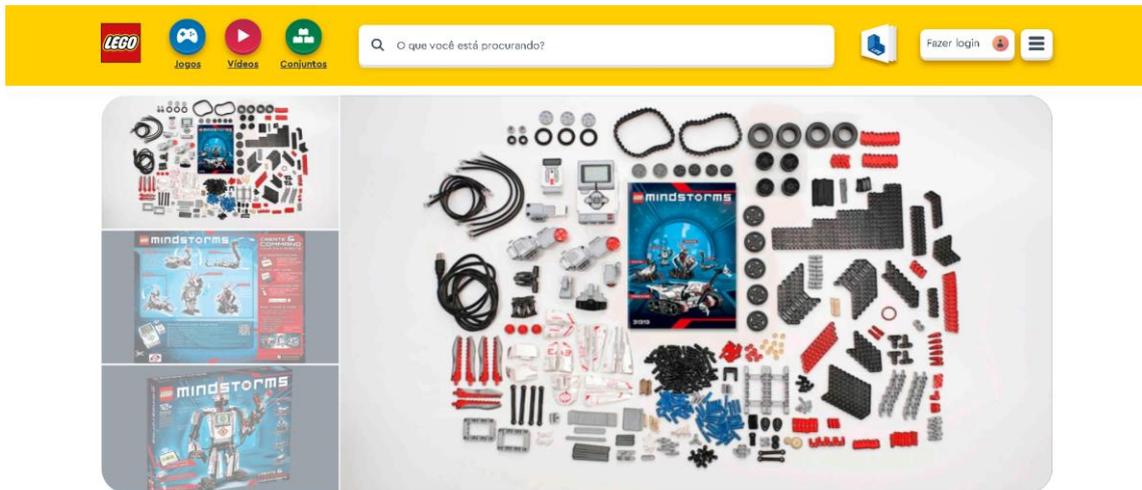


Figura 27. LEGO Mindstorms.
Fonte: www.lego.com.

3.1.1.6 Benchmarking.

O Benchmarking é uma abordagem que consiste em medir e comparar de forma contínua os processos empresariais de uma organização em relação aos líderes globais. Seu propósito é adquirir informações que possam orientar a organização na tomada de medidas para aprimorar seu desempenho (Madeira, 1999). De acordo com CIMA (1996), isso implica na criação de uma abordagem que envolve a coleta de dados, o estabelecimento de metas e pontos de referência que permitem identificar níveis específicos de desempenho. Ao adotar essas práticas identificadas, a melhoria do desempenho em áreas específicas da empresa é esperada.

Apesar do viés estratégico-econômico por trás do Benchmarking, este estudo utilizou esta estratégia com o objetivo de analisar as semelhanças e diferenças entre os recursos mais utilizados

¹⁹ Disponível em: <https://www.lego.com/pt-br/kids>

para o ensino de Algoritmo e Programação, identificados na Pesquisa Desk. Compreender as características distintivas de cada recurso é fundamental para identificar padrões, modelos de programação para crianças já consolidados, infraestrutura de equipamentos necessários. Esta análise visa fornecer informações valiosas para ajudar na ideação de novos recursos para o ensino de conceitos de Algoritmos e Programação para crianças, partindo de soluções já validadas.

Foram identificados 7 critérios de análise baseados em 7 conhecimentos a serem alcançados a partir da análise:

CRITÉRIOS	CONHECIMENTO
1. Plataforma	Conhecimento 1: A maioria das ferramentas é destinadas a qual público alvo?
2. Idade-alvo	Conhecimento 2: Qual o nível ou tipo de programação é mais utilizada?
3. Tipo de Programação	Conhecimento 3: Quais os recursos didáticos e de interação são usados para ensinar?
4. Métodos	Conhecimento 4: Quais os métodos são usados para ensinar programação para crianças?
5. Disponibilidade	Conhecimento 4: São digitais ou não digitais? Estão disponíveis para todos os dispositivos?
6. Recursos necessários	Conhecimento 5: Em caso de ferramenta digital, quais os recursos necessários para utilizar esta ferramenta?
7. Conceitos de Algoritmo e Programação abordados	Conhecimento 6: Quais os principais conteúdos ou conceitos de Algoritmo e Programação abordados nesses recursos?

Quadro 5. Critérios para Benchmarking.

Fonte: O autor (2024).

O Benchmarking foi realizado utilizando o motor de busca da Google. As plataformas que foram validadas para esta comparação foram identificadas na etapa de Pesquisa Bibliográfica e Pesquisa Desk. O quadro comparativo com os 7 critérios é apresentado a seguir (Quadro 6).

Plataforma	Idade-Alvo	Tipo de Programação	Método	Disponibilidade	Recursos Necessários	Conceitos de Algoritmo e Programação Abordados
Scratch	6-18 anos	Blocos Visuais	Amplas atividades, projetos, tutoriais	Online	Computador com navegador web	Sequência lógica, estruturas de controle, loops
ScratchJr Cards	5-7 anos	Blocos Visuais Físicos	Amplas atividades de criação de algoritmos	Offline	Kit de Cartões ScratchJr	Sequência lógica, estruturas de controle, loops

Code.org	4-18 anos	Blocos Visuais, Texto	Cursos abrangentes, jogos interativos	Online	Computador com navegador web	Lógica de programação, algoritmos simples
Tynker	5-18 anos	Blocos Visuais, Texto	Cursos, jogos, projetos, desafios	Online	Computador com navegador web	Pensamento computacional, estruturas de dados
Khan Academy	8-18 anos	Texto (JavaScript)	Lições de programação em JavaScript	Online	Computador com navegador web	Programação em JavaScript, algoritmos avançados
Roblox	10-18 anos	Lua (Scripting)	Criação de jogos, scripting em Lua	Online	Computador com cliente Roblox	Scripting, desenvolvimento de jogos
Minecraft Education	6-18 anos	Blocos Visuais, Texto	Lições de codificação, modificações	Plataforma de jogo	Computador com Minecraft Education Edition	Codificação em ambiente de jogo, automação
Lego Mindstorms	7-16 anos	Blocos Visuais	Construção de robôs, programação	Kits físicos e software	Kit Lego Mindstorms	Programação de robôs, sensores
Swift Playgrounds	12-18 anos	Swift (Textual)	Introdução à linguagem Swift	iOS App	iPad ou Mac compatível	Programação em Swift, desenvolvimento de aplicativos
Blockly	6-18 anos	Blocos Visuais	Usado em várias plataformas	Varia (online e offline)	Computador ou dispositivo móvel	Lógica de programação, algoritmos simples
Kodu Game Lab	8-15 anos	Blocos Visuais	Criação de jogos interativos	Software para PC	Computador com Kodu Game Lab	Desenvolvimento de jogos, sequência de comandos

Quadro 2. Resultado do Benchmarking.
Fonte: O autor (2024).

As técnicas utilizadas na fase de Imersão geram grande quantidade de dados. Para que estes dados auxiliem na identificação de oportunidades e desafios, precisam ser analisados e transformados em informações úteis. Por esse motivo, a etapa seguinte é a de Análise e Síntese, cujo propósito é organizar esses dados e transformá-los em informações, que por conseguinte, serão usados como insumos para a fase de Ideação.

3.1.2 Análise e síntese

A segunda fase do Design Thinking, como o nome sugere, tem por objetivo a análise e a síntese das informações coletadas. Nesta fase, os insights são organizados para identificar padrões de comportamentos, características e necessidades de modo a auxiliar na compreensão do problema (Viana, 2012).

Para auxiliar neste processo, usar-se-á algumas estratégias, tais como **Cartões de Insights**, **Diagrama de Afinidade** e **Personas**. Ao final desta fase, espera-se ter a definição do desafio central da pesquisa a ser alcançado no final pelo produto a ser concebido.

3.1.2.1 Cartões de Insights

Cartões de Insights é uma técnica para documentar os dados coletados na fase de Imersão. A justificativa por trás do uso desta técnica repousa sobre a facilidade que ele oferece para organizar os insights. Um insight é como um momento de “aha” – uma mudança repentina e consciente na representação de um estímulo, problema ou evento. Tal processo acontece após um processamento inconsciente e muitas das vezes parece estar desconectado do fluxo contínuo de pensamento consciente (Kaplan e Simon, 1990; van Steenburgh et al., 2012; Smith & Kounios, 1996). A MJV Innovation, empresa a qual Vianna – autor referência desta pesquisa – faz parte, resume que Insight “é a capacidade de ter um entendimento de alguma coisa ou algo, de forma intuitiva. é a capacidade de ter um entendimento de alguma coisa ou algo, de forma intuitiva (MJV Innovation, 2022).

O Cartão de Insights geralmente possui um título que representa o insight, uma explicação sobre o insight e a fonte original onde foi coletado. Neste estudo foi utilizado um modelo criado pela MJV Innovation, disponível em seu sítio oficial²⁰, contendo algumas informações adicionais, conforme apresentado na Figura 28, disponível no APÊNDICE D.

²⁰ Blog MJV Innovation: <https://www.mjvinnovation.com/library/>

CARTÃO DE INSIGHTS

Título _____	Número _____
Tema: _____	
Fato: _____ _____ _____ _____	
Fonte: _____ _____	

Figura 28. Cartão de Insights.
Fonte: Adaptado de MJV Innovation (2024).

Há diversas possibilidades para o uso dos Cartões de Insights. Nesta pesquisa, foram usadas duas estratégias. A primeira é durante a Pesquisa Desk. Sempre que uma questão relevante para o projeto é identificada, ela é registrada em um cartão, incluindo o achado principal, a fonte e uma explicação detalhada do assunto. A segunda possibilidade é durante a Pesquisa de Campo. Geralmente, os cartões são elaborados quando o pesquisador retorna ao local de trabalho e revisita o que foi visto e ouvido, registrando as questões que se destacaram durante a experiência.

Ao todo, foram gerados 31 cartões semelhantes ao da Figura 19, que foram digitalizados e agrupados no Diagrama de Afinidade.

CARTÃO DE INSIGHTS

Título OS ALUNOS GOSTAM DE EXPLORAR NOVAS POSSIBILIDADES	Número 2
Tema: HABILIDADES COM PORTAMENTAIS	
Fato: QUANDO UMA ATIVIDADE É PASSADA COM DIRETRIZES, OS ALUNOS TENTAM FAZER RAPIDAMENTE, VISANDO DO SOBRAO TEMPO PARA EXPLORAR NOVAS POSSIBILIDADES COM O RE- CURSO QUE ESTÃO TRABALHANDO.	
Fonte: OBSERVAÇÃO PARTICIPATIVA, CURSO I.	

Figura 29. Cartão de Insight 2 preenchido.
Fonte: O autor (2024).

3.1.2.2 Diagrama de Afinidade

Segundo Viana et. al., (2012), o Diagrama de Afinidade organiza e agrupa os Cartões de Insights com base em afinidade, similaridade, dependência ou proximidade, resultando em um diagrama que apresenta as áreas principais que abrangem o tema em questão, suas subdivisões e inter-relações. Esta técnica é empregada quando há uma grande quantidade de dados derivadas da fase de Imersão, visando identificar conexões entre temas e de áreas de oportunidade para o projeto. Nesta etapa busca-se identificar temas, subgrupos e/ou critérios que auxiliam no entendimento dos dados.

Como ferramenta para a construção do Diagrama de Afinidade, utilizou-se Figma²¹, plataforma online de prototipação e criação de diagrama, disponível gratuitamente.

²¹ Disponível em: <https://www.figma.com/>

O Diagrama foi estruturado a partir de 3 tópicos: Recursos Didáticos, que compreende os conhecimentos e insights construídos a respeito dos materiais e ferramentas utilizados para o ensino de programação e algoritmos; Comportamento dos Alunos, que compreende insights de como os alunos interagem com o aprendizado, suas reações e modos de engajamento; e, Metodologias de Ensino, que organiza insights sobre as abordagens e estratégias pedagógicas para ensinar conceitos de programação e pensamento computacional para crianças. O Diagrama de Afinidade, estruturado a partir dos supracitados tópicos e com seus respectivos Cartões de Insights é apresentado a seguir:

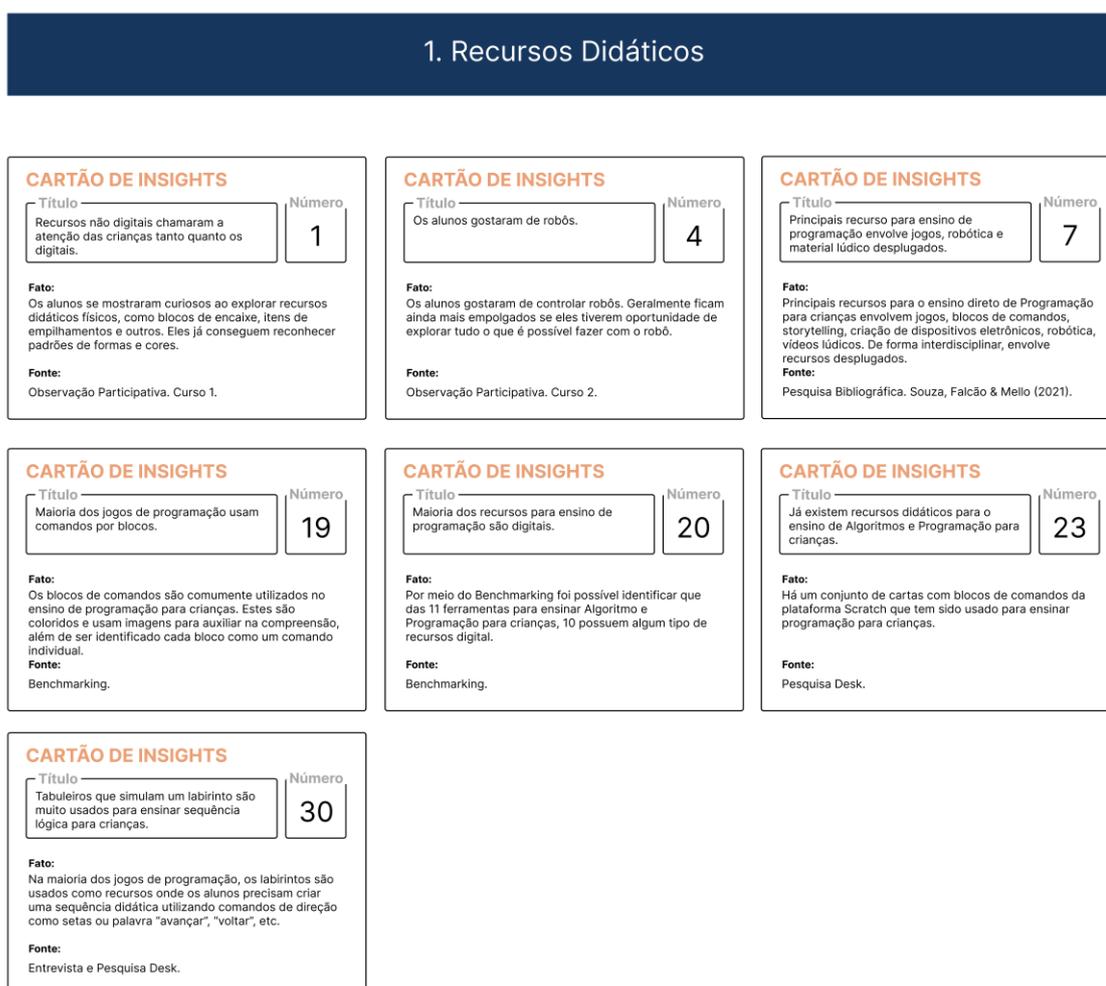


Figura 30 - 1. Recursos Didáticos (Parte 1 - Diagrama de Afinidade).
Fonte: O autor (2024).

2. Comportamentos dos Alunos

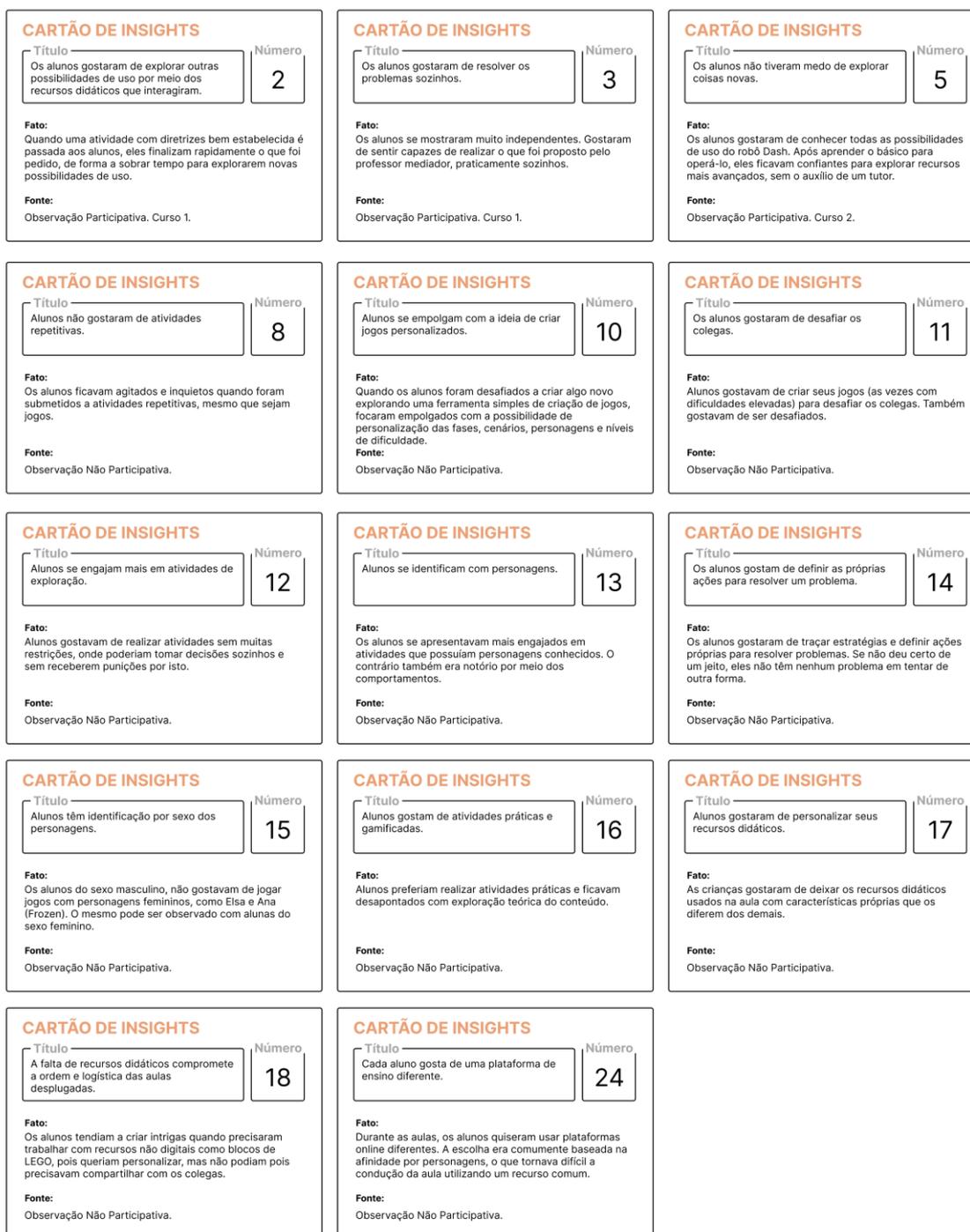


Figura 31. Comportamentos dos Alunos (Parte 2 - Diagrama de Afinidade).

3. Metodologias de Ensino

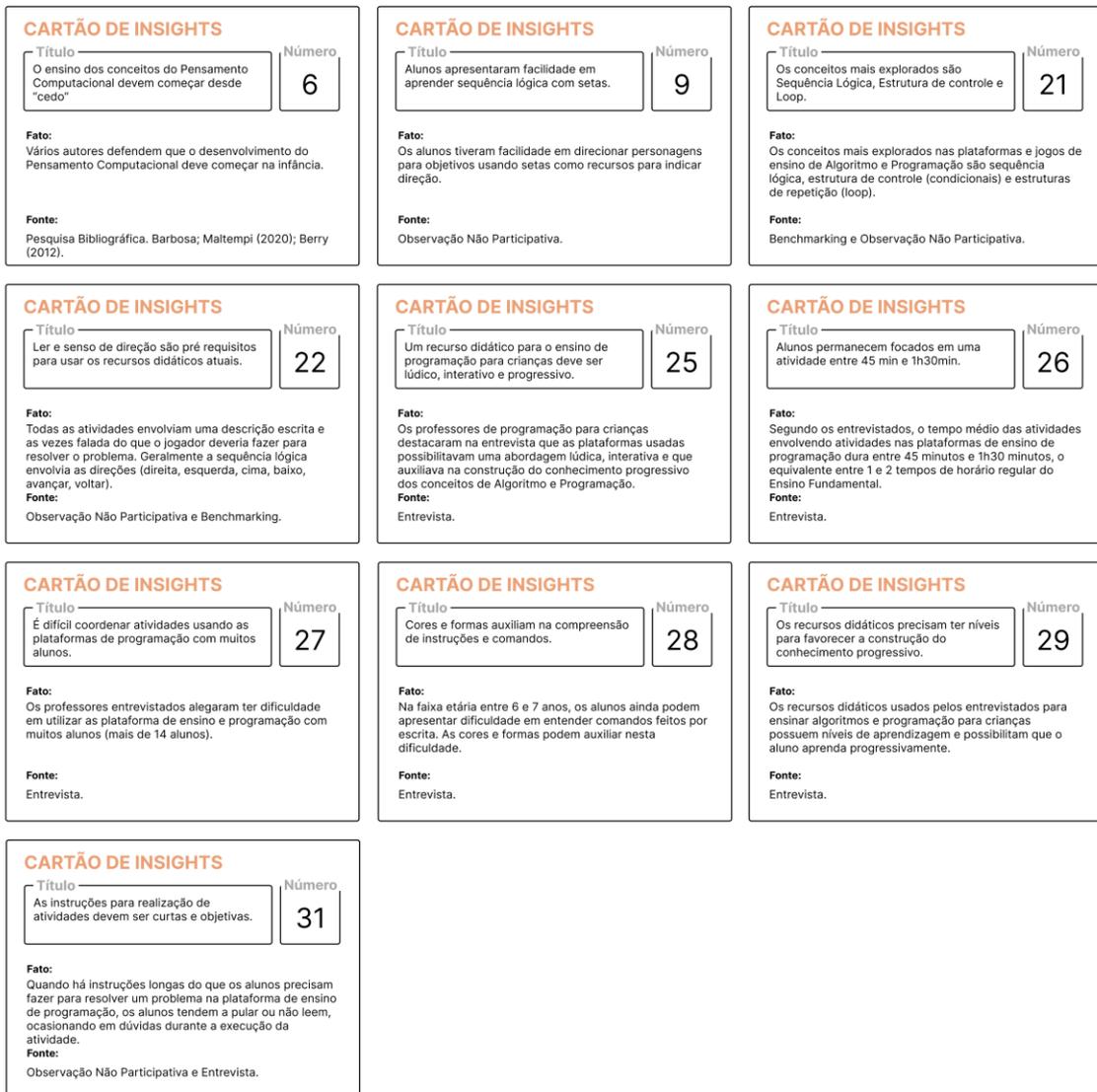


Figura 32. Metodologias de Ensino (Parte 3 - Diagrama de Afinidade).

Fonte: O autor (2024).

3.1.2.3 Personas

Personas são arquétipos baseados em uma síntese de comportamentos observados no público alvo de uma pesquisa. Elas encapsulam motivações, desejos, expectativas e necessidades, refletindo

características importantes de um segmento mais amplo de usuários. Essas representações são valiosas em diversas fases do processo de design e desenvolvimento, sendo particularmente benéficas para a geração e validação de ideias. Ao incorporar as necessidades das personas durante a fase de ideação, é possível criar soluções inovadoras mais alinhadas às demandas dos usuários. Posteriormente, as mesmas soluções são revisadas sob a perspectiva das personas para refinar e escolher as opções mais promissoras. As personas direcionam o foco do design para as experiências do usuário, ajudando na interpretação das informações e facilitando decisões mais informadas. São construídas a partir de dados reais e variam em características demográficas e comportamentais, resultando em um grupo diversificado que representa os extremos do espectro de usuários (Viana et. al., 2012).

Existem diversos layouts de personas validadas na bibliografia. Cada persona é detalhadamente descrita com nome, história e necessidades específicas, tornando-se um arquétipo tangível para referência ao longo do desenvolvimento do produto ou serviço. Para esta pesquisa, será usado o layout de Jaime Levy, construído e disponibilizado em seu livro UX Strategy (Levy, 2015).

Para Levy, uma persona deve ser composta de 4 partes:

1. Nome e imagem: Descreve o nome de um usuário; usa-se fotos para representar como ele é; em caso de ser caracterizado por um sexo, e uma idade, a foto deve ser represar tais características.

2. Descrição: Descreve os dados demográficos e apresentação de características gerais que são ligadas ao problema em questão.

3. Comportamentos: Apresenta como o usuário tem tentado resolver o problema atualmente; quais soluções alternativas têm encontrado para satisfazer necessidades próprias sem a solução que está sendo desenvolvida; o problema é realmente relevante ao ponto de influenciar o comportamento dos usuários.

4. Necessidades e Objetivos: Descreve porque o usuário age da maneira descrita nos comportamentos; o que o usuário deseja alcançar; quais são suas metas; quais necessidades não estão sendo satisfeitos; quais situações lhe impedem de alcançar seus objetivos.

Nesta pesquisa, houve a necessidade de construir 2 personas, tendo em vista os dois tipos de usuários afetados pelo problema em questão: Professor e Aluno.

Professor de Programação para Ensino Fundamental 1



Descrição

- Ana Carolina.
- Pessoa casada na faixa dos 37 anos.
- Mora em Manaus.
- Classe média.
- Trabalha em horário comercial, 5 dias/semana.

Comportamentos

- Adora dedicar momento para si após o trabalho.
- Não leva trabalho para casa.
- Prefere realizar atividades que ofereçam praticidade.
- Planeja e separa recursos para as aulas semanalmente.
- Utiliza recursos online para ensinar programação.

Necessidades & Objetivos

- Busca constantemente por atividades que engaje todos os alunos da sala mas tem dificuldade em encontrá-los.
- Deseja um recurso didático que sejam simples e práticos para crianças.
- Busca minimizar o tempo e esforço empregado na construção de recurso didático lúdico para o ensino de conceitos de programação para crianças.

Figura 33 - Persona Professora.

Fonte: O autor (2024).

Aluno de Ensino Fundamental 1 que deseja aprender programação



Descrição

- Chloe Araújo.
- Criança, na faixa dos 6 anos.
- Estuda em escola particular no período da manhã.
- Pratica aulas de dança no tempo livre.

Comportamentos

- Gosta de jogar jogos de aventura e exploração.
- Frequenta eventos de tecnologia com seu pai.
- Gosta de resolver problemas comuns do dia a dia sozinha.
- Gosta de se caracterizar de Unicórnio.

Necessidades & Objetivos

- Gostaria de criar um jogo mas não sabe como começar.
- Assiste vídeos de Introdução a Programação no YouTube mas não entende nada.
- Gosta de ver robôs se mexendo mas não faz ideia de como eles funcionam.

Figura 34. Persona Aluna.

Fonte: O autor (2024).

Apesar de possuir autores que levantam questionamento sobre a eficiência do uso de Persona como recurso metodológico, os mesmos atestam a eficácia desta técnica para reunir em um arquétipo, todos os dados empíricos levantados em pesquisa de campo (Chapman & Milham, 2006).

Ao concluir a fase de Análise e Síntese, é possível levantar duas hipóteses para o problema em questão:

1 – Os alunos buscam solucionar os problemas de forma particular, onde situações lineares e previsíveis comprometem o engajamento na proposta de solução.

2 – Recursos didáticos não digitais e personalizáveis contribuem para o engajamento das atividades lúdicas em sala de aula.

Visando testar tais hipóteses, O método escolhido para proporcionar a base investigativa desta pesquisa é o hipotético-dedutivo, uma vez que, conforme afirma Gil (2008, p. 30):

O cientista, através de uma combinação de observações cuidadosas, alcança um conjunto de postulados que governam os fenômenos pelos quais está ingressado, daí deduz ele as consequências por meio de experimentação e, dessa maneira, refuta os postulados, substituindo-os quando necessário por outros e assim prossegue.

Desta forma, a fase seguinte visa levantar possíveis soluções e testá-las, buscando satisfazer as hipóteses levantadas.

3.1.3 Ideação

A fase de Imersão é a fase onde são geradas ideias inovadoras que estejam de acordo com as necessidades dos usuários. Nesta fase, é salutar que a equipe que realiza o projeto de concepção de produto seja multidisciplinar. Além disto, busca-se envolver outros membros como usuários e profissionais de áreas que sejam convenientes ao tema em estudo, buscando reunir diferentes expertises, de modo a contribuir com diferentes perspectivas, tornando o resultado final consequentemente mais rico e assertivo (Viana et. al., 2012).

A fase de Imersão se iniciou com um Brainstorming, uma técnica projetada para promover a criação de muitas ideias em um intervalo de tempo reduzido, normalmente conduzido em um ambiente coletivo. Após o Brainstorming, foram realizadas dois Workshops de Cocriação, reuniões ou encontros estruturados como uma sequência de exercícios colaborativos que visa incentivar o pensamento criativo e o trabalho em equipe, com a finalidade de gerar soluções criativas e inovadoras.

3.1.3.1 Atividade e Brainstorming

Para testar a Hipótese 1 - Os alunos buscam solucionar os problemas de forma particular, onde situações lineares e previsíveis comprometem o engajamento na proposta de solução – os alunos foram envolvidos em uma atividade que seria sucedida por um Brainstorming.

A atividade consistia em reproduzir a ideia de controlar um robô real, fora do computador. Para esta atividade, foi necessário transformar o Laboratório de Inovações em um grande labirinto, semelhante as atividades realizadas muitas vezes na plataforma Hora do Código. Foram necessários em torno de 7 horas entre o planejamento da atividade, a busca por materiais pela escola para confecção do labirinto, preparação da sala de aula para se tornar um ambiente imersivo.

Os alunos foram divididos em equipes e teriam que simular o controle de um robô por meio do comando de voz. Um da equipe foi sorteado para ser vendado para simular um robô real, que nada pode fazer senão obedecer a comandos e instruções dados por um humano. As regras definidas somente permitiam comandos como Avançar, Voltar, Girar à Direita, Girar à Esquerda e Parar. Não era permitido tocar no robô.



Figura 35. Atividade Imersiva.
Fonte: O autor (2024).

Objetivo da imersão foi simular novos modelos de ensino e aprendizagem e identificar a aderência dos alunos objetos do estudo a recursos metodológicos desplugados – sem o uso de elementos eletrônicos.

Após a atividade de imersão, os alunos foram reunidos em uma roda de conversas onde foi realizado o Brainstorming com o objetivo de identificar a satisfação dos usuários principalmente em relação ao tipo de atividade desplugada. Os feedbacks obtidos nesta etapa, geraram outros Cartões de Insights que foram reagrupados no Diagrama de Afinidade, apresentados no tópico 3.1.2.2.

3.1.3.2 Workshops de Cocriação

Para validar a segunda hipótese - Recursos didáticos não digitais e personalizáveis contribuem para o engajamento das atividades lúdicas em sala de aula – foram idealizados dois Workshops de Cocriação, realizada no Laboratório de Inovações do Colégio Connexus, no turno da manhã e posteriormente a tarde e contou com a presença de todos os alunos da turma.

Workshop de Cocriação 1

O Workshop de Cocriação 1 foi realizado com uma turma do 1º ano do Ensino Fundamental I, no turno Matutino e posteriormente, no turno Vespertino e dividido em 3 momentos. No primeiro momento, de forma desplugada, foi usado um tabuleiro confeccionado pelos professores para auxiliar na aprendizagem de conceitos de Algoritmos e Programação, por meio de sequência lógica. Foram utilizadas algumas peças de LEGO coloridas para definir objetivos a serem alcançados por meio da sequência.

As Peças Vermelhas foram usadas como bloqueios, por onde a sequência não podia passar, com o objetivo de limitar as possibilidades e favorecer o pensamento lógico e a tomada de decisão. O ponto de partida era marcado por um *post-it* escrito “GO” e as sequências lógicas eram construídas por meio de setas desenhadas, também em *post-it*, e postas sobre o tabuleiro.

No segundo momento os alunos foram os protagonistas. Eles fizeram desafios uns aos outros, movendo as peças de objetivo (coloridas) e as de bloqueio (vermelhas). Notou-se nessa segunda etapa que os alunos começaram a perceber e diferenciar os desafios mais complexos dos mais simples e desafiavam uns aos outros sobre as diferentes formas de dificultar a sequência para intensificar o desafio.

O terceiro momento foi de uso na plataforma Hora do Código, Curso 1, Atividade 4 como recurso plugado, no qual os alunos deveriam arrastar blocos de comandos e construir uma sequência do Angry Bird até o Porco Verde.

Parte do material empregado precisou ser confeccionado pelo pesquisador com ajuda da professora titular, e outra foi adaptada do conjunto de blocos da empresa LEGO.

Material confeccionado:

Tabuleiro: Para confeccionar o tabuleiro foi usado cartolina, papel cartão, pincel. O tabuleiro linhas na horizontal e na vertical que se cruzam entre si, formando os espaços onde serão organizados os demais elementos da atividade.



Figura 36. Material usado na confecção do Tabuleiro.
Fonte: O autor (2024).

Cartas de direção: As cartas de direção foram feitas com papel A4 Branco e Pincel. As cartas representam setas que podem ser posicionadas nas direções Norte, Sul, Leste e Oeste. As setas são usadas pelos alunos para construir o caminho (sequência lógica) para chegar a um objetivo.

Carta de inicialização da sequência: A carta de inicialização possui a palavra “GO” escrita no seu interior, em cor Verde. Ela define o ponto de partida da sequência em qualquer posição do Tabuleiro. Quanto mais longe a carta de inicialização da sequência estiver do objetivo, maior será a sequência a ser implementada pelo aluno.

Material adaptados:

LEGOs Objetivos: Os LEGOs de Objetivos determinavam onde os alunos deveriam chegar por meio da sequência lógica. Eram representadas por peças Coloridas e poderiam ser reposicionadas a qualquer momento visando aumentar ou diminuir a dificuldade para cada aluno individualmente.

LEGOs Bloqueio: Os LEGOs de Bloqueio ou Empecilhos determinavam áreas no tabuleiro pelas quais o aluno não poderia construir a sequência e eram representadas por peças Vermelhas. Serviam como bloqueio e aumentavam a dificuldade dependendo da posição onde a carta de inicialização foi colocada no tabuleiro.



Figura 37. Execução do Workshop de Cocriação 1
Fonte: O autor (2024).

Vale destacar que após o Workshop de Cocriação 1, faltando cerca de 10 minutos para fim do tempo de aula, os alunos precisaram se dirigir aos computadores para realizar atividades da plataforma Hora do Código. Neste momento foi identificadas frases de insatisfação e descontentamento, tais como “De novo Hora do Código?” e “Deixa a gente ficar jogando aqui, professora!”, que demonstram uma desmotivação na realização desta atividade proposta, realizada repetidamente em aulas anteriores.

Workshop de Cocriação 2

O Workshop de Cocriação 2 também consistia em trabalhar conceitos de Algoritmos e Programação, por meio de Sequência Lógica. Foi dividido em 3 momentos. O primeiro permitiu que os próprios alunos confeccionassem seus elementos para atuar no tabuleiro, assim como foram usados os LEGOs no primeiro experimento. Foram distribuídos 4 post-it para cada aluno. Os alunos deveriam escolher um personagem a qual se identificassem e desenharem no 1º *post-it*.

Após isto, os alunos desenharam no 2º *post-it* um elemento do qual o personagem precisasse, algo importante para o personagem. Este foi usado como objetivo da Sequência Lógica, tais como os LEGOs coloridos no Experimento 1. Nos últimos 2 *post-its* os alunos desenharam elementos que impediam o personagem de chegar ao objetivo. Estes elementos foram os elementos de bloqueio de onde a sequência deveria se desviar, como os LEGOs Vermelhos do Experimento I.

No segundo momento, os alunos construíram uma narrativa estruturada em "Contexto - Problema - Solução e Recompensa", onde explicaram individualmente para a turma qual o contexto do seu personagem, qual o problema ele enfrentaria pelo tabuleiro, relataram a história por trás do surgimento deles, ressaltaram detalhes que ajudavam os demais alunos a entender o porquê existia aquele objetivo e aqueles empecilhos e qual seria o prêmio pela correta execução da Sequência Lógica.

No terceiro momento, os alunos formaram duplas para interagir entre si. Nesse momento cada integrante da dupla propôs um desafio de Sequência Lógica ao outro integrante. O integrante desafiado deveria lembrar da narrativa contada para identificar qual era o objetivo do personagem e quais eram os empecilhos que poderiam o impedir de alcançá-lo (Figura 33).

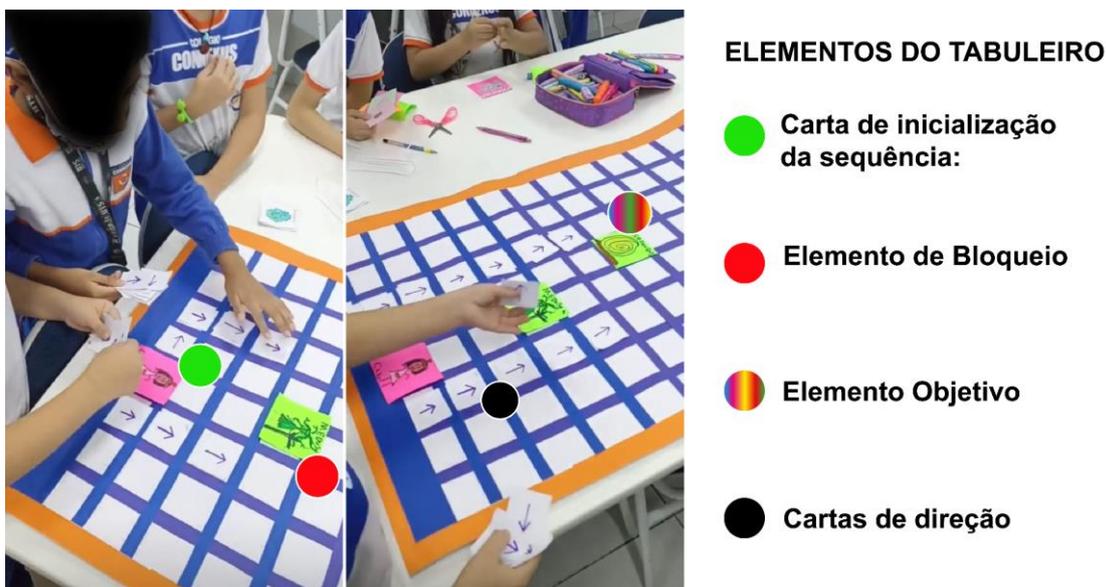


Figura 38. Execução do Workshop de Cocriação 2
Fonte: O autor (2024).

Os resultados destas aplicações evidenciaram aspectos importantes do processo de ensino e aprendizagem. Os alunos demonstraram-se engajados nas atividades que aplicavam conceitos de sequência lógica por meio da Computação Desplugada. Acredita-se que esse fato está relacionado a padronização e delimitação dos elementos lúdicos pelos próprios alunos, o que afirma a hipótese 2.

Além disso, o fato de os alunos estarem completamente no controle da atividade, definindo as dificuldades das sequências lógicas e desafiando os seus colegas, contribuiu para que a atividade fosse realizada por meio do aprendizado mútuo entre os próprios alunos, com mínima interferência docente.

Sendo assim, após a observação destes fatos, o método indutivo permite inferir que as atividades que trabalharam com Computação Desplugada e com elementos lúdicos do contexto sociocultural dos alunos tiveram maiores engajamentos.

3.1.3.3 Matriz de Posicionamento

A Matriz de Posicionamento é um instrumento usado para examinar estrategicamente as ideias propostas, empregando-a para validar tais ideias frente aos **Critérios Norteadores** e atender às necessidades específicas das Personas estabelecidas no projeto. Os Critérios Norteadores - CN sintetizam requisitos do produto a ser prototipado na fase de Prototipação.

É comumente empregada para fornecer suporte ao processo de tomada de decisão, comunicando claramente os prós e contras de cada alternativa proposta, permitindo assim a seleção das ideias mais alinhadas com a estratégia para desenvolvimento e prototipagem.

Os 7 Critérios Norteadores estabelecidos são listados abaixo:

CN01: Possibilitar multijogadores.

CN02: Possibilitar personalização de elementos lúdicos.

CN03: Possibilitar tomada de decisões ilimitadas sem perdas.

CN04: Possibilitar elementos lúdicos como cenários, personagens e outros baseado na fauna e flora amazônica.

CN05: Possibilitar o aprendizado de Sequência Lógica

CN06: Possibilitar o aprendizado de Estrutura de Controle.

CN07: Possibilitar o aprendizado de Estrutura de Decisão.

Foram ainda geradas 3 Ideias, as mesmas podem ser resumidas no quadro abaixo:

Quadro de Ideias			
	Labirinto 3D	Tabuleiro de Labirinto	Conjunto de Cartas
Descrição	Kit de personalizados para construção de ambiente imersivo que possibilite a transformações de espaços como a sala de aula em um Labirinto.	O "Tabuleiro de Labirinto" é um jogo de estratégia e lógica, onde os jogadores constroem e navegam por um labirinto em busca de alcançar um objetivo. O foco é criar e solucionar desafios lógicos, representando o personagem que deve chegar a um destino final, atravessando um	Conjunto de cartas para ensinar conceitos fundamentais de programação, como sequência lógica, estruturas de controle e repetição. O objetivo é familiarizar os jogadores com o pensamento lógico e estruturado, essencial para programação e resolução de

		percurso cheio de obstáculos e desvios inteligentes.	problemas, através de um método divertido e interativo.
Recurso lúdico	<ul style="list-style-type: none"> • Fitas para demarcação de limites e construção do Labirinto no chão. • Máscaras em papéis brancos com elástico, a serem personalizadas com rosto de Robôs pelos próprios jogadores. • Conjunto de Pincéis apagáveis para desenho e personalização da máscara. • Cartas em tamanho A4 contendo objetivos a serem alcançados dentro do Labirinto. • Cada equipe deve personalizar a face do robô na máscara, utilizando o conjunto de pincéis. 	<ul style="list-style-type: none"> • Tabuleiro Modular: Um tabuleiro dividido em seções que podem ser reorganizadas para formar diferentes configurações de labirinto. • Peças de Caminho: Diversas peças que representam diferentes tipos de caminho (reta, curva, bifurcação) que os jogadores podem usar para construir o labirinto. • Peças de Personagem: Representações dos jogadores no tabuleiro, que podem variar em forma e cor. • Cartas de Ação/Objetivo: Cartas que adicionam elementos de surpresa ao jogo, desafiando os jogadores a adaptarem suas estratégias. 	<ul style="list-style-type: none"> • Cartas de Instrução: Cartas que representam comandos básicos de programação, como 'mover para frente', 'virar à direita', entre outros. • Cartas de Controle: Incluem instruções de controle de fluxo, como 'se-então', 'enquanto', ou 'repetir'. • Cartas de Desafio: Apresentam problemas ou quebra-cabeças que os jogadores devem resolver usando uma combinação das cartas de instrução e controle. Exemplo: Chegue ao objetivo utilizando melhor quantidade de instruções (Cartas). É este objetivo que deve ser montado no Painel Depurador de forma que os alunos visualizem de forma lúdica o Desafio Proposto. • Painel Depurador: Um painel em material de velcro em forma de matriz 24x24, na qual o professor pode

			<p>apresentar as ações em tempo real, baseado nas cartas usadas para criar a solução. Será usado para fazer o Teste de Mesa (Dry run), que se trata de uma forma de acompanhar o funcionamento de um algoritmo instrução a instrução podendo validar sua lógica (Batista, 2013).</p> <ul style="list-style-type: none"> Os elementos lúdicos que serão apresentados como personagens, animais, objetos, podem ser confeccionados por meio de desenho e recorte em uma folha branca de velcro a ser disponibilizada junto com o Painel Depurador, permitindo que o professor mediador personalize os elementos lúdicos para o contexto de seus alunos.
Jogabilidade	<p>Preparação: Divide-se os participantes em equipes de número igual. As equipes devem vender o robô do adversário, afim de certificarem-se de que o mesmo não usará a própria visão para realizar o percurso do labirinto. O objetivo é</p>	<p>Preparação: Os jogadores começam montando o tabuleiro, organizando as peças de caminho para criar um labirinto inicial. Em seguida, cada jogador coloca sua peça de personagem no ponto de partida. São</p>	<p>Preparação: No painel depurador, utilizando vários elementos como objetos lúdicos (casinhas, animais, personagens), o professor mediador apresenta o desafio a ser alcançado por meio das cartas. Os alunos, após terem</p>

	<p>alcançar a carta final de objetivo, posicionada em alguma parte do labirinto que somente os jogadores que não estão vendados conseguem ver.</p> <p>Forma de Jogar: As equipes devem conduzir o robô por meio do comando de voz.</p> <p>As atividades podem ser feitas e repetidas, inúmeras vezes.</p> <p>Condição de Vitória: Podem existir 2 modos de jogo. O primeiro, é baseado na velocidade e ganha a equipe que realizar o percurso em menor tempo. O segundo, são inseridos mais de um objetivo no Labirinto e é cronometrado um tempo padrão para as duas equipes realizarem o percurso. Quem atingir o maior número de pontos que podem ser atribuídos individualmente a cada objetivo estabelecido, dentro do tempo estimado, ganha.</p>	<p>divididas 6 Cartas de Ações (Avançar, Voltar, Virar à Esquerda, Virar a Direita) iniciais para cada jogador, empilhando as demais cartas no centro da mesa.</p> <p>Além das cartas de ações existem as Cartas de Repetição (Repita 2x, Repita 4x e Repita 6x), que permite o usuário combinar cartas de ações e repeti-las 2, 4 e 6 vezes, dependendo da quantidade de vezes especificadas na Carta de Repetição.</p> <p>Existe ainda as Cartas de Modificação, que servem como um recurso para alterar o labirinto visando facilitar a estratégia do jogador que a usa.</p> <p>Forma de Jogar: Os jogadores se revezam para mover suas peças de personagens através do labirinto, usando Cartas de Ação (Avançar, Voltar, Virar à Esquerda, Virar a Direita), sendo a quantidade de ações determinadas pela rolagem de dados. Os jogadores podem também modificar partes do labirinto durante o jogo, adicionando um elemento de estratégia dinâmica, visando potencializar sua estratégia lógica.</p> <p>Condição de Vitória: O primeiro jogador a navegar com sucesso através do labirinto e alcançar o objetivo</p>	<p>sido apresentado a cada conceito por trás dos comandos, apresentam soluções de forma colaborativa, inferindo quais cartas devem ser usadas e em qual sequência lógica devem ser estruturadas para atingir o objetivo proposto.</p> <p>O professor mediador utiliza do Painel Depurador para ir apresentando em tempo real o que as Cartas de Instrução, Controle apresentadas pelos alunos como sequência lógica. As mudanças na sequência lógica podem ser feitas várias vezes até que a sequência lógica seja suficiente para alcançar o objetivo.</p> <p>Conforme as cartas da sequência vão sendo solicitadas, o professor representa a ação no Painel Depurador, simulando a execução da instrução em tempo real.</p>
--	--	---	---

		final é o vencedor. Alternativamente, o jogo pode terminar quando um jogador alcança um número específico de objetivos ou coleta itens específicos. A regra de jogo deve ser estabelecida antes do início da partida.	
Habilidades e Competências desenvolvidas	O jogo promove a habilidades de Sequência Lógica e Estrutura de Repetição. Também possibilita o desenvolvimento de habilidades de colaboração e comunicação precisa.	O jogo promove habilidades de raciocínio lógico e estratégico, planejamento, e tomada de decisão. Ele incentiva os jogadores a pensar antecipadamente, prever movimentos dos oponentes e adaptar estratégias em resposta a mudanças inesperadas no tabuleiro.	O jogo permite que a construção da sequência seja feita por meio da tentativa e erro, possibilitando que o aluno perceba que o conhecimento a respeito de Algoritmos e Programação é construído sobre este princípio e que ele não sofrerá nenhuma punição pelo erro, mas que a tentativa e erro é natural e importante no processo de aprendizado dos conceitos de Sequência Lógica, Estrutura de Controle e Estrutura de Repetição.
Tempo especulado	40 min a 60 min.	30min a 60 min.	Indefinido.

Quadro 3. Quadro de Ideias.
Fonte. O autor (2024).

A Matriz de Posicionamento é construída listando os Critérios Norteadores na primeira coluna, bem como as Ideias originadas são listadas nas colunas consecutivas, formando uma matriz, que é preenchida, avaliando o atendimento de cada ideia aos requisitos estabelecidos (Viana et. al., 2012).

Matriz de Posicionamento			
Critérios	Ideias		
	Labirinto 3D	Tabuleiro de Labirinto	Conjunto de Cartas
1. Possibilitar multijogadores.	SIM	SIM	SIM
2. Possibilitar personalização de elementos lúdicos.	SIM	NÃO	SIM

3. Possibilitar tomada de decisão ilimitadas e sem perdas	NÃO	NÃO	SIM
4. Possibilitar elementos lúdicos como cenários, personagens e outros, baseados na fauna e flora amazônica.	SIM	SIM	SIM
5. Favorecer o aprendizado de Sequência Lógica.	SIM	SIM	SIM
6. Favorecer o aprendizado de Estrutura de Controle.	NÃO	SIM	SIM
7. Favorecer o aprendizado de Estrutura de Repetição.	NÃO	SIM	SIM

Quadro 4. Matriz de Posicionamento.
Fonte: O autor (2024).

Ao analisar a Matriz de Posicionamento, é possível notar que as ideias do Tabuleiro de Labirinto e Conjunto de Cartas atenderam 5/7 e 7/7 dos Critérios Norteadores, respectivamente, enquanto que a ideia do Labirinto 3D atendeu somente 4/7 dos critérios. Portanto, a fase de Prototipação priorizará a construção de protótipos baseados nas duas ideias que atenderam a maior quantidade de critérios, inferindo esforços para integrar elementos de ambas as ideias em uma única solução que a partir de agora, será considerado como um jogo.

3.1.4 Prototipação

A fase de Prototipação é um estágio crucial no desenvolvimento de qualquer produto, representando a transição do conceitual para o tangível. É o processo de transformar ideias em modelos físicos ou digitais que podem ser testados e refinados. Esta fase é particularmente crítica em projetos que envolvem soluções lúdicas, como jogos e atividades educativas, pois permite que conceitos abstratos de aprendizagem e engajamento sejam explorados e materializados de forma prática.

No contexto de desenvolvimento de soluções lúdicas, a prototipação não apenas traduz ideias em realidades tangíveis, mas também incorpora todo o conhecimento adquirido nas etapas anteriores de Imersão, Análise e Síntese, e Ideação. Compreender o público-alvo, os objetivos educacionais e as dinâmicas de jogabilidade são aspectos que influenciam diretamente a forma e a função do protótipo desenvolvido.

A fase de Prototipação foi subdividida em duas etapas – 1. Definição dos objetivos de aprendizagem; 2. Definição e design dos elementos lúdicos - estruturadas para orientar o desenvolvimento de maneira eficiente e eficaz. Cada etapa é delineada com o propósito de refinar o jogo lúdico, garantindo que o produto final seja não apenas funcional, mas também engajador e educativo.

Como material, utilizou-se folhas de papel A4, papelão, cola, tesoura, réguas, canetas esferográficas, lápis, borracha, lápis de colorir e algumas poucas informações foram impressas para agilizar o processo de confecção dos esboços mais ricos em detalhes.

Para a elaboração de artes em alta resolução e a vetorização de rascunhos feitos à mão, o software escolhido foi o Adobe Illustrator 2021, reconhecido por sua excelência em gráficos vetoriais. Este programa faz parte do renomado pacote Adobe Creative Cloud e é uma ferramenta favorita entre profissionais como designers gráficos, artistas e ilustradores. A preferência pelo Illustrator deve-se à sua habilidade de criar imagens vetoriais. Diferentemente das imagens rasterizadas, que são baseadas em pixels, as imagens vetoriais são construídas usando fórmulas matemáticas, o que possibilita o redimensionamento para qualquer escala sem perda de qualidade. Essa característica faz do Illustrator a escolha perfeita para a criação de logotipos, ícones, diagramas e outros elementos de design que necessitam de escalabilidade, como é o caso do tabuleiro e de outros componentes prototipados nesta fase do projeto.



Figura 39. Ícone do Software Adobe Illustrator 2021
Fonte: O autor (2024).

3.1.4.1 Definição dos objetivos de aprendizagem

A etapa de definição dos objetivos de aprendizagem é fundamental no processo de design de jogos educativos, pois estabelece as bases para todas as decisões de design subsequentes. Ao iniciar este processo, é crucial entender as necessidades e capacidades do público-alvo. Para crianças, em particular, os objetivos devem ser alinhados com o seu estágio de desenvolvimento cognitivo, físico e social, garantindo que o jogo seja ao mesmo tempo desafiador e acessível.

A clarificação dos objetivos de aprendizagem começa com uma análise detalhada das habilidades que se deseja desenvolver nas crianças. Isto pode incluir habilidades cognitivas, como raciocínio lógico, resolução de problemas e memória; habilidades físicas, como coordenação motora e percepção espacial; e habilidades sociais, como comunicação, colaboração e empatia. A identificação precisa dessas habilidades direciona o design do jogo para resultados educativos específicos e são apresentadas no quadro a seguir.

Habilidades para aprendizagem	
Tipo de Habilidade	Habilidade Alvo
Cognitivas	Abstração: Identificar e Filtrar Informações Essenciais.
	Decomposição: Quebrar Problemas em Partes Menores.
	Reconhecimento de Padrões: Identificar Semelhanças e Diferenças.
	Algoritmos: Desenvolver Passo a Passo Soluções.
Físicas	Coordenação motora fina: Realizar movimentos intencionais com os músculos menores do corpo (mãos e dedos).
	Coordenação Mão-Olho: Realizar movimento sincronizados entre ação e visão.
	Percepção Espacial: Perceber o espaço e ambientação do jogo e seus elementos.
Sociais	Comunicação e Expressão: Comunicar-se de maneira eficiente e respeitosa com demais jogadores.
	Colaboração e Trabalho em equipe: Realizar atividades colaborativas em conjunto com outros.
	Paciência e Espera pela Vez: Saber aguardar o momento apropriado para jogar.
	Respeito pelas Regras: Realizar ações permitidas e honestas na partida.
	Adaptação e Flexibilidade: Adaptar-se a elementos surpresa ou mudanças.

Quadro 5. Habilidades para aprendizagem.
Fonte: O autor (2024).

Uma vez que as habilidades alvo são identificadas, os objetivos de aprendizagem devem ser formulados de forma SMART: específicos, mensuráveis, alcançáveis, relevantes e temporais. Esta abordagem ajuda a criar metas claras e tangíveis para o jogo.

Objetivos de Aprendizagem		
Tipo de Habilidade	Habilidade Alvo	Objetivos de Aprendizagem
Cognitivas	Abstração	OA01 - Filtrar informações essenciais para compreender e resolver problemas complexos.
	Decomposição	OA02 - Dividir um problema complexo em componentes menores e mais gerenciáveis para facilitar a compreensão e solução.
	Reconhecimento de Padrões	OA03 - Identificar padrões comuns ou tendências que podem ser usadas para prever ou simplificar soluções futuras.
	Algoritmos	OA04 - Criar uma série ordenada de instruções lógicas para resolver problemas ou realizar tarefas.
Físicas	Coordenação motora fina	OA05 - Manusear elementos pequenos da interface do jogo com maestria.
	Coordenação Mão-Olho	OA06 - Orquestrar a visão para percepção de jogo e as mãos para ação de forma sincronizada para realizar jogadas.
	Percepção Espacial	OA07 - Compreender e administrar as jogadas mediante a percepção de organização do jogo por meio dos seus elementos de interface.

Sociais	Comunicação e Expressão	OA08 - Comunicar de forma clara, eficiente e respeitosa suas vontades, questionamentos e arguições aos demais jogadores.
	Colaboração e Trabalho em equipe	OA09 - Realizar ações coordenadas com outro(s) jogador(es) a fim de atingirem objetivos em comum.
	Paciência e Espera pela Vez	OA10 - Aguardar pacientemente a vez de jogar enquanto os demais jogadores realizam suas jogadas.
	Respeito pelas Regras	OA11 - Respeitar e atender a todas as regras do jogo previamente conhecidas.
	Adaptação e Flexibilidade	OA12 - Adaptar e criar novas estratégias mediante a mudanças no jogo.

Quadro 6. Objetivos de Aprendizagem.

Fonte: O autor (2024).

A escolha de objetivos de aprendizagem influencia diretamente o tipo de conteúdo e atividades-chaves incluídas no jogo. Por exemplo, se o objetivo é desenvolver habilidades matemáticas, o jogo pode incluir puzzles e desafios que requerem cálculo e lógica. Esta congruência entre objetivos e atividades garante que cada elemento do jogo contribua para o resultado educativo desejado.

Desta forma, estabeleceu-se como atividades-chaves para o jogo, ações conscientes que os jogadores podem realizar – comumente chamadas de jogadas – que permitam alcançar o objetivo final que é ganhar o jogo. Estas ações estão a base para a construção da Jogabilidade e Regras de Jogo, garantindo que todas as possibilidades de jogadas estejam em consonância com os Objetivos de Aprendizagem.

Atividades-chaves
HC01 - Tomada de decisão mediante a objetivos pré-definidos.
HC02 - Elaborar sequência de jogadas estruturadas, utilizando somente os recursos do jogo.
HC03 - Condicionar a jogada mediante a eventos imprevisíveis do jogo.
HC04 - Permitir repetição de jogadas sequencialmente.

Quadro 7. Atividades-chaves.

Fonte: O autor (2024).

Ressalta-se que a definição dos objetivos de aprendizagem não é um processo estático; deve ser revisado e ajustado com base no feedback e nos resultados dos testes. À medida que protótipos são desenvolvidos e testados com crianças reais, observações e dados podem revelar necessidades de ajustes nos objetivos para melhor alinhar o jogo com as necessidades e capacidades do seu público. Este ciclo iterativo de avaliação e ajuste é crucial para o sucesso do jogo como uma ferramenta educativa.

3.1.4.2 Definição e Design dos Elementos Lúdicos

A Etapa 2, Definição e Design dos Elementos Lúdicos, é onde os componentes tangíveis do jogo começam a tomar forma. Esta etapa envolve a seleção cuidadosa de elementos como: Formato e

características do Tabuleiros; Layout, tipos e informações das cartas; Tipos de dados que serão usados; e as características das peças dos personagens. Estes elementos não só definem a mecânica do jogo, mas também como ele será percebido e interagido pelos jogadores. A escolha consciente desses elementos pode aumentar significativamente o engajamento e a eficácia do aprendizado.

A escolha do Tabuleiro é uma das primeiras decisões nesta etapa. O design do tabuleiro influencia diretamente a jogabilidade e a interação dos jogadores com o jogo. Por exemplo, um tabuleiro simples pode ser adequado para jogos rápidos e diretos, enquanto um mais complexo pode criar uma experiência mais imersiva e estratégica. O tabuleiro deve ser projetado não apenas para ser esteticamente agradável, mas também funcional e alinhado com os objetivos de aprendizagem.

Existem diversos tipos de Tabuleiros, cada um se adequa a uma necessidade específica de jogo. Uma característica comum nos tabuleiros em jogos de sequência lógica é o formato de Matriz. Esta característica está ligada ao fato de que uma sequência lógica precisa conter instruções objetivas e estruturadas, isto é, um sentido linear.

Uma matriz é formada pela intercessão de linhas e colunas. A intercessão entre uma linha e uma coluna, forma uma célula. Cada célula deve ser reconhecida como uma instrução individual. A junção de várias células forma um conjunto de instrução. Esse conjunto de instrução deve ser estruturado de forma linear e lógica, visando alcançar o objetivo que está disposto no tabuleiro.

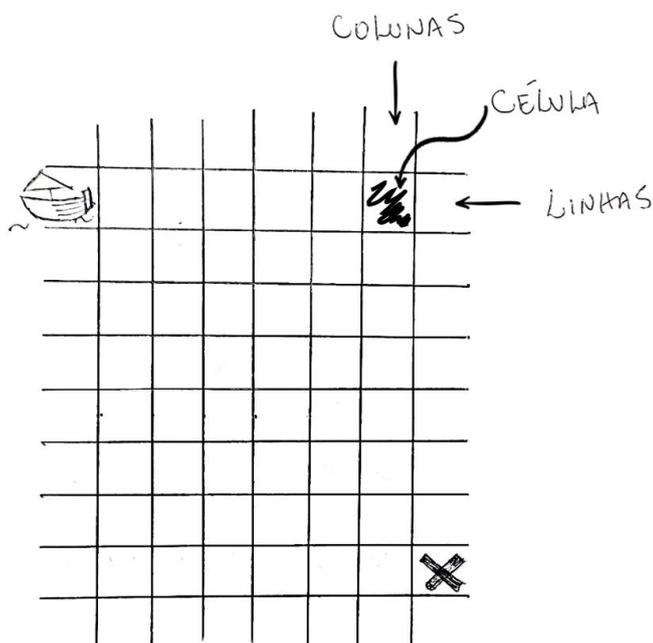


Figura 40. Tabuleiro em forma de matriz
Fonte: O autor (2024).

Para facilitar a compreensão dos conceitos, contextualizar o esboço do tabuleiro e os elementos dispostos nele, criou-se uma narrativa.

Era uma vez um navegante que viajava por mares distantes em busca de tesouros. Certa vez, ele encontrou um mapa com um grande X, que demarcava o exato local onde foi enterrado um tesouro cheio de riquezas. Segundo o mapa, o tesouro seria capaz de enriquecer até 10 gerações de quem o encontrar. O grande navegante, subiu em seu barco e agora precisa traçar a rota para encontrar o tesouro perdido. Ajude o viajante a encontrar o tesouro, utilizando uma sequência de passos.

O barco representará o objeto que irá percorrer o tabuleiro, isto é, o navegante da história, e o X representará o objetivo a ser alcançado, a saber, o tesouro.

Um comando representa uma ação isolada e independente que um objeto faz, o que é chamado na programação de Instrução. Os comandos podem ser repetidos, formando um grupo de comandos ou Conjunto de Instruções. Um conjunto de instruções formam uma Sequência Lógica e linear. Para este projeto, a Sequência Lógica deve ser representada sempre da esquerda para a direita, conforme representado na figura a seguir.

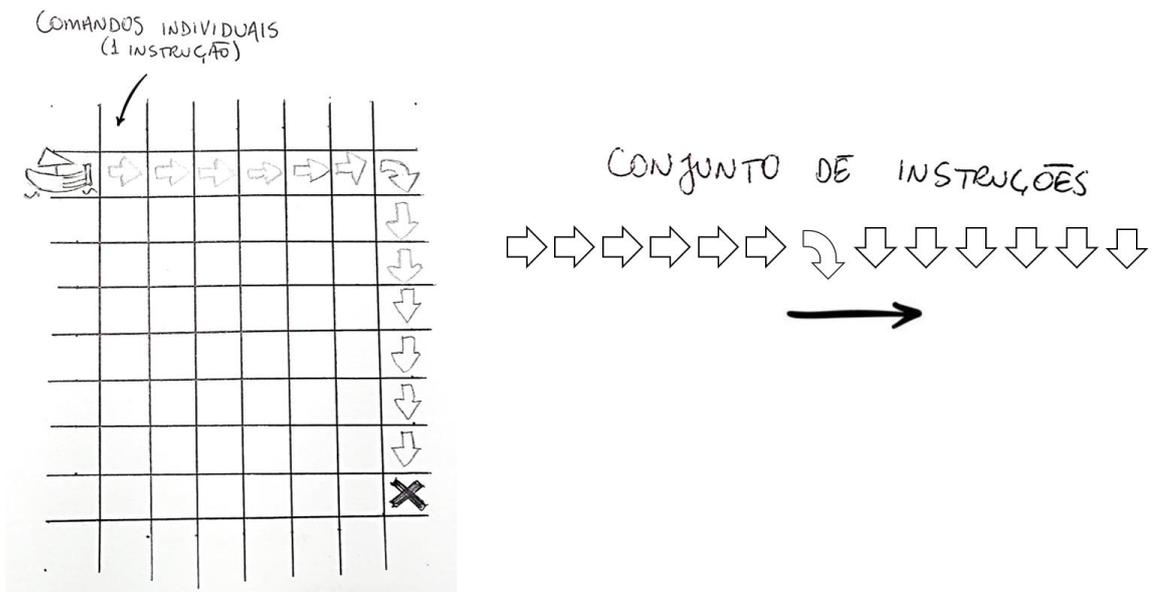


Figura 41. Instruções e Sequência Lógica
Fonte: O autor (2024).

Outra característica em um tabuleiro em forma de matriz, é possibilitar a construção de diferentes tipos de sequência lógica, tendo em vista que até o presente esboço, não há limitadores que impeçam o uso de qualquer célula do tabuleiro, sendo possível fazer diferentes modelos de Sequências Lógicas nele.

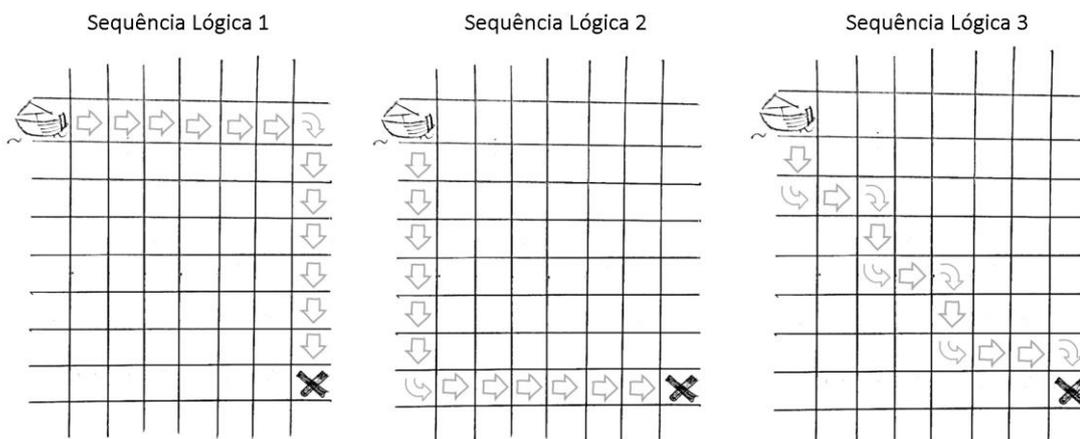


Figura 42. Modelos de Sequências Lógica.
Fonte: O autor (2024).

Para adicionar dinamismo ao tabuleiro e introduzir um primeiro nível de desafio, é essencial implementar certos elementos restritivos. A finalidade destes elementos é demandar dos jogadores um raciocínio mais aprofundado, permitindo-lhes formular sequências lógicas que superem tais restrições e os levem até o objetivo final. Com isso em mente, dois elementos restritivos foram estabelecidos: as **Rotas Possíveis** e os **Obstáculos**. Até o momento, nenhum tamanho foi definido para o tabuleiro. Para os esboços iniciais, será usado um tabuleiro de 30x30, numerando as linhas de 1 a 30 e as colunas, identificadas com as 26 letras do alfabeto, adicionando ainda as combinações AB, AC, AD e AE para as quatro últimas colunas excedentes.

	A	B	C	D	E	F	G	H	I	J	I
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
...											

Figura 43. Tamanho do Tabuleiro - 1º Esboço.
 Fonte: O autor (2024).

As **Rotas Possíveis** são caminhos pré-definidos no tabuleiro que os jogadores podem seguir para alcançar seus objetivos. Estas rotas são projetadas para oferecer várias opções de trajetória, cada uma com seus próprios desafios e recompensas. A seleção de uma rota exige que os jogadores avaliem as opções disponíveis e tomem decisões estratégicas baseadas em seus objetivos de jogo e nas condições atuais do tabuleiro.

As rotas podem variar em comprimento, dificuldade e recompensas, oferecendo aos jogadores escolhas que podem influenciar significativamente o desenrolar do jogo. Rotas mais longas podem ter menos obstáculos, mas exigem mais tempo para serem concluídas, enquanto rotas mais curtas podem ser mais arriscadas ou desafiadoras.

Ao desenvolver os esboços iniciais das rotas no tabuleiro, percebeu-se uma questão importante: quando o tabuleiro era estruturado como uma matriz com dimensões de números pares, surgiam disparidades no número de passos necessários para os jogadores alcançarem determinados pontos. Isso ocorria porque um tabuleiro de dimensões pares não possui uma célula verdadeiramente central. Para resolver este desafio, decidiu-se adicionar uma linha extra e uma coluna adicional ao tabuleiro. Com a inclusão da linha 31 e da coluna AF, a célula P16 passou a ser o centro de uma matriz agora expandida para 31x31 células. Essa alteração garantiu uma distribuição mais equitativa de distâncias e oportunidades para todos os jogadores, mantendo a integridade e equidade do design do jogo, de forma que para atingir a célula central cada jogador terá que fazer uma sequência lógica de 29 passos.

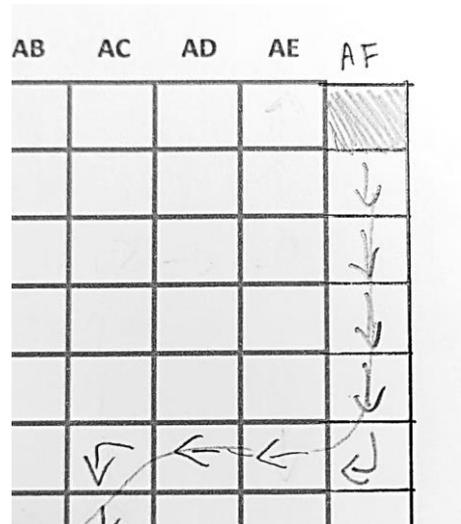


Figura 44. Coluna adicionada - AF.
 Fonte: O autor (2024).

O design do tabuleiro inclui quatro rotas principais, expressas pelo símbolo (\rightarrow), idealizadas para permitir que quatro jogadores participem simultaneamente e interajam entre si. Além dessas rotas principais, foram desenvolvidas rotas secundárias. As rotas secundárias, representadas pelos símbolos (\Rightarrow) são projetadas para serem mais desafiadoras, pois incluem obstáculos e enigmas adicionais. No entanto, elas oferecem uma vantagem única: a capacidade de realizar saltos diagonais, uma opção que não está disponível nas rotas principais. Apesar das diferenças nos desafios e recursos entre as rotas principais e adicionais, ambas requerem um total de 29 passos para que os jogadores alcancem o objetivo final. Este layout garante um equilíbrio entre variedade e uniformidade, mantendo a coerência e a competitividade do jogo.

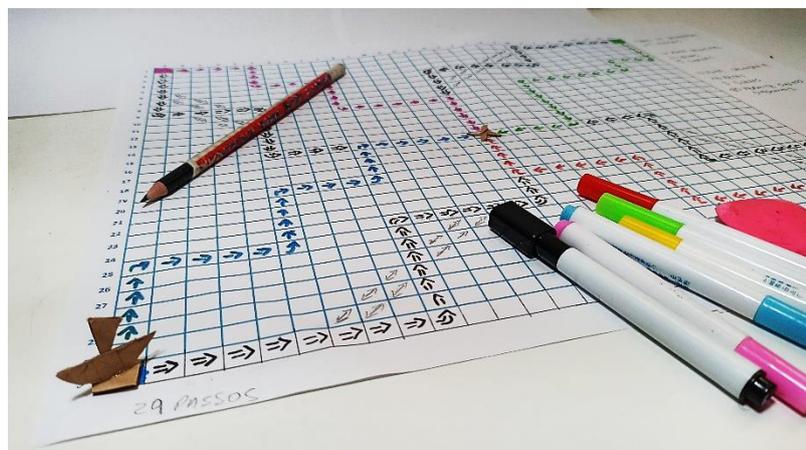


Figura 45. Rotas principais e secundárias.

Fonte: O autor (2024).

Para facilitar a visualização da área que será empregada no design do tabuleiro, desenvolveu-se uma representação vetorizada dela no Illustrator. Nessa versão digital, os espaços destinados para as rotas já estão claramente demarcados, proporcionando uma compreensão mais clara da distribuição e do layout das rotas no tabuleiro. Essa representação digital serve como uma ferramenta valiosa para o planejamento e aperfeiçoamento do design do jogo, permitindo ajustes precisos e uma visão geral do espaço de jogo.

As rotas em cor representam as rotas principais de cada jogador. As rotas em cinza representam as rotas secundárias ou alternativas.

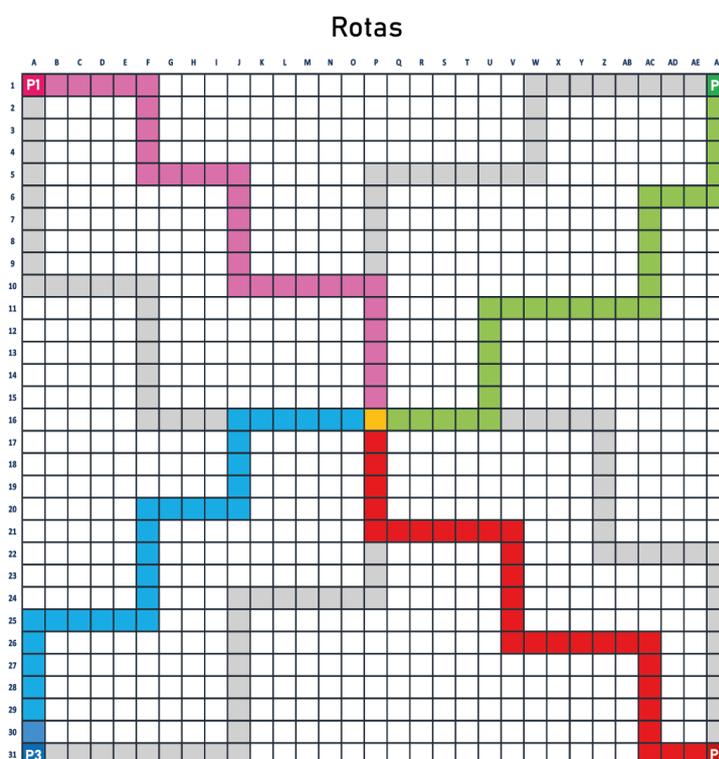


Figura 46. Rotas do Tabuleiro.
Fonte: O autor (2024).

Ao observar as rotas, é possível perceber que os 4 jogadores verão o tabuleiro em perspectivas diferentes, como que divididos por quadrantes.

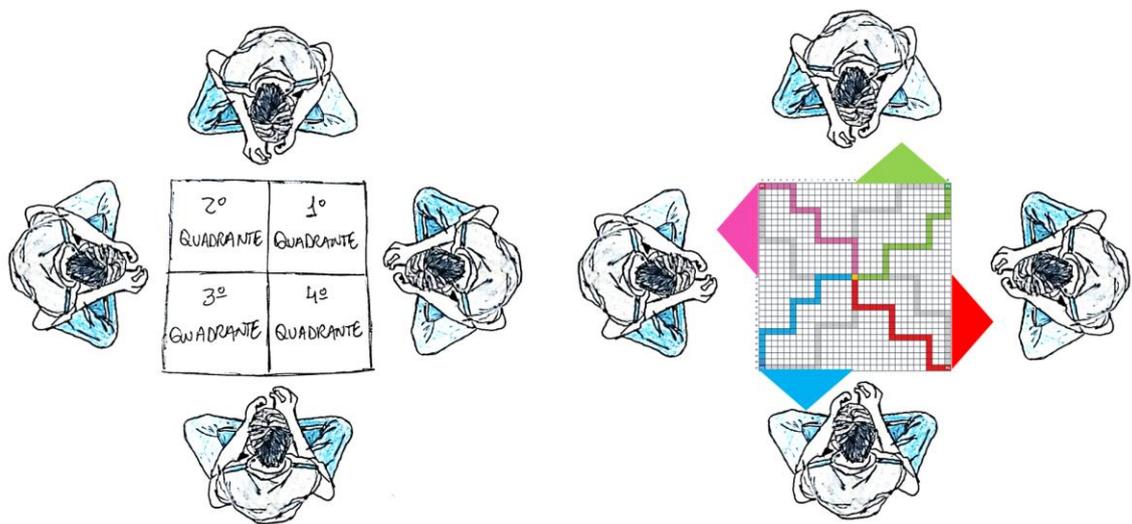


Figura 47. Visão do tabuleiro a partir das perspectivas dos 4 jogadores.
 Fonte: O autor (2024).

Portanto, é necessário criar as guias de colunas e linhas a partir da perspectiva individual de cada jogador. O novo modelo de guia utilizando números e letras para atender este critério é apresentado a seguir.

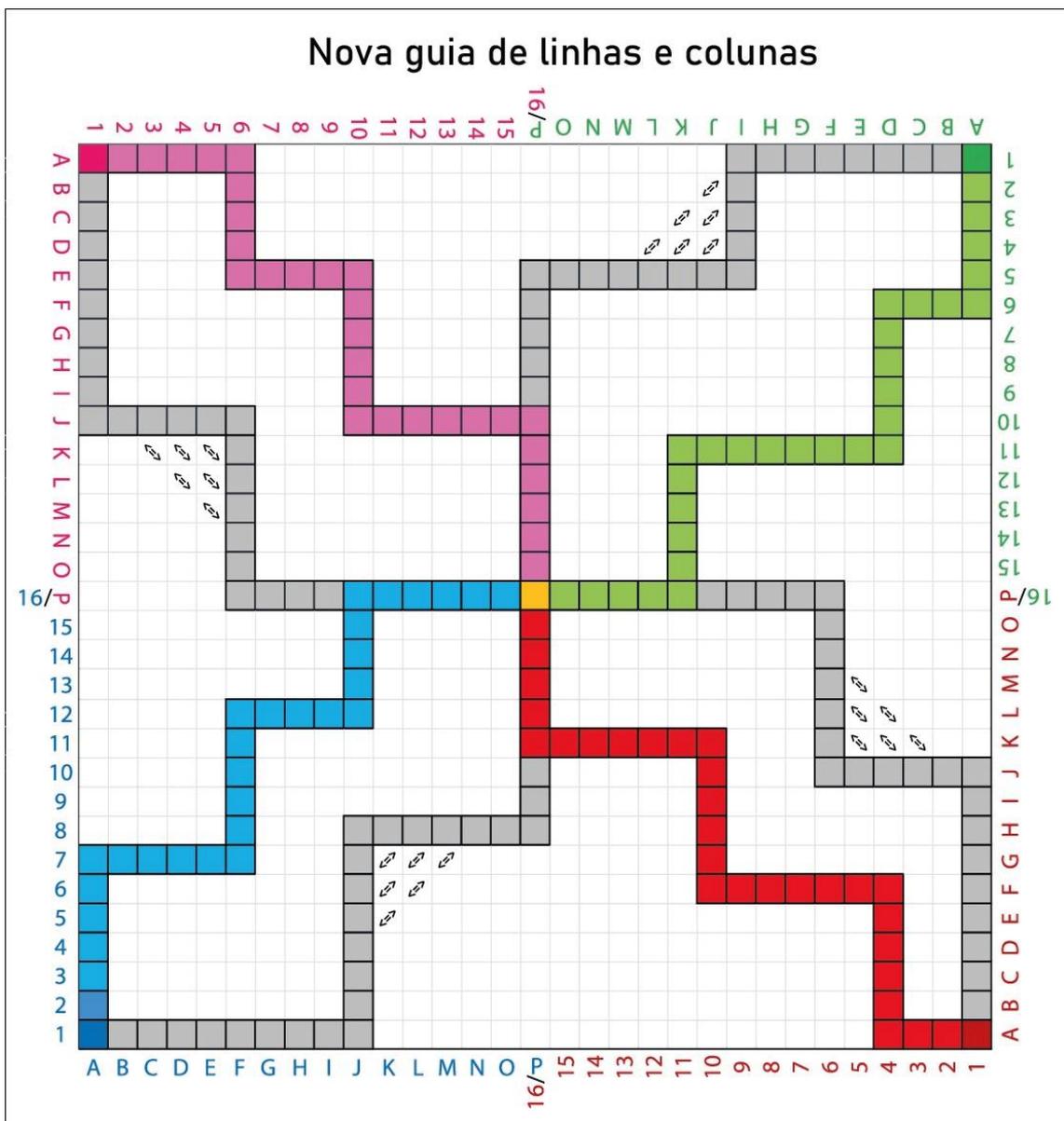


Figura 48. Novas guias de linhas e colunas.
 Fonte: O autor (2024).

Após estabelecer as rotas principais e secundárias, tornou-se evidente que certas áreas do tabuleiro permaneciam inutilizadas. Estas áreas serão aproveitadas para incorporar elementos de design lúdico. Com isso em mente, o foco desta etapa inicial se concentra na elaboração do design do tabuleiro, inspirado no contexto sociocultural amazônico. Este cenário será desenvolvido com atenção aos detalhes, visando capturar a essência e as características únicas da região,

proporcionando uma experiência imersiva e educativa para os jogadores. A área destinada ao Design do tabuleiro é apresentada em cinza, conforme figura a seguir.

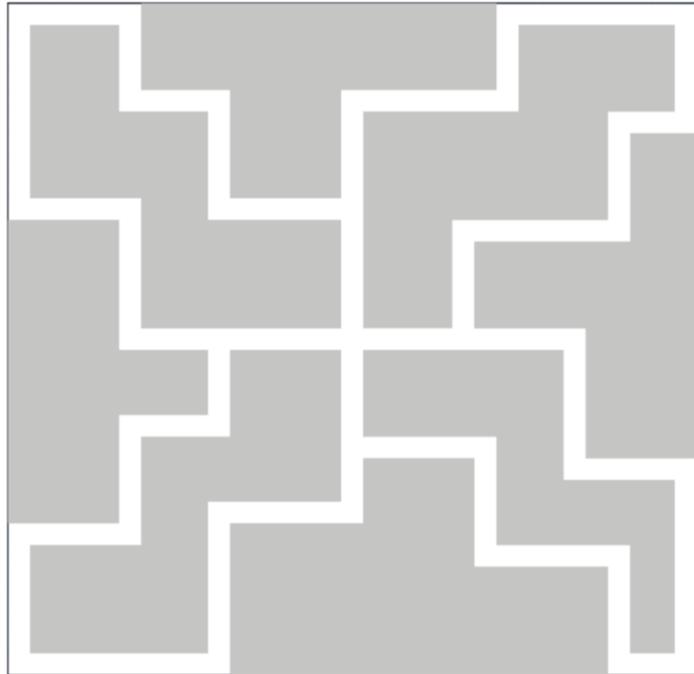


Figura 49. Área para design do tabuleiro.
Fonte: O autor (2024).

Para começar a criar os primeiros esboços, foi feita uma reprodução da área em um papel A4, com o cuidado de criar o design e não invadir as rotas, nem ultrapassar os limites do tabuleiro.

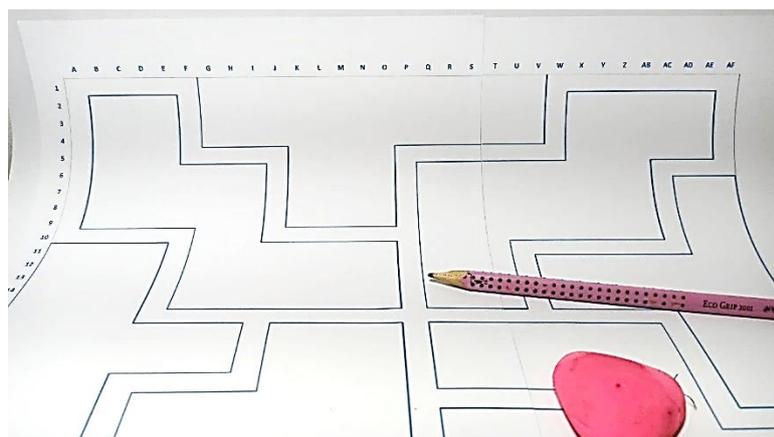


Figura 50. Primeiros esboços do design do tabuleiro.
Fonte: O autor (2024).

Cartas são frequentemente utilizadas em jogos educativos para introduzir elementos de surpresa, estratégia e memorização. A definição de cartas inclui não apenas o conteúdo e as regras associadas a elas, mas também seu design visual e tátil. Cada carta deve ser projetada para ser intuitiva e informativa, garantindo que contribua para a experiência de aprendizado de forma significativa. O objetivo de uso das cartas é formar um *deck*, um conjunto de cartas com a sequência lógica exata para fazer o personagem – representado nos esboços iniciais por um barco – chegar até o centro do tabuleiro, que é o objetivo final – representado em nosso esboço por um “X”. Nesta versão, serão prototipadas 5 grupos de cartas.

Cartas de Ação: Estas cartas representam instruções individuais ou passos que, quando agrupados, formarão a sequência lógica para alcançar o objetivo. Para construção de uma sequência lógica a partir das rotas idealizadas 5 cartas: **Avançar**, **Virar à Esquerda**, **Virar à Direita**, **Avançar para Esquerda** e **Avançar para Direita**. O design das cartas deve ser sugestivo e as informações contidas nelas devem expressar explicitamente seu efeito dentro do jogo.

Desta forma, as cartas foram construídas em formato de wireframes, representações sem uso de cores e com tipografia de formato geral, que tem por objetivo estruturar as informações das cartas, a saber, as informações a serem percebidas pelo usuário.

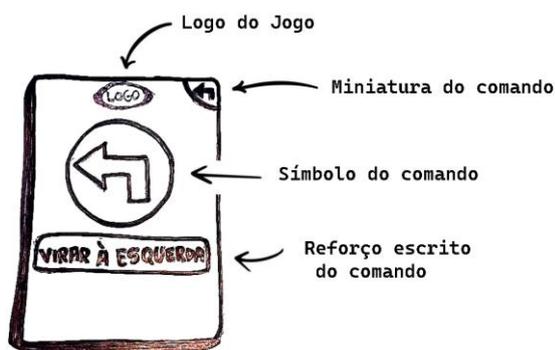


Figura 51. Arquitetura das informações das cartas de ação.
Fonte: O autor (2024).

Ao imaginar uma criança, na faixa dos 6 a 7 anos de idade, segurando um deck de cartas de 29 cartas, representando os 29 passos necessários para fazer toda a sequência lógica, é previsível que haja uma certa dificuldade no manuseio, o que pode prejudicar o jogador. Pensando nisto, acrescentou-se a miniatura do comando, no canto superior direito. Assim, não é necessário segurar

todas as cartas ao mesmo tempo; elas podem ser empilhadas, com uma ligeira inclinação para a direita, revelando o ícone. Isso facilita o acompanhamento da sequência, como mostrado na figura a seguir.

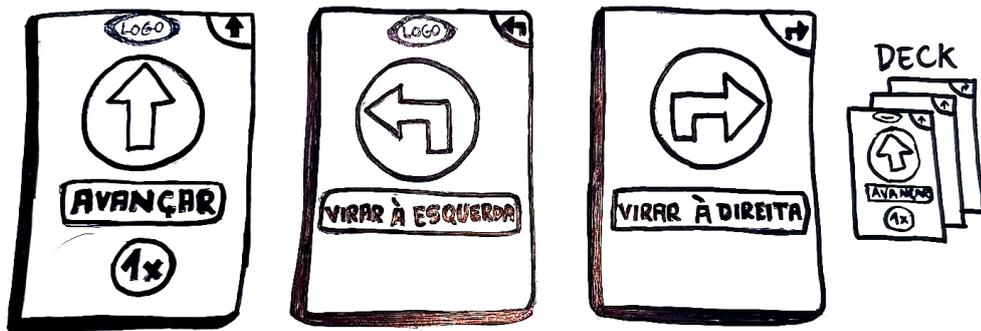


Figura 52. Exemplo de um deck de cartas.
Fonte: O autor (2024).

Após a definição da arquitetura das informações da carta, foi então esboçado o design com todos os elementos visuais, tais com cores, símbolos e tipografias baseada no contexto sociocultural amazônico.

Cartas de Repetição: As cartas de repetição servirão para potencializar o uso das Cartas de Ação, permitindo repetir o comando delas n vezes. Por exemplo, com uma **Carta de Repetição 3x** + uma única carta de ação **Avançar**, pode ser feito o seguinte sequência lógica:

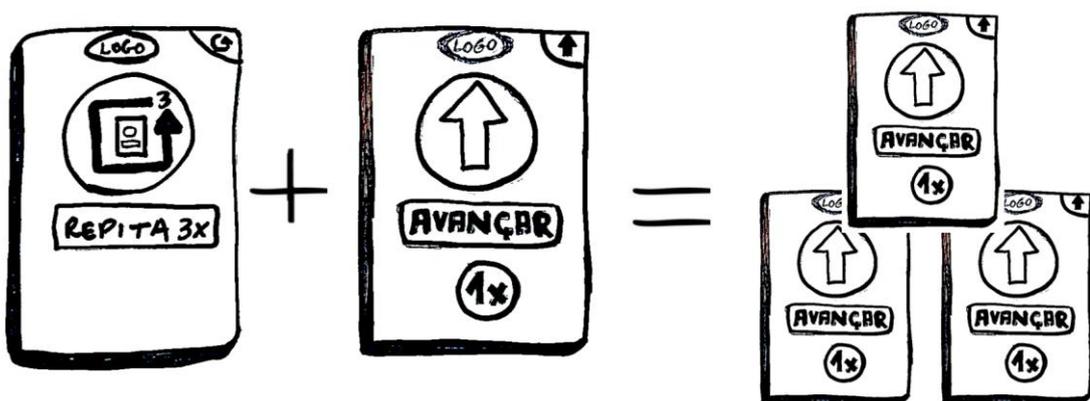


Figura 53. Cartas de Repetição
Fonte: O autor (2024).

Carta de Salto: As cartas de salto possibilitam jogadas alternativas às convencionais. A carta **Saltar** poderá ser usada para fazer um atalho utilizando cantos de rotas. As casas que permitem o salto são identificadas com o símbolo \Leftrightarrow .



Figura 54. Carta Especial: Saltar
Fonte: O autor (2024).

Em uma jogada normal, para ir da casa J4 até a casa N8, por exemplo, o jogador poderia usar 9 cartas para construir o seguinte sequência lógica.

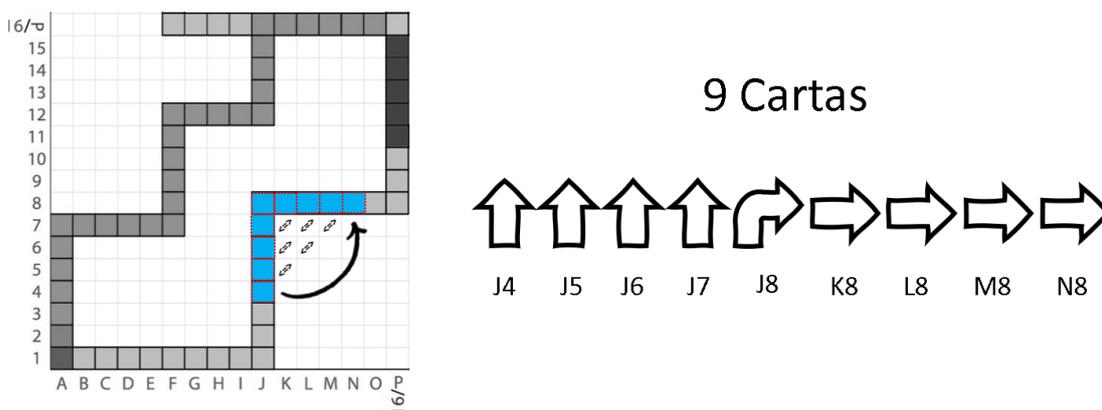
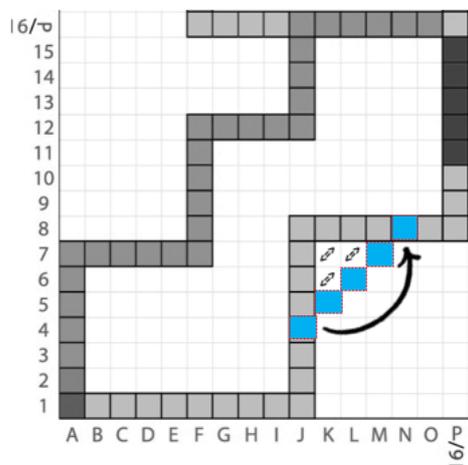
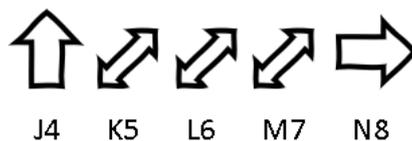


Figura 55. Sequência lógica sem uso da carta SALTAR
Fonte: O autor (2024).

Com o uso da carta Saltar, é possível realizar um “salto” e assim fazer a sequência lógica usando menor quantidade de cartas (menor quantidade de rodadas), permitindo que o jogador alcance o objetivo no centro do tabuleiro mais rapidamente.



Usando 3 cartas SALTAR

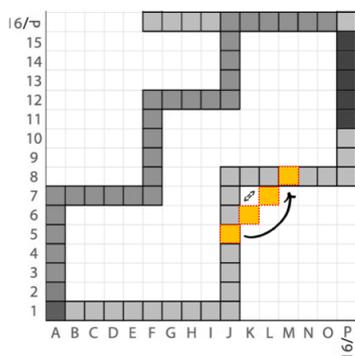


J4 K5 L6 M7 N8

5 cartas

Figura 56. Sequência lógica com a carta SALTAR
 Fonte: O autor (2024).

Outras duas variações de jogadas para o uso das cartas de salto são permitidas, com 1 e com 2 cartas SALTAR.

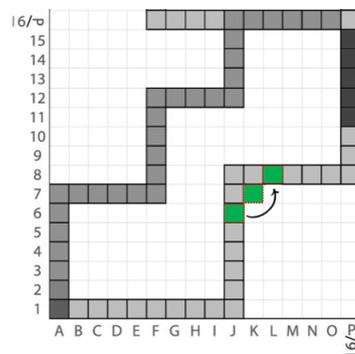


Usando 2 cartas SALTAR



J5 K6 L7 M8

4 cartas



Usando 1 cartas SALTAR



J6 K7 L8

3 cartas

Figura 57. Variações de jogadas com a carta SALTAR

Fonte: O autor (2024).

Carta de Controle: A Carta de Controle permite personalizar um novo fluxo diferente para a Sequência lógica. Em Algoritmos, o conceito de Estrutura de Controle é baseado em condição: **Se (condição é verdadeira) seguir Fluxo 1; senão, seguir Fluxo 2.**

Neste jogo, a Carta de Controle poderá ser usada única e exclusivamente na casa do tabuleiro que permite realizar um salto. Ao usar a Carta de Controle, o jogador estará verificando a seguinte condição:

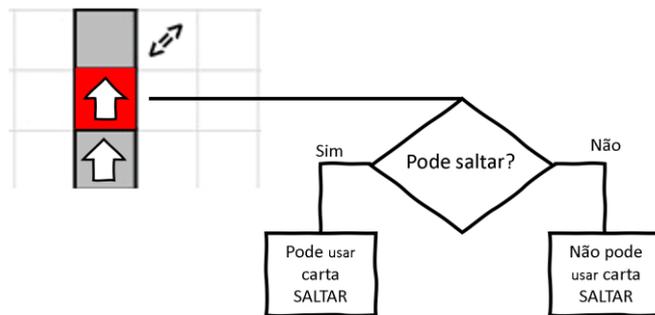


Figura 58. Casa do Tabuleiro que permite salto.

Fonte: O autor (2024).

Portanto, quando um jogador chega à casa do tabuleiro que permite realizar um salto, ele deve primeiro invocar a Carta de Controle antes de usar a carta SALTAR. Se não possuir a Carta de Controle, o jogador é obrigado a seguir o caminho normal da rota alternativa. Além disso, se a Carta de Controle for usada em uma casa do tabuleiro onde o salto não é permitido, o jogador também deve continuar pelo caminho normal. O propósito dessa regra é habilitar o jogador a executar o movimento alternativo, que é o salto diagonal. A combinação dessas ações pode ser feita da seguinte maneira:

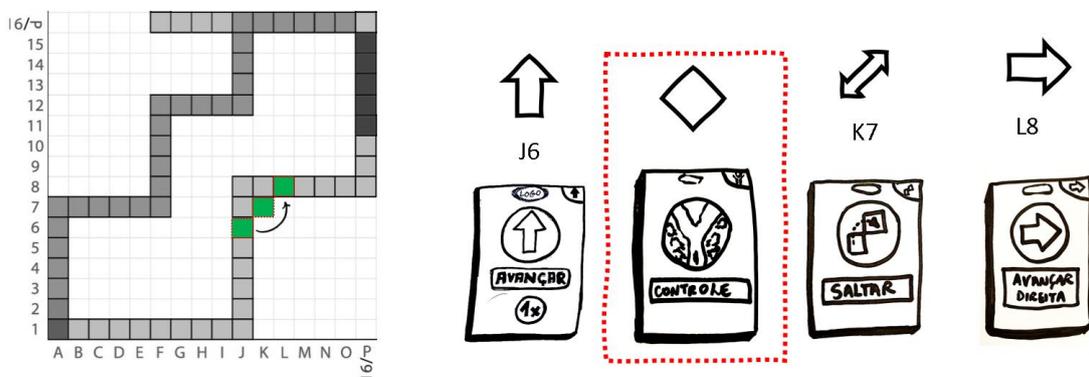


Figura 59. Uso da Carta de Controle em uma Sequência Lógica.
Fonte: O autor (2024).

O design da carta representa uma bifurcação que dá acesso a dois rios, muito comum no Amazonas.

Carta de Obstáculo: Esta carta foi criada para tornar mais difícil o percurso do oponente pela rota alternativa, colocando um obstáculo nesse caminho. Assim, se o oponente estiver utilizando a rota alternativa, que inclui um trecho do trajeto adversário, ele precisará de uma Carta de Quebra de Obstáculo para superar os obstáculos estabelecidos. O design da carta representa uma árvore que não resistiu aos eventos climáticos e caiu sobre uma estrada, impedindo o trânsito de veículos.



Figura 60. Carta de Obstáculo.
Fonte: O autor (2024).

Quando usada, o jogador que invocou a Carta de Obstáculo deve utilizar um pino para marcar no mapa o exato local do bloqueio, permitindo que o seu oponente planeje uma estratégia para quebrá-lo ou faça outra rota.

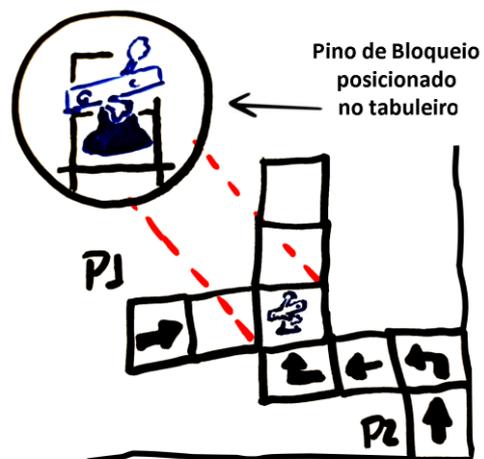


Figura 61. Pino de Bloqueio indicado no tabuleiro
 Fonte: O autor (2024).

O design do Pino de Bloqueio corresponde ao significado da carta, a queda de árvores em estradas e vias de trânsito.



Figura 62. Protótipo de Pino de Bloqueio.
 Fonte: O autor (2024).

Carta Remover Obstáculo: usada para ultrapassar uma Carta de Obstáculo. Pode ser invocada para substituir a carta AVANÇAR no bloco onde está posicionado o Pino de Bloqueio. Sem ela, o jogador fica impossibilitado a seguir pelo trajeto, podendo começar uma nova estratégia por outra rota.



Figura 63. Carta Remover Obstáculo.
Fonte: O autor (2024).

Carta Teste de Mesa: Quando esta carta é invocada, o jogador pausa o jogo para realizar um teste de mesa. O teste de Mesa é uma análise dos resultados obtidos, visando comprovar que a Sequência lógica alcança o resultado esperado. Assim, ele pode reproduzir todos comandos estruturados através da ordem das cartas para simular a ação do personagem no tabuleiro. Este recurso permite prever e corrigir erros na jogada final.



Figura 64. Carta Teste de Mesa
Fonte: O autor (2024).

Personagens permitem que os jogadores se envolvam mais profundamente com o jogo, oferecendo uma conexão pessoal e emocional. Para definir os personagens, é necessário considerar o que eles representam e como se comunicam com o público-alvo, bem como avaliar se eles encorajam a identificação e a empatia por parte dos alunos. Personagens bem projetados podem servir como guias, adversários ou avatares no processo de aprendizagem. Neste jogo, serão personalizados alguns personagens com características da fauna amazônica.

Para representar a fauna amazônica, foram idealizados 4 personagens principais: O Boto Vermelho, o Papagaio, a Onça Pintada e o Pirarucu.

A escolha dos personagens visou evidenciar por meio de suas características, cada um dos 4 pilares do Pensamento Computacional.

A Onça Pintada é um dos maiores felinos a habitar as florestas amazônicas. Ela representa a habilidade de Algoritmo, pois, na hora de caçar, ela traça uma estratégia, pensa nas ações que deve realizar e por fim, executa toda a ação como planejado.

O Papagaio é uma das aves mais lindas do Amazonas. Ele caracteriza a habilidade de Reconhecimento de Padrões, pois consegue imitar sons de outros animais e falas humanas.

O Boto Vermelho é ótimo nadador. Ele representa a habilidade de Abstração, pois com sua boa visão, são ágeis para se desviar de obstáculos debaixo d'água para alimentar de peixes.

O Pirarucu, por sua vez, consegue respirar embaixo d'água e fora dela. Ele representa a habilidade de Decomposição, pois, mesmo com o problema para respirar em águas com pouco oxigênio, ele é capaz de resolver este problema respirando parcialmente pelas brânquias e pela bexiga.

Os personagens criados representarão os 4 Pinos disponíveis, que devem ser escolhidos antes do início do jogo por cada jogador e ser posicionado em cada uma das 4 extremidades do tabuleiro, marcando o ponto de partida para a Sequência Lógica. Os pinos serão usados ainda para o Teste de Mesa, na qual o jogador poderá testar se a sequência lógica estruturada por meio das cartas está correta.



Figura 65. Protótipo de Pino. Personagem Onça Pintada
Fonte: O autor (2024).

Os Pinos serão acompanhados das respectivas Cartas de Personagens, que apresenta uma narrativa sobre o personagem, com suas características físicas e comportamentais, buscando gerar uma identificação com o aluno.

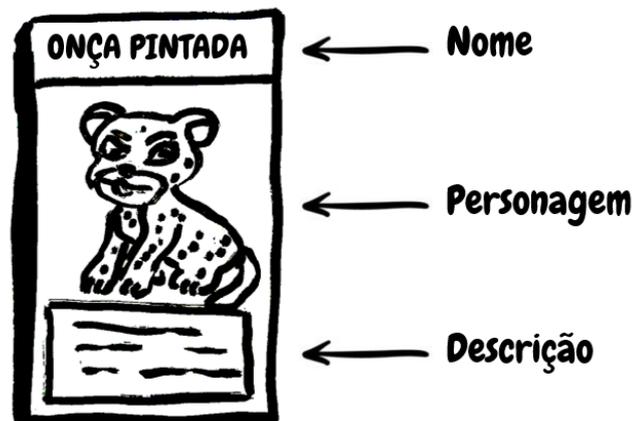


Figura 66. Protótipo de Carta de Personagem.
Fonte: O autor (2024).

Pensando ainda na personalização, necessidade esta percebida nas fases iniciais do projeto, o jogo disponibilizará alguns pinos em branco, onde os jogadores poderão personalizar com seus próprios

personagens, utilizando canetinhas coloridas. Esses pinos serão acompanhados ainda de Cartas de Personagem com espaço onde o jogador pode desenhar o personagem idealizado e escrever uma narrativa sobre ele.



Figura 67. Protótipo de Pino e Carta Personalizável.
Fonte: O autor (2024).

CAPÍTULO 4

Resultados

Neste capítulo será apresenta o jogo concebido e todos os seus elementos lúdicos, bem como as regras de jogo, tipos de jogadas e cenários de uso.

4.1 Codezônia – Resumo do jogo

Codezônia é um jogo de tabuleiro educacional desenvolvido para ensinar conceitos de Algoritmos e Programação de forma lúdica e envolvente para crianças. Ambientado na floresta amazônica, o jogo apresenta quatro personagens principais inspirados na fauna local: Boto, Papagaio, Onça Pintada e Pirarucu. Cada personagem representa um conceito-chave do pensamento computacional.

Foi projetado para ser usado como recurso didático com estudantes do Ensino Fundamental 1, especialmente Séries Iniciais, incorporando elementos lúdicos do contexto sociocultural da Amazônia em sua representação visual.

O objetivo de Codezônia é que os jogadores usem cartas de ação para mover seus personagens ao longo do tabuleiro, superando obstáculos. O objetivo final é chegar ao centro do tabuleiro, que simboliza a descoberta de um tesouro perdido na floresta.

O jogo é jogado por 2 a 4 jogadores, cada um representando um dos personagens principais do jogo: Boto Vermelho, Papagaio, Onça Pintada e Pirarucu. Os jogadores recebem um conjunto de cartas de ação que representam movimentos específicos que seus personagens podem fazer no tabuleiro. Eles devem usar essas cartas de forma estratégica para criar uma sequência lógica perfeita para navegar pelo tabuleiro e alcançar o objetivo central.

Sua logo é representada por uma Onça Pintada, animal característico da região que muitas vezes é retratado como símbolo da Amazônia.



Figura 68. Logo do jogo Codezônia.
Fonte: O autor (2024).

4.2 O Tabuleiro Codezônia

A integração dos elementos lúdicos é vital para criar uma experiência coesa e engajadora. Cada elemento deve ser considerado em relação aos outros para garantir que trabalhem juntos harmoniosamente. Neste sentido, a estética dos elementos lúdicos desempenha um papel importante na atração e retenção do interesse das crianças. Cores vibrantes, designs imaginativos e materiais táteis podem tornar o jogo mais atraente e memorável. A estética deve, no entanto, equilibrar atração visual com clareza e funcionalidade, assegurando que os elementos lúdicos sejam fáceis de entender e usar.

O Tabuleiro foi projetado para 4 jogadores jogarem simultaneamente. Para isto, foi necessário projetar os elementos visuais como a floresta, os rios, as vitórias-régias e as cachoeiras em duas dimensões - 2D, tendo em vista que a representação 3D iria favorecer a visualização de alguns jogadores e outros não.



Figura 69. Tabuleiro Codezônia
Fonte: O autor (2024).

O ponto de partida de cada jogador é indicado pelo quadrado de cor azul nas 4 extremidades do jogo. O ponto central indica o objetivo final a ser alcançado pela sequência lógica.

A escolha de representar a Floresta Amazônica, os rios, a vitória-régia e as cachoeiras no jogo foi motivada por diversos fatores que se alinham com os objetivos educativos e lúdicos do projeto.

Primeiramente, a Floresta Amazônica é um ecossistema único e fascinante, repleto de biodiversidade e mistérios. Ao ambientar o jogo nesse cenário, busca-se despertar o interesse das crianças pela natureza e pela conservação ambiental, promovendo a conscientização sobre a importância da preservação da Amazônia. Os rios amazônicos desempenham um papel fundamental na região, sendo vias de transporte, fonte de alimento e habitat para diversas espécies. Ao integrar os rios ao tabuleiro, pretende-se destacar a importância da água para a vida e para as comunidades

loais, tudo isso de forma lúdica. A vitória-régia, uma das plantas aquáticas mais emblemáticas da Amazônia, foi escolhida como elemento do jogo devido à sua beleza e importância cultural. As cachoeiras são elementos característicos da geografia amazônica, sendo um destaque para o turismo local.

4.3 Os Personagens

A escolha dos personagens da fauna amazônica para representar os pilares do Pensamento Computacional no jogo "Codezônia" foi cuidadosamente pensada para criar uma conexão significativa entre as habilidades dos animais e os conceitos que eles representam.

A Onça Pintada foi selecionada para representar a habilidade de Algoritmo devido ao seu comportamento de caça. Assim como um algoritmo, a Onça Pintada traça uma estratégia antes de agir, pensando nas ações que deve realizar para alcançar seu objetivo. Essa analogia é fácil de entender para as crianças, pois podem associar a forma como a Onça Pintada caça com a forma como devem criar sequências lógicas no jogo.

O Papagaio foi escolhido para representar a habilidade de Reconhecimento de Padrões devido à sua capacidade de imitar sons. Os padrões de som que o Papagaio reconhece e reproduz são semelhantes aos padrões lógicos que as crianças precisam identificar para resolver problemas no jogo. Além disso, a natureza colorida e divertida do Papagaio torna-o um personagem cativante para as crianças.

O Boto Vermelho foi designado para representar a habilidade de Abstração por sua habilidade de se desviar de obstáculos debaixo d'água. Assim como no jogo, onde os jogadores precisam abstrair obstáculos e pensar em soluções criativas para avançar, o Boto Vermelho utiliza sua habilidade de abstração para se alimentar de peixes, desviando-se habilmente de obstáculos subaquáticos.

Por fim, o Pirarucu foi escolhido para representar a habilidade de Decomposição por sua capacidade de respirar embaixo d'água e fora dela. Essa habilidade do Pirarucu é semelhante ao conceito de decomposição na programação, onde problemas complexos são quebrados em partes menores para facilitar a resolução. O Pirarucu exemplifica como a decomposição pode ajudar a resolver problemas aparentemente difíceis.

Ao escolher esses personagens, o objetivo é não apenas ensinar os conceitos de Algoritmo, Reconhecimento de Padrões, Abstração e Decomposição, mas também criar uma conexão emocional e lúdica com o ambiente amazônico, tornando o aprendizado mais envolvente e significativo para as crianças.



Figura 70. Personagens do jogo Codezônia.
Fonte: O autor (2024).

Cada personagem será representado pelo seu respectivo pino no tabuleiro.

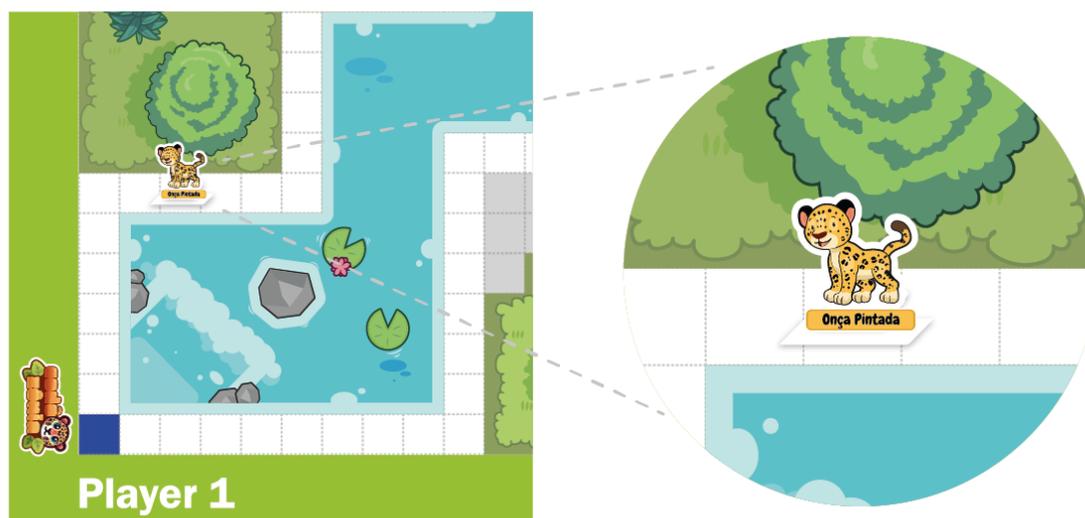


Figura 71. Pino da personagem Onça Pintada
Fonte: O autor (2024).

4.4 As Cartas

As cartas desempenham um papel fundamental no jogo, pois são elas que permitirão que os jogadores façam a formação das sequências lógicas necessárias para avançar pelo tabuleiro. A escolha de utilizar cartas como elemento central do jogo foi motivada pela sua versatilidade e capacidade de representar diferentes instruções de forma clara e concisa.

Para garantir que as cartas fossem eficazes como ferramentas de aprendizagem, optou-se por criar desenhos alegres e divertidos, que cativam a atenção das crianças e as incentivam a se envolver no jogo. Além disso, cada carta foi cuidadosamente projetada para comunicar claramente o seu objetivo no jogo, seja indicando uma direção a seguir no tabuleiro, repetindo um comando ou superando obstáculos. Os desenhos foram desenvolvidos para representar de forma visual e intuitiva as ações que os jogadores devem executar, facilitando a compreensão e o uso das cartas durante o jogo.

As Cartas de Ações simbolizam instruções individuais e o conjunto estruturado e sequencial delas formarão a sequência lógica.



Figura 72. Cartas de Ação.
Fonte: O autor (2024).

As Cartas de Repetição remete ao conceito de Estruturas de Repetição. Por meio delas, os jogadores ganharão velocidade em suas jogadas, multiplicando em 2, 3, 4 e 5 vezes uma Carta de Ação de seu interesse.



Figura 73. Cartas de Repetição.
Fonte: O autor (2024).

A Carta de Controle remete ao conceito de Estrutura de Controle. O tabuleiro foi construído para possibilitar um atalho na rota (salto), em uma área específica. Para realizar um salto, o jogador deve invocar primeiro uma Carta de Controle. A Carta de Controle representa uma bifurcação entre braços de rios, na qual os navegantes devem optar por uma das rotas. Este aspecto geográfico é muito comum no rios da Amazônia.



Figura 74. Carta de Controle.
Fonte: O autor (2024).

Após invocar a Carta de Controle e a sequência lógica do jogador estiver em área possível de realizar um salto que abrevia a jogada em números de cartas, o jogador então pode usar a Carta de Salto.

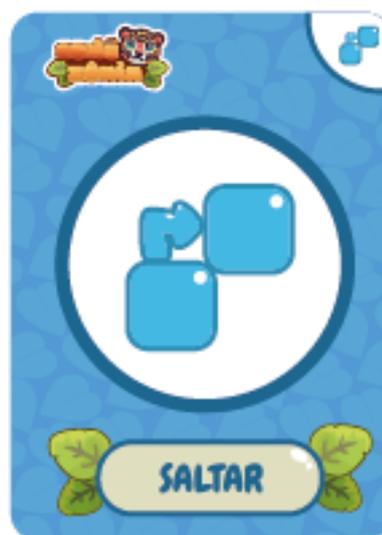


Figura 75. Carta de Salto.
Fonte: O autor (2024).

Para dar mais dinamismo no jogo e permitir maior competitividade, foram criadas as Cartas de Obstáculo. Esta carta permite que um jogador posicione um obstáculo no meio da rota de seu adversário.



Figura 76. Carta de Obstáculo.
Fonte: O autor (2024).

Para ultrapassar o obstáculo. O jogador prejudicado precisará invocar a Carta Remover Obstáculo.



Figura 77. Carta Remover Obstáculo.
Fonte: O autor (2024).

Para permitir que o jogador faça uma simulação de sua sequência lógica e a valide a qualquer momento do jogo, mesmo sem estar em sua vez de jogar, foi criada a Carta Teste de Mesa.

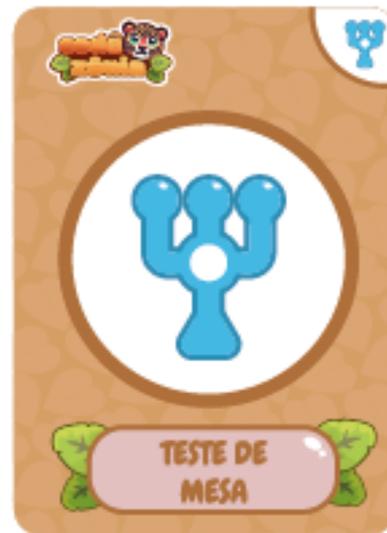


Figura 78. Carta Teste de Mesa.
Fonte: O autor (2024).

Visando garantir o envolvimento dos jogadores com a temática do jogo e seu contexto geográfico, foram criadas as Cartas de Personagens. Estas contam um pouco sobre eles e destacam uma característica que faz referência a cada um dos 4 pilares do Pensamento Computacional. A utilização de desenhos alegres e divertidos, juntamente de cores vibrantes nas cartas também está alinhada com a abordagem lúdica do jogo, tornando a experiência mais agradável e estimulante para as crianças.



Figura 79. Cartas de Personagens.
Fonte: O autor (2024).

Buscando garantir ainda a necessidade de personalização, foram criadas cartas com um espaço para o desenho de um personagem, juntamente com um espaço para a sua descrição. As Cartas de

Personagem Personalizáveis permitem que um jogador desenhe facilmente outros personagens não contemplados pelo jogo, possibilitando que o jogador crie novas histórias e enredos envolventes, de forma que elas cumpram seu papel educativo de forma eficaz, ao mesmo tempo em que mantêm o jogo divertido, envolvente e personalizável neste aspecto. Assim como os demais personagens são representados por pinos no tabuleiro, as Cartas Personalizáveis serão acompanhadas de pinos Personalizáveis.

As Cartas de Personagens e as Cartas de Personagens Personalizáveis não devem fazer parte do baralho do jogo e devem ser guardadas durante o jogo pelo respectivo jogador que a escolheu. Esta serve para indicar o seu pino no tabuleiro, caso o jogador esqueça.



Figura 80. Carta de Personagem Personalizável.
Fonte: O autor (2024).

4.5 Jogabilidade

Nesta primeira versão do jogo Codezônia, haverá apenas o modo de jogo Velocidade. Neste modo de jogo, os jogadores competem para completar uma sequência lógica ou um algoritmo no menor tempo possível. O tabuleiro apresenta um percurso com casas sequenciais, representando etapas do algoritmo ou elementos da sequência. Cada jogador possui um marcador, chamado também de pino, que avança pelo percurso conforme eles corretamente constrói as sequências lógicas.

O objetivo é chegar ao centro do tabuleiro primeiro que os adversários, completando todas as etapas corretamente. Este modo de jogo promove a velocidade de raciocínio e a capacidade de

executar sequências lógicas, incentivando os jogadores a pensar rapidamente e tomar decisões estratégicas.

4.5.1 Regras de Jogo

O ponto de partida, sempre será as extremidades do tabuleiro, podendo o jogador optar pela rota principal ou pela rota alternativa.

INÍCIO: Cada jogador iniciará com 5 cartas, a ser sorteado do baralho de cartas. Ao tirar a sorte, um jogador deve iniciar, seguindo pelo próximo a sua esquerda. O objetivo é montar a sequência Lógica utilizando as 5 cartas disponíveis.

Após o início do jogo, o jogador pode puxar uma carta do baralho e descartar outra, não podendo permanecer com mais de cinco cartas em mãos. As cartas descartadas por jogadores devem ser reintroduzidas no baralho de forma aleatória ou ao final da pilha.

O baralho de cartas deve ser colocado em uma das extremidades do tabuleiro, de forma que fique acessível a todos. Caso necessário, pode ser particionada em dois blocos e posicionadas em 2 extremidades diferentes, afim de facilitar o acesso de todos os jogadores.

COMO AVANÇAR AS CASAS: Para avançar no tabuleiro, o jogador precisa montar a sequência correta para uma jogada que lhe fará percorrer 5 passos pelo tabuleiro. Para isto, deve usar as CARTAS DE AÇÃO (Avançar, Virar à Direita, Virar à Esquerda, Avançar Esquerda e Avançar Direita).



1 Jogada com 5 cartas



Figura 81. Exemplo de jogadas com 5 cartas.
Fonte: O autor (2024).

Ao montar a sequência lógica para os 5 passos, na sua vez de jogar, o jogador deve executar a sequência, apresentando as cartas sequencialmente organizadas para os demais jogadores e

posicionando o seu respectivo pino na casa final da sequência lógica construída, descartando todas as cartas que devem voltar para o baralho de cartas de forma aleatória ou no final da pilha.

CARTAS DE REPETIÇÃO: As cartas de repetição permitem repetir CARTAS DE AÇÃO por algumas vezes. Desta forma, é possível fazer uma jogada com uma carta de repetição 5X + uma carta de ação AVANÇAR, por exemplo.



Jogada com 5 cartas, usando Carta de Repetição.



Figura 82. Exemplo de uso de Carta de Repetição.
Fonte: O autor (2024).

O QUE FAZER SE FICAR SEM CARTAS: Para fazer outra jogada, o jogador deve pegar cartas do baralho, sempre na sua vez e individualmente, até somar novamente 5 cartas para realizar outra jogada e percorrer o tabuleiro por mais 5 passos. O objetivo é chegar ao centro do tabuleiro, completando todas as casas da sequência lógica.

OBSTÁCULOS: Algumas cartas permitem colocar obstáculo na rota de um jogador adversário. Quando isto acontece, o jogador prejudicado só poderá seguir pela rota após usar uma carta de TIRAR OBSTÁCULO. Cartas de OBSTÁCULO e TIRAR OBSTÁCULO são escassas, devem ser usadas para impedir um jogador de ganhar muita vantagem ou de finalizar a sequência. Outra opção para o jogador prejudicado é começar as jogadas por outra rota.

Quando invocada, a carta de Obstáculo dá direito ao jogador que à invocou de posicionar um pino de Obstáculo em uma casa da rota do seu adversário.

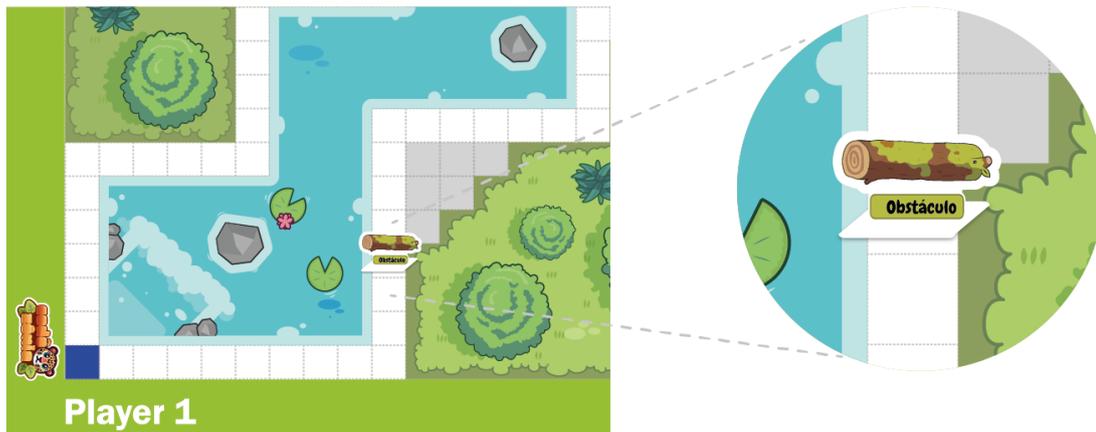


Figura 83. Pino posicionado na rota de um jogador.
Fonte: O autor (2024).

A carta de Obstáculo pode ser invocada em qualquer rodada, mas somente na vez do jogador que a detém. O pino de Obstáculo deve ser posicionado na exata casa desejada. Após invocada e o pino posicionado conforme a escolha do jogador que detém a carta, ela deve voltar ao baralho de cartas. O pino de Obstáculo pode ser posicionado inclusive nas casas de salto.

Para ultrapassar o obstáculo, o jogador prejudicado deve invocar a Carta de Remover Obstáculo.

CARTAS DE SALTO: Algumas cartas permitem vantagens, como as de salto. Elas encurtam o caminho e somente podem ser usadas nas rotas alternativas.

Para utilizar as Cartas de Salto, o jogador precisa estar na entrada de uma casa de salto e possuir uma carta de Condição.

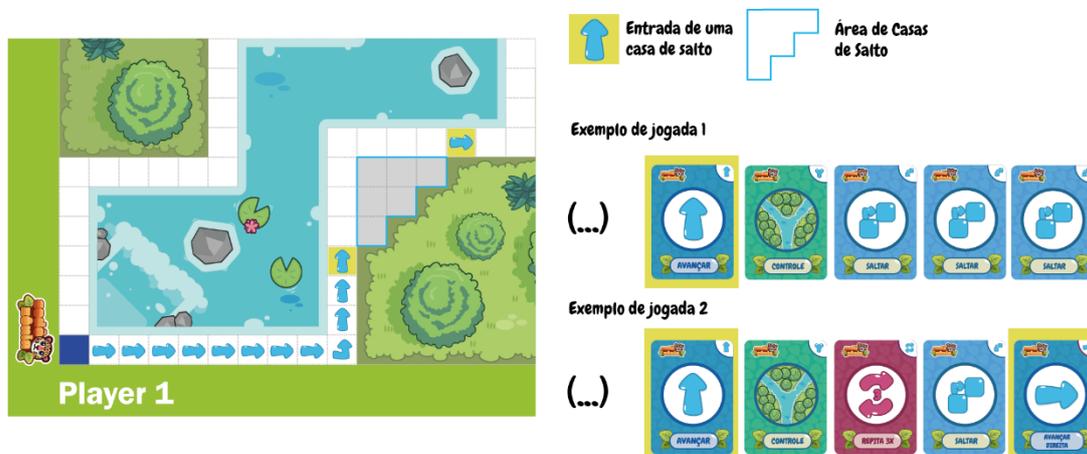


Figura 84. Exemplo de uso da Carta de Controle e Cartas de Salto.
 Fonte: O autor (2024).

CONDIÇÃO DE VITÓRIA: Ganha o jogador que chegar primeiro ao centro do tabuleiro.

4.6 Pensamento Computacional e os conceitos de Programação no Codezônia

Os elementos do jogo e toda a jogabilidade foi idealizada para desenvolver habilidades e competências do Pensamento Computacional, representado pelos 4 pilares: Abstração, Decomposição, Reconhecimento de Padrões e Algoritmos.

A habilidade de Abstração pode ser percebida no jogo através dos recursos e jogadas possíveis. Os jogadores precisam selecionar e combinar os comandos disponíveis de forma abstrata para criar sequências lógicas que os levem ao objetivo. Isso incentiva os jogadores a pensar de forma abstrata, concentrando-se nos passos necessários para alcançar um resultado sem se preocupar com os detalhes de como cada comando é executado.

A habilidade de Decomposição, pode ser percebida quando os jogadores enfrentam o desafio de percorrer todo o trajeto para alcançar o objetivo central do tabuleiro. Eles precisam dividir esse desafio em partes menores, usando sequências de até 5 cartas no máximo. Isso permite que eles decomponham o desafio principal em tarefas menores e mais gerenciáveis, facilitando a construção de algoritmos que os levem ao objetivo final.

O Reconhecimento de padrões, por sua vez, é evidente no jogo quando os jogadores percebem que a construção das sequências lógicas segue um padrão repetitivo. Eles aprendem a identificar

esses padrões e a aplicá-los em diferentes partes do tabuleiro, o que os ajuda a resolver problemas de forma mais eficiente e rápida.

E enfim, a construção de Algoritmos é a habilidade central no jogo, pois os jogadores precisam desenvolver sequências lógicas de comandos para percorrer as casas do tabuleiro e alcançar o objetivo. Eles aprendem a pensar de forma algorítmica, estruturando uma série de passos ordenados e lógicos que os levam à solução do problema. Essa habilidade é fundamental na programação, onde algoritmos são essenciais para resolver problemas computacionais complexos.

Além das habilidades e competências do Pensamento Computacional, é importante destacar os conceitos centrais ligados a Algoritmos e Programação, como Sequência Lógica, Estrutura de Controle e Estrutura de Repetição.

A Sequência Lógica é essencial em Codezônia, pois os jogadores devem montar uma sequência de Cartas de Ação para mover seus personagens pelo tabuleiro. Cada movimento precisa ser pensado de forma lógica e sequencial para alcançar o objetivo final. Saber construir uma sequência lógica e estruturada é o que permite que um jogador se mova pelo tabuleiro e alcance o objetivo final.

As Estruturas de Repetições são introduzidas de forma implícita no jogo, especialmente através das Cartas de Repetição. Desta forma, o jogador pode visualizar o conceito de repetir passos, por vezes limitadas e predeterminadas. No jogo, o aprendizado deste conceito dá ao jogador vantagens e conhecimento para construir uma sequência lógica maior com menos recursos de Cartas de Ação.

As Estruturas de Controle tendem a ser as mais complexas dos 3 conceitos principais. Contudo, ao adaptar para um jogo no qual o público-alvo são crianças, preferiu-se simplificar o conceito. Desta forma, as Estruturas de Controle, implementadas por meio das Cartas de Controle, possibilitam que um jogador faça um teste lógico a partir de uma única preposição: “Estou na entrada de uma casa de salto?”. Se SIM, o jogador poderá utilizar cartas específicas para realizar o salto. Se NÃO, o jogador não pode realizar o salto e deve seguir pelo percurso padrão.

4.7 Cenários de Uso

Codezônia pode ser utilizado de forma eficaz em diversos contextos educacionais, proporcionando benefícios significativos para os alunos.

4.7.1 Salas de aula

O jogo pode ser introduzido como uma atividade complementar em aulas de informática, para auxiliar no ensino de conceitos de Algoritmos e Programação. Os alunos podem jogar em grupos,

onde cada grupo representa um personagem do jogo, incentivando a colaboração e o trabalho em equipe. Exemplo: Os alunos devem desenvolver uma sequência lógica para alcançar o objetivo central do tabuleiro, discutindo e decidindo juntos as melhores estratégias.

4.7.2 Clubes de Programação

Em clubes ou grupos de programação, o jogo pode ser utilizado como uma forma de praticar e reforçar os conceitos aprendidos nas aulas. Os alunos podem criar seus próprios desafios e compartilhar com os colegas, estimulando a criatividade e a resolução de problemas. Por exemplo, os alunos podem modificar as regras do jogo para incluir novas cartas e desafios, explorando diferentes possibilidades de programação.

4.7.3 Competições

O Codezônia apresenta-se como uma boa ferramenta para cenários de competições entre grupos ou individuais. Para isto será necessário mais um exemplar do jogo, de forma a possibilitar os formatos classificatórios. Os vencedores de cada tabuleiro são classificados para fases seguintes até chegar à última rodada com os 4 melhores jogadores da competição, dentre eles, se confirmando o campeão da turma.

Os alunos podem ainda participar de torneios ou competições do jogo "Codezônia" entre diferentes escolas, incentivando a prática e o aprimoramento das habilidades de programação.

4.7.4 Eventos Educacionais

Em eventos educacionais, como feiras de ciências ou semanas temáticas, o jogo pode ser utilizado como uma atividade interativa para demonstrar conceitos de Algoritmos e Programação de forma acessível e envolvente. Uma possibilidade para eventos consiste em montar um estande interativo onde os visitantes podem jogar o Codezônia e aprender sobre programação de maneira divertida.

Em todos esses cenários, o jogo oferece benefícios como o desenvolvimento do pensamento lógico, o estímulo à criatividade, o aprendizado prático de conceitos complexos de forma simplificada, a promoção da resolução de problemas e o desenvolvimento de habilidades de colaboração e trabalho em equipe. Além disso, o jogo também pode ajudar a despertar o interesse dos alunos pela ciência da computação e tecnologia, preparando-os para os desafios do mundo digital.

4.8 Validação da Solução e refinamento

Um teste piloto do jogo Codezônia está programado para ser realizado em uma escola parceira, envolvendo crianças do Ensino Fundamental 1, especialmente Séries Iniciais. O objetivo principal deste teste é validar a solução proposta, verificando sua eficácia no ensino de Algoritmos e Programação de forma lúdica e envolvente.

Durante o teste piloto, os alunos serão divididos em grupos para jogar o "Codezônia" em sala de aula, sob a supervisão de um professor e este pesquisador. Serão observados aspectos como a compreensão das regras do jogo, a capacidade de resolver desafios propostos e o nível de engajamento dos alunos.

Além das observações diretas, serão aplicados questionários e entrevistas para coletar feedbacks sobre a experiência dos alunos com o jogo. Isso permitirá identificar pontos fortes e áreas de melhoria no jogo, bem como possíveis ajustes necessários para torná-lo mais eficaz no contexto educacional.

Com base nos resultados do teste piloto, serão realizados ajustes e refinamentos na proposta do jogo. Esses ajustes podem incluir alterações nas instruções do jogo para torná-las mais claras e acessíveis aos alunos, bem como a inclusão de novos elementos que possam enriquecer a experiência educativa proporcionada pelo jogo.

O processo de refinamento será conduzido com a participação ativa da professora do Laboratório de Inovações, pesquisador e alunos envolvidos no teste piloto. O objetivo é garantir que o jogo atenda às necessidades educacionais e às expectativas dos alunos, proporcionando uma experiência de aprendizado significativa e estimulante.

Ao final do processo de refinamento, espera-se ter uma versão mais refinada do jogo Codezônia que seja eficaz, envolvente e adequada para o ensino de Algoritmos e Programação para crianças do Ensino Fundamental 1.

Almeja-se ainda apresentar os resultados desta aplicação em um trabalho científico e publicar em uma revista de alto impacto, com o objetivo de expandir o alcance deste recurso no ensino dos conceitos da Ciência da Computação.

CAPÍTULO 5

Considerações finais

Neste capítulo apresentamos as considerações finais desta pesquisa, bem como damos luz sobre pesquisas futuras a serem realizadas.

A pesquisa realizada concebeu um jogo de tabuleiro e cartas, que utiliza elementos lúdicos baseados no contexto sociocultural amazônico para ensinar conceitos de programação para crianças a partir de 6 anos, em idade escolar padrão do Ensino Fundamental 1: Séries Iniciais. Através da metodologia do Design Thinking, utilizando-se de diferentes técnicas design, o produto foi construído centrado no usuário, a partir da identificação de necessidades no processo de ensino e aprendizagens de conceitos de Algoritmos e Programação dentro de sala de aula.

A pesquisa contribuirá para expandir o conhecimento de conceitos da Ciência da Computação entre crianças, por meio de recursos lúdicos e desplugados, demonstrando sua viabilidade para ser aplicado em ambientes de sala de aula. O resultado obtido apresenta o jogo Codezônia como uma ferramenta flexível, capaz de proporcionar experiências competitivas e envolventes. Além disso, o jogo é especialmente vantajoso para áreas remotas do estado do Amazonas, onde recursos digitais para o ensino desses conceitos são limitados, permitindo que o conhecimento alcance essas regiões distantes.

Durante a pesquisa, foi possível aprender sobre as diversas técnicas de levantamento de requisitos centrados no usuário, bem como as diferentes formas de analisar a massa de dados geradas pelas etapas da metodologia do Design Thinking. Neste sentido, destaca-se que o processo de concepção não é um processo linear e centralizado, mas iterativo, incremental, multidisciplinar e colaborativo. Para tornar isto possível, o Designer deve planejar as ações considerando como fundamental e imprescindível a participação do usuário final durante todo o processo de concepção do produto, garantindo que o resultado da pesquisa seja um produto desejável, usável e prazeroso.

Portanto, o jogo Codezônia concebido a partir de um processo de design centrado no usuário mostra-se como uma alternativa promissora para uso com recurso lúdico para ensino de conceitos de Programação para crianças a partir de 6 anos. Os resultados sugerem um caminho promissor para a confecção deste produto em escala industrial, com o fim de expandir os conceitos de Ciência da Computação para mais indivíduos.

Sugestões para trabalhos futuros

Nesta pesquisa os resultados produzidos na concepção do jogo Codezônia para o ensino dos conceitos de Algoritmo e Programação para crianças, foram satisfatórios.

Portanto, acredita-se que o jogo possui diversos pontos favoráveis para investigações em pesquisas futuras, tais como:

- Construção de outros cenários, outros personagens e outros modos de jogo, tais como Coleta de itens.
- Investigação a respeito de outras regras para aumentar o grau de aplicação dos conceitos de Sequência Lógica, Estrutura de Controle e Estrutura de Repetição em diferentes níveis, de forma a possibilitar a construção do conhecimento progressivo.

REFERÊNCIAS BIBLIOGRÁFICAS

- Alben, L. (1996). Quality of experience: defining the criteria for effective interaction design. *interactions*, 3(3), 11-15.
- Alves, B., Romani, L. A. S., & Otavian, A. F. (2016). AgritempoGIS: um aplicativo para auxiliar agricultores em processos de tomada de decisão. In: MOSTRA DE ESTAGIÁRIOS E BOLSISTAS DA EMBRAPA INFORMÁTICA AGROPECUÁRIA, 12., 2016, Campinas. Resumos expandidos... Brasília, DF: Embrapa, 2016. Disponível em: <<http://www.alice.cnptia.embrapa.br/alice/handle/doc/1065064>>. Acesso em: 22 de jun. 2022.
- AP Program (2016). AP Computer Science Principles. Disponível em: <<https://apcentral.collegeboard.org/courses/ap-computer-science-principles>>. Acesso em: 07 de abr. 2022.
- Araújo, D., Rodrigues, A., Silva, C., & Soares, L. (2015). O Ensino de Computação no Ensino Fundamental 1: Séries Iniciais apoiado por Problemas: Práticas de Licenciandos em Computação. In *Anais do XXIII Workshop sobre Educação em Computação*, (pp. 130-139). Porto Alegre: SBC. doi:10.5753/wei.2015.10229
- Araújo, G. de O., Rocha, A. C. B., & Prata, W. S. (2022). Hotel Reform: reflections on emerging innovations in the Covid-19 pandemic. *DAT Journal*, 7(2), 210-221. <https://doi.org/10.29147/datjournal.v7i2.612>
- Azevedo, P. M. de, & Gibertoni, D. (2020). A IMPORTÂNCIA DO DESIGN CENTRADO NO USUÁRIO EM METODOLOGIAS AGEIS COMO REQUISITO DE USABILIDADE. *Revista Interface Tecnológica*, 17(2), 293-305. <https://doi.org/10.31510/infa.v17i2.986>
- Barcelos, T. S., & Silveira, I. F. (2012). Pensamento computacional e educação matemática: Relações para o ensino de computação no Ensino Fundamental 1: Séries Iniciais. In *XX Workshop sobre Educação em Computação*, Curitiba. Anais do XXXII CSBC (Vol. 2, p. 23).
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?. *ACM Inroads*, Volume 2, Issue 1 March 2011 pp 48-54 <https://doi-org.ez2.periodicos.capes.gov.br/10.1145/1929887.1929905>
- Base Nacional Comum Curricular, 2018. Disponível em: <chrome-extension://efaidnbmnnnibpcajpcgclefindmkaj/http://basenacionalcomum.mec.gov.br/imagens/BNCC_EI_EF_110518_-versaofinal_site.pdf>. Acesso em: 12 de mar. 2022.
- Batista Correia, M. da C. (2009). A observação participante enquanto técnica de investigação. *Pensar Enfermagem*, 13(2), 30-36. <https://doi.org/10.56732/pensarenf.v13i2.32>
- Benedet, D. C. F., Walll, M. L., Lacerda, M. R., Souza, S. R. R. K., Trigueiro, T. H., & Thuler, A. C. D. M. C. (2019). A observação não participante no reconhecimento da competência profissional. *CLAIQ2019*, 2, 1286-1291.
- Berry, M. (2013). Computing in the national curriculum: A guide for primary teachers.
- Blikstein, Paulo. 2008. O pensamento computacional e a reinvenção do computador na educação. Disponível em <http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html> Acesso em: 05 de jun. 2022.
- Bogdan, R. & Biklen, S. (1994). Notas de campo. In: R. Bogdan & S. Biklen, *Investigação qualitativa em educação -uma introdução às teorias e aos métodos*. (150-175). Porto: Porto Editora.
- Bonini, L. A., & Sbragia, R. (2011). O modelo de design thinking como indutor da inovação nas empresas: um estudo empírico. *Gestão e Projetos: GeP*, 2(1), 3-25.
- Bordini, A., Avila, C. M. O., Weissshahn, Y., Cunha, M. M. da, Cavalheiro, S. A. da C., Foss, L., Aguiar, M. S., & Reiser, R. H. S. (2016). Computação no Ensino Fundamental 1: Séries Iniciais no Brasil: o Estado da Arte. *Revista De Informática Teórica E Aplicada*, 23(2), 210-238. <https://doi.org/10.22456/2175-2745.64431>
- Brackmann, C. P. 2017. Desenvolvimento do pensamento computacional através de atividades desplugadas no Ensino Fundamental 1: Séries Iniciais. Disponível em: <<http://hdl.handle.net/10183/172208>>. Acesso em: 8 de fev. 2022.
- Brackmann, C., Boucinha, R. M., Román-González, M., Barone, D. A. C., & Casali, A. (2017b, October). Pensamento computacional desplugado: Ensino e avaliação na educação primária espanhola. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 6, No. 1, p. 982). <http://dx.doi.org/10.5753/cbie.wcbie.2017.982>

- Brangier, É., & Barcenilla, J. (2003). *Concevoir un produit facile à utiliser*. Paris: Editions d'organisation.
- BRASIL. Ministério da Educação. Base Nacional Comum Curricular – BNCC 2ª versão. Brasília, DF, 2016.
- Brown, T. (2009). *Change by design: how design thinking transforms organizations and inspires innovation*. New York: HarperCollins.
- Brown, T. (2010). *Design Thinking: uma metodologia poderosa para decretar o fim das velhas ideias*. Alta Books.
- BUNDY, A. 2007. Computational Thinking is Pervasive. *Journal of Scientific and Practical Computing*, v. 1, p. 67–69. Disponível em: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://core.ac.uk/download/pdf/28961399.pdf>. Acesso em: 23 de jun. 2022.
- Cafezeiro, Isabel; Costa, Leonardo Cruz da e Kubrusly, Ricardo da Silva. *Ciência da Computação, Ciência da Informação, Sistemas de Informação: uma reflexão sobre o papel da informação e da interdisciplinaridade na configuração das tecnologias e das ciências. Perspectivas em Ciência da Informação [online]*. 2016, v. 21, n. 03 [Acessado 23 Junho 2022], pp. 111-133. Disponível em: <https://doi.org/10.1590/1981-5344/2681>. ISSN 1981-5344. <https://doi.org/10.1590/1981-5344/2681>.
- Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, 15(4), 9-14.
- CAS, C. AT S. (2014). *Developing Computational Thinking*. Teaching London Computing. Disponível em: <<http://teachinglondoncomputing.org/resources/developing-computational-thinking/>>. Acesso em: 02 de out. 2022.
- Chapman, C. N., & Milham, R. P. (2006, October). The personas' new clothes: methodological and practical arguments against a popular method. In *Proceedings of the human factors and ergonomics society annual meeting (Vol. 50, No. 5, pp. 634-636)*. Sage CA: Los Angeles, CA: SAGE Publications.
- CIMA (1996). *Management Accounting Official Terminology*, CIMA, Portland, pp.20.
- Computer Science Teachers Association – CSTA (2011) "Computational thinking: teacher resources." 2. ed. Disponível em: <https://cdn.iste.org/www-root/2020-10/ISTE_CT_Teacher_Resources_2ed.pdf>. Acesso em: 24 de jun. 2022.
- Contel, F. B. (2014), « As divisões regionais do IBGE no século XX (1942, 1970 e 1990) », *Terra Brasilis (Nova Série)* [Online], 3, posto online no dia 26 agosto 2014, consultado o 22 junho 2022. URL : <http://journals.openedition.org/terrabrasilis/990> ; DOI : <https://doi.org/10.4000/terrabrasilis.990>
- Correia, M. M. P. D. R. K. D. (2012). *Integração dos recursos educativos digitais no 1º ciclo do ensino básico: uma realidade ou uma utopia (Doctoral dissertation)*.
- Cuny, J., Snyder, L., & Wing, J. M. (2010). *Demystifying computational thinking for non-computer scientists*. Disponível em: <<http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>>. Acesso em: 22 de jun. 2022.
- de Diretrizes, L. (1996). *bases da Educação Nacional*.
- Del-Claro, K., Torezan-Silingardi, H. M., Belchior, C., & Silva, E. A. (2009). *Ecologia Comportamental: uma ferramenta para a compreensão das relações animais-plantas*. *Oecologia Brasiliensis*, 13(1), 16-26 Disponível em: <<https://dialnet.unirioja.es/servlet/articulo?codigo=2883467>>. Acesso em: 23 de mai. 2022.
- Denning, P. J. (2005). Is computer science science?. *Communications of the ACM*, 48(4), 27-31. Disponível em: <https://dl.acm.org/doi/abs/10.1145/1053291.1053309?casa_token=Wb0-WrwKtmMAAAAA:EeAXsWX04-6v2Dmggy5IRh2qL7bs4KCgkBi8bmFZB08a-z7yubRYW-B8ov7kXL1NIZ8GDwYtlArqQA>. Acesso em: 5 de jun. 2022.
- Denning, P. J. 2009. The profession of IT Beyond computational thinking. *Communications of the ACM*, 52(6), 28-30. Disponível em: <https://dl.acm.org/doi/fullHtml/10.1145/1516046.1516054?casa_token=7TW6qAC0czUAAAAA:ig21M1mBB7WUYI8HKeedCWdEeM9F3sL3UThkUZ0eBtjSDLGbnZWl0mEYg4JQ2bepq7VHUVGRnaQLzQ>. Acesso em: 23 de mai. 2022.
- do Santos, E. R., Soares, G., Dal Bianco, G., da Rocha Filho, J. B., & Lahm, R. A. (2016). *Estímulo ao Pensamento Computacional a partir da Computação Desplugada: uma proposta para Educação Infantil*. *Revista Latinoamericana de Tecnología Educativa*.
- Galvão, M. C. B., & Ricarte, I. L. M. (2019). *Revisão sistemática da literatura: conceituação, produção e publicação*. *Logeion: Filosofia da informação*, 6(1), 57-73.

- Falcão, T. P., Gomes, T. C. S., & Albuquerque, I. R. (2015). O pensamento computacional através de jogos infantis: uma análise de elementos de interação. *Anais do XVI IHC-Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais-IHC*.
- Forlizzi, J., & Ford, S. (2000, August). The building blocks of experience: an early framework for interaction designers. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques* (pp. 419-423). <https://doi.org/10.1145/347642.347800>
- França, R., & Tedesco, P. (2015, October). Desafios e oportunidades ao ensino do pensamento computacional no Ensino Fundamental 1: Séries Iniciais no Brasil. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 4, No. 1, p. 1464). DOI: <http://dx.doi.org/10.5753/cbie.wcbie.2015.1464>
- Garlet, D., Bigolin, N. M., & Silveira, S. R. (2018). Ensino de Programação de Computadores no Ensino GIMPS, 2016. Disponível em: <https://www.merzenne.org/>. Acesso em: 15 de jun. 2022.
- Gödel, K. 1995. On formally undecidable propositions of principia mathematica and related systems. In: DAVIS, M. (Org). *The undecidable: basic papers on undecidable propositions, unsolvable problems and computable functions*. New York: Dover Publications, 1965. p. 4-38.)
- Gomes, T., & Alencar, A. (2015). Análise Empírica de Jogos Educativos para Dispositivos Móveis voltados a Disseminação do Pensamento Computacional no Ensino Fundamental 1: Séries Iniciais. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 4, No. 1, p. 731).
- Gomes, T., Melo, J., & Tedesco, P. (2016, November). Jogos digitais no ensino de conceitos de programação para crianças. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 27, No. 1, p. 470).
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational researcher*, 42(1), 38-43.
- Guarda, G., & Pinto, S. (2020). Dimensões do Pensamento Computacional: conceitos, práticas e novas perspectivas. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, (pp. 1463-1472). Porto Alegre: SBC. doi:10.5753/cbie.sbie.2020.1463
- Fundamental 1: Séries Iniciais: um estudo de caso. *Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica*, 9(2).
- Gaspar, W (2021). Disponível em: <https://wagnergaspar.com/exercicio-1-calculando-a-idade-do-usuario-em-fluxograma/>. Acesso em: 7 de ago. 2022.
- Gerhardt, T. E., Ramos, I. C. A., Riquinho, D. L., & Santos, D. L. D. (2009). Estrutura do projeto de pesquisa. *Métodos de pesquisa*. Porto Alegre: Editora da UFRGS, 2009. p. 67-90.
- Gibson, B., & Bell, T. (2013). Evaluation of games for teaching computer science. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education* (pp. 51-60).
- Gil, A. C. (2002). Como elaborar projetos de pesquisa (Vol. 4, p. 175). São Paulo: Atlas.
- Guzdial, M. (2008). Educação abrindo caminho para o pensamento computacional. *Comunicações da ACM*, 51 (8), 25-27. <https://doi.org/10.1145/1378704.1378713>
- Hartson, R., & Pyla, P. S. (2012). *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier.
- Hartson, Rex, and Pardha S. Pyla. *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier, 2012.
- Henderson, P. B., Cortina, T. J., & Wing, J. M. 2007. Computational thinking. In *Proceedings of the 38th SIGCSE technical symposium on Computer science education* (pp. 195-196). Disponível em: https://dl.acm.org/doi/pdf/10.1145/1227310.1227378?casa_token=Luhb4ExLuYoAAAAA:3bc7tyG0R9oqZtT42w7M1Octe01hu2HpD3xlbU0AjQdUsMJA2pnNGVEL5NAgohpnHBQRHQPOeSIB7A. Acesso em: 22 de jun. 2022.
- Henderson, Peter, et al. 2001. Striving for mathematical thinking. *ITiCSE 2000 Working Group Report, SIGCSE Bulletin - Inroads*, Vol. 33, No. 4.
- Hutchison, A.J.; Johnston, L.H.; Breckon, J.D. (2010). Using QSR-NVivo to facilitate the development of a grounded theory project: an account of a worked example. *International Journal of Social Research Methodology*, 13(4). Doi: <https://doi.org/10.1080/13645570902996301>

- IBGE, 2021. Disponível em: <
<https://www.ibge.gov.br/cidades-e-estados/am.html>>.
Acesso em: 15 de mai. 2022.
- ISO 9241-11. Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)/ Part 11 – Guidance on usability, 1998.
- Kaminski, M. R., Klüber, T. E., & Boscaroli, C. (2021). Pensamento computacional na educação básica: Reflexões a partir do histórico da informática na educação brasileira. *Revista Brasileira de Informática na Educação*, 29, 604-633.
- Kaplan CA, Simon HA. 1990. In search of insight. *Cogn. Psychol.* 22:374–419
- Karadeniz, S., Samur, Y., & Ozden, M. Y. (2014). Playing with Algorithms to Learn Programming A Study on 5 Years Old Children. In *International Conference on Information Technology and Applications* (Vol. 9).
- Kenski, V. M. (2003). Aprendizagem mediada pela tecnologia. *Revista diálogo educacional*, 4(10), 1-10.
- Kramer, Jeff. 2007. Is Abstraction the Key to Computing? *CACM*, Vol 50, No 4.
- Kuniavsky, M. (2010). *Smart things: ubiquitous computing user experience design*. Elsevier.
- LDB : Lei de diretrizes e bases da educação nacional. – Brasília : Senado Federal, Coordenação de Edições Técnicas, 2017. 58 p.
- Leal da Silva Barbosa, L., & Maltempi, M. (2020). Matemática, Pensamento Computacional e BNCC: desafios e potencialidades dos projetos de ensino e das tecnologias na formação inicial de professores. *Revista Brasileira De Ensino De Ciências E Matemática*, 3(3).
<https://doi.org/10.5335/rbecm.v3i3.11841>
- Levy, J. (2015). UX strategy: How to devise innovative digital products that people want. " O'Reilly Media, Inc."
- Lévy, P. (1995). *Tecnologias da inteligência*. São Paulo: Editora 34.
- Lévy, P. (1996) *O que é o virtual?* São Paulo: Editora 34
- Lévy, P. (1998). *A máquina universo – criação, cognição e cultura informática*. Trad. de Bruno Charles Magno. Porto Alegre: Artmed;
- Lévy, P. (1999). *Cibercultura*. Trad. de Carlos Irineu Costa. São Paulo: Editora 34;
- Lin, K. Y., Chien, C. F., & Kerh, R. (2016). UNISON framework of data-driven innovation for extracting user experience of product design of wearable devices. *Computers & Industrial Engineering*, 99, 487-502.
<https://doi.org/10.1016/j.cie.2016.05.023>
- Liukas, L. (2015). *Hello Ruby: adventures in coding* (Vol. 1). Macmillan.
- Lopes, M. J. M. (1993). *Les soins, images et réalités: le quotidien soignant au Brésil* (Doctoral dissertation, Paris 7).
- Luftman, J. N. 1996. Applying the strategic alignment model. *Competing in the information age: Strategic alignment in practice*, 43-69.
- Madeira, P. (1999). *Benchmarking: a arte de copiar*. *Jornal do Técnico de Contas e da Empresa (JTCE)*, 364-367.
- Magin, D. P., Maier, A., & Hess, S. (2015, August). Measuring negative user experience. In *International Conference of Design, User Experience, and Usability* (pp. 95-106). Springer, Cham.
- Marconi, M. D. A., & Lakatos, E. M. (2003). *Fundamentos de metodologia científica*. atlas.
- Marietto, M. L. (2018). Observação participante e não participante: contextualização teórica e sugestão de roteiro para aplicação dos métodos. *Revista Ibero Americana de Estratégia*, 17(4), 05-18.
- Marques, M.T.P.M (2009). *Recuperar o engenho a partir da necessidade, com recurso às tecnologias educativas: contributo do ambiente gráfico de programação Scratch em contexto formal de aprendizagem*. Dissertação de mestrado não publicada. Universidade de Lisboa, Faculdade de Psicologia e de Ciências da Educação.
- McGARRY, K. J. *Da documentação à informação: um conceito em evolução*. Lisboa: Editorial Presença, 1984. 196p.
- McMaster, K., Rague, B., & Anderson, N. (2010, October). Integrating mathematical thinking, abstract thinking, and computational thinking. *IEEE Frontiers in Education Conference (FIE)*, 2010, pp. S3G-1-S3G-6, doi: 10.1109/FIE.2010.5673139. Disponível em: <<https://ieeexplore.ieee.org/document/5673139/metrics#metrics>>. Acesso em: 24 de jun. 2022.
- MJV Team. **MJV Innovation**, 2022. Disponível em: <https://www.mjvinnovation.com/pt-br/blog/como-utilizar-os-cartoes-de-insight/>. Acesso em: 10 jun. 2023.

- Morales, A. (2013). Programação de computadores para melhoria da aprendizagem. *Cadernos de Gestão e Empreendedorismo*, 1(3), 1-15.
- Moran, J.M. (2002). Educação inovadora na Sociedade da Informação. In: REUNIÃO ANUAL DE PÓS-GRADUAÇÃO E PESQUISA EM EDUCAÇÃO 25., Anais...Caxambu.
- Norman, D. A., Draper S. W. (1986). *User Centered System Design: New Perspectives on Human-Computer Interaction*. New Jersey: Lawrence Erlbaum Associates, 1986
- Norman, D. A. (1986). Engenharia Cognitiva. Em DA Norman & SW Draper (Eds.). Projeto de sistema centrado no usuário: Nova perspectiva na interação humano-computador (pp. 31-61).. Hillsdale, NJ: L. Erlbaum Associates.
- Nunes, D. J. Ciência da Computação no Ensino Fundamental 1: Séries Iniciais. ADUFRGS - Sindical, 6. jun. 2011. Disponível em: <<http://gestaouniversitaria.com.br/artigos/ciencia-da-computacao-na-educacao-basica--3>>. Acesso em: 23 de jun. 2022.
- Okamoto, Edgar; Paiva, Rodrigo Ernandes de, 2005. "UTILIZAÇÃO DA COMPUTAÇÃO GRÁFICA NO CINEMA." V EPCC - Encontro Internacional de Produção Científica (23 a 26 de Outubro de 2007). Disponível em: <<http://rdu.unicesumar.edu.br/handle/123456789/7066>>. Acesso em: 22 de jun. 2022.
- Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism*. (pp. 1–11). Norwood, NJ: Ablex.
- Papert, S. 1980. *Children, computers, and powerful ideas*. Harvester Press (Unitend Kingdom). DOI, 10, 978-3.
- Papert, S., Solomon, C., Soloway, E., & Spohrer, J. C. 1971. Twenty things to do with a computer. Studying the novice programmer, 3-28. Disponível em: <https://www.researchgate.net/publication/37596692_Twenty_Things_to_Do_with_a_Computer> . Acesso em: 22 de jun. 2022.
- Peixoto, J., & Araújo, C. H. D. S. (2012). Tecnologia e educação: algumas considerações sobre o discurso pedagógico contemporâneo. *Educação & Sociedade*, 33, 253-268. <https://doi.org/10.1590/S0101-73302012000100016>
- Petzold, C. (2008). *The annotated Turing: a guided tour through Alan Turing's historic paper on computability and the Turing machine*. Wiley Publishing.
- Piaget, J. (1968). *La psychologie de l'intelligence*. Paris: A. Colin;
- Piaget, J. (1973). *Estudos sociológicos*. Rio de Janeiro: Forense.
- Pinheiro, L. V. R. 2014. Informação: esse obscuro objeto da Ciência da Informação. *Revista Morpheus - Estudos Interdisciplinares Em Memória Social*, 3(4). Disponível em: <<http://seer.unirio.br/morpheus/article/view/4108>>. Acesso em: 22 de jun. 2022.
- Pinto, A. S. (2010). *Scratch na aprendizagem da Matemática no 1.º Ciclo do Ensino Básico: estudo de caso na resolução de problemas (Doctoral dissertation)*.
- Prodanov, C. C., & De Freitas, E. C. (2013). *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição*. Editora Feevale.
- Quaresma, M. (2018). *UX Designer: quem é este profissional e qual é a sua formação e competências?. Design para acessibilidade e inclusão*. São Paulo: Blucher, 88-101. DOI: 10.5151/9788580393040-07
- Reasoning, ed. K Holyoak, R Morrison, pp. 475–91. New York: Oxford Univ. Press
- Resolução CNE/CEB 1/2022. *Diário Oficial da União*, Brasília, 6 de outubro de 2022, Seção 1, p. 33
- Royal Society. (2012). *Shut down or restart: The way forward for computing in UK schools*. Disponível em: <<https://royalsociety.org/~media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>>. Acesso em: 07 de abr. 2022.
- SANKARAN, R. 2007. *Burning questions*. US Department of Energy. Disponível em: <<http://ascr-discovery.science.doe.gov/2007/04/burning-questions/>>. Acesso em: 23 de mai. 2022.
- Scaico, P. D., Henrique, M. S., Cunha, F. O. M., & de Alencar, Y. M. (2012). Um relato de experiências de estagiários da licenciatura em computação com o ensino de computação para crianças. *RENOTE*, 10(3).
- Schatzman, L. & Strauss, A. (1973). *Field Research. Strategies for a natural sociology*. Englewoods, Prentice Hall.
- Sitio Scratch. Disponível em: <<https://scratch.mit.edu/>> . Acesso em: 10 de out. 2022.

- Seibt, I. P. D. S. (2019). BNCC-Base Nacional Comum Curricular: a abordagem da Tecnologia da Informação e Comunicação (TIC's) nas séries finais do ensino fundamental.
- Sítio Governo Federal, 2022. Disponível em: <<https://www.gov.br/mcti/pt-br/composicao/rede-mcti/conselho-nacional-de-desenvolvimento-cientifico-e-tecnologico>>. Acesso em: 3 de mar. 2022.
- Sítio VisuAlg. Disponível em: <https://visualg3.com.br/>. Acesso em: 10 de out. 2022.
- Smith RW, Kounios J. 1996. Sudden insight: all-or-none processing revealed by speed-accuracy decomposition. *J. Exp. Psychol.: Learn. Mem. Cogn.* 22(6):1443-62
- Souza, F., Falcão, T., & Mello, R. (2021). O Ensino de Programação no Ensino Fundamental 1: Séries Iniciais: Uma Revisão da Literatura. In *Anais do XXXII Simpósio Brasileiro de Informática na Educação*, (pp. 1265-1275). Porto Alegre: SBC. doi:10.5753/sbie.2021.218461
- SOUZA, P. N. P. DE; SILVA, E. B. DA. Como entender e aplicar a nova LDB: LEI no. 9.394/96. Sao Paulo, Brazil: Cengage Learning, 1997.
- Souza, V. V. Soares (2007). Letramento digital e formação de professores. *Revista Língua Escrita*, n. 2, p. 55-69.
- Spradley, J. P. (1980). *Participant Observation*. Orlando, FL: Harcourt Brace Jovanovich College Publishers.
- Tonin, L.; Lacerda, M.R.; Favero, L.; Nascimento, J.D.; Zagonel, I.P.S. Diário de campo na pesquisa qualitativa de enfermagem: da teoria à prática. In: Lacerda, M.R.; Ribeiro, R.P. y Costenaro, R.G.S. (orgs.) (2018). *Metodologias da pesquisa para a enfermagem em saúde: da teoria à prática*. Volume 2. Porto Alegre: Morá.
- Valente, J. (1993). Diferentes usos do computador na educação. *Em aberto*, 12(57).
- van Steenburgh, J. J., Fleck, J. I., Beeman, M., & Kounios, J. (2012). *Insight*. W: KJ Holyoak, RG Morrison (red.).
- Vianna, M., Vianna, Y., Adler, I. K., Lucena, B., & Russo, B. (2012). *Design thinking: inovação em negócios*. Rio de Janeiro : MJV Press, 2012. 162p. : il. ; 24 cm
- Vygotsky, L.S. (2003). *A formação social da mente: o desenvolvimento dos processos psicológicos superiores*. 5. ed. São Paulo: Martins Fontes.
- Weber, F., & Beaud, S. (2010). *Guide de l'enquête de terrain. Produire et analyser des données*.
- Wing, J. 2011. Research notebook: Computational thinking—What and why. *The link magazine*, 6, 20-23. Disponível em: <<https://people.cs.vt.edu/~kafura/CS6604/Papers/CT-What-And-Why.pdf>>. Acesso em: 22 de jun. 2022.
- Wing, J. 2014 *Computational Thinking with Jeannette Wing*. Columbia Journalism School. Disponível em: <<https://www.youtube.com/watch?v=U67utvZai8s>>. Acesso em: 23 de jun. 2022.
- Wing, J. M. 2006. Computational thinking. *Communications of the ACM*, v. 49, n. 3, p. 33. Disponível em: <https://dl.acm.org/doi/fullHtml/10.1145/1118178.1118215?casa_token=AWy_8ZWxIIQAAAAA:WkjgAruncZkrLQE43aKHpiltlqU0SQlflo1zyGNlfd-sKqumyBgZOr04rwjeiXZ-qwk-ZPUY4Lg1NA>. Acesso em: 22 de jun. 2022.
- Wing, J. M. *Computational Thinking: What and Why?*, 2010. Disponível em: <<https://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf>> Acesso em: 5 de jun, 2022.
- Wing, J. M. 2008 "Computational Thinking and Thinking about Computing.". *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical, and Engineering Sciences* 366.1881: 3717-725. Web. Disponível em:<<https://www.jstor.org/stable/25197357>>. Aceso em: 22 de jun. 2022.
- World Design Organization (2022). Disponível em: <<https://wdo.org/about/definition/>>. Acesso em: 10 de ago. 2022.

APÊNDICE A

FICHA DE OBSERVAÇÃO

Esta ficha tem como objetivo registrar e documentar observações detalhadas de eventos, comportamentos ou fenômenos específicos da pesquisa intitulada HELLO (PAPER) WORLD: DESENVOLVIMENTO DE SOLUÇÕES DESPLUGADAS PARA O ENSINO COMPUTACIONAL NO CONTEXTO SOCIAL AMAZÔNICO. Todos os critérios e dados expressos neste documento são de caráter sigiloso e servirão única e exclusivamente como coletar dados e informações importantes em contextos de Observação Participativa.

Critérios Observáveis
1 - Faixa etária do público de consonância com a pesquisa.
O público observado contempla o público alvo da pesquisa – alunos de 5 a 10 anos, entre a 1º e 5º ano do Ensino Fundamental I?
2 - Conhecimentos prévios sobre Algoritmo e Programação
Os alunos possuem algum tipo de conhecimento prévio sobre os conceitos de Algoritmo e Programação, tais como: Sequência Lógica, Condicionais e Estrutura de Repetição?
3 - Nível do conteúdo apresentado
O Nível do Conteúdo ministrados aos alunos, está adequado ao seu nível de escolaridade, a sua faixa etária, a sua visão de mundo e de acordo com os conhecimentos prévios?
4 - Recurso Lúdico Adequado
Recursos lúdicos foram empregados como recursos didáticos na mediação das atividades de aprendizagem?

5 - Nível de engajamento nas atividades	
Exploração Curiosa: Os alunos estão ativamente envolvidos na atividade, explorando os recursos lúdicos com entusiasmo. Eles fazem perguntas, tocam, experimentam e demonstram um forte interesse pelo que estão fazendo.	
Participação Ativa: Os alunos estão engajados na atividade e interagem com os recursos lúdicos de forma consistente. Eles respondem às diretrizes do professor e estão envolvidos em discussões ou colaboração com os colegas.	
Foco e Concentração: Os alunos demonstram concentração e atenção à atividade, mostrando comprometimento com a tarefa. Eles estão imersos no processo de aprendizado e demonstram um alto nível de envolvimento.	
Criatividade e Exploração Profunda: Alunos neste nível não apenas completam a atividade, mas também exploram novas maneiras de usá-la. Eles aplicam criatividade e pensamento crítico, buscando desafios adicionais ou soluções alternativas.	
Autonomia e Liderança: Alguns alunos podem atingir um nível de engajamento em que são capazes de liderar partes da atividade ou até mesmo ensinar seus colegas. Eles demonstram um domínio significativo do conteúdo e são capazes de aplicá-lo de forma independente.	
6 - Palavras de Feedbacks Positivos	
Alunos expressaram alguma palavra POSITIVA sobre a atividade que estão participando?	

7 - Palavras de Feedbacks Negativas
Alunos expressaram alguma palavra NEGATIVA sobre a atividade que estão participando?
8 - Sentimentos/Sensações Percebidas
Os alunos manifestam algum sentimento, seja por meio do comportamento, ou por meio de palavras?

APÊNDICE B**Diário de Campo****Nota Descritiva**

Observador:			
Data:		Horário:	
Local:			
Características do Espaço físico:			
Descrição do Sujeito			
Quantidade:	Idade média:		M: F:
Atividade			
Descrição da atividade chave: (descrever o objetivo da aula)			
Metodologia de Ensino: (descrever a metodologia, recursos didáticos empregados na aula)			
Descrição de comportamentos: (descrever o que os alunos fizeram, durante a aula)			

Reconstrução de falas: (descrever ou transcrever falas pronunciadas por alunos)	
Emoções evidenciadas: (descrever manifestações de emoções como raiva, descontentamento, satisfação, empenho)	
Comportamentos e comentários do observador: (descrever qualquer observação ou postura não categorizado anteriormente)	

APÊNDICE C

Roteiro de Entrevista

Participante: (Nome completo)

Função:

Pensamento Computacional

1. Pais são sempre curiosos. Como você explicaria sobre o que é Pensamento Computacional para um pai de um aluno?
2. Quais competências e habilidades você acredita que o Pensamento Computacional pode desenvolver nas crianças?
3. Você conhece os **quatro** pilares do Pensamento Computacional?
4. Para você há diferença entre Pensamento Computacional e Programação? Se “Sim”, qual a diferença entre as duas coisas?

Embasamento Teórico

5. Você tem costume de ler artigos científicos?
6. Você acha que a simples utilização de aparatos tecnológicos na sala de aula garante a melhoria do ensino?
7. A partir de qual idade você considera ideal ensinar programação?
8. Na sua opinião, quais os requisitos mínimos que uma criança deve ter para poder ser inserida na programação? *(Considere a faixa etária que você afirmou na pergunta 7)*

Identificação de recurso

9. Me conta sobre as ferramentas (tecnológicas ou não) que você usa para ensinar programação para crianças. *(Considere a faixa etária que você afirmou na pergunta 7)*
10. Quais software/aplicativos são mais bem aceitos pelas crianças? **Cite apenas 3.** *(Considere a faixa etária que você afirmou na pergunta 7)*

11. Você vê algo em comum entre estes softwares/aplicativos?
12. Você considera que estes softwares/aplicativos são adequados para crianças? *(Considere a faixa etária que você afirmou na pergunta 7)*
13. Se tivesse que escolher somente 1 software para ser seu recurso principal no ensino de programação para crianças, qual seria?

Metodologia

14. Sua metodologia de ensino é baseada em algum autor? Se sim, qual?
15. Você prefere ensinar programação de forma Plugada, Desplugada ou Mista? Por quê?
16. Como você desenvolve a habilidade de **Decomposição** com seus alunos? Definição: Processo pelo qual os problemas são quebrados em partes menores (Liukas, 2015).
17. Como você desenvolve a habilidade de **Reconhecimento de Padrões** com seus alunos? Definição: Habilidade que visa encontrar similaridades e padrões com o intuito de resolver problemas complexos de forma mais eficiente (Liukas, 2015).
18. Como você desenvolve a habilidade de **Abstração** com seus alunos? Definição: Habilidade que visa escolher o detalhe a ser ignorado para que o problema seja mais fácil de ser compreendido (CAS, 2014).
19. Como você desenvolve a habilidade de construir **Algoritmos** com seus alunos? Definição: Um plano, uma estratégia ou um conjunto de instruções claras necessárias para a solução de um problema (CSIZMADIA et al., 2015).

Experiência do Usuário

20. Qual a reação dos alunos quando você fala que vai ensiná-los a programar jogos e robôs? *(Considere a faixa etária que você afirmou na pergunta 7)*
21. Quanto tempo em média os alunos permanecem usando o software/aplicativo que você escolheu como recurso principal (pergunta 13), sem perder o interesse?

22. Na maioria dos casos, a sensação das crianças após o uso do software/aplicativo escolhido como recurso principal é de **Satisfação** ou **Frustração**?

23. Você consegue utilizar este software simultaneamente com quantas crianças no máximo?

Design de Interface/Usabilidade

24. O que você acha da interface do software/aplicativo que você escolheu como recurso principal?

25. As crianças conseguem ler e entender as instruções e legendas do software/aplicativo?

26. Há algum recurso do software/aplicativo que as crianças têm mais dificuldade em acessar?

27. O tempo de resposta do software/aplicativo causa algum incômodo nas crianças?

28. Há possibilidade de corrigir erros durante o uso do software/aplicativo?

29. Há possibilidade de salvar progressos no aplicativo?

Outras

30. Você gostaria de acrescentar algo sobre o que a gente?

APÊNDICE D

Modelo de Cartão de Insights

CARTÃO DE INSIGHTS

Título	Número
---------------	---------------

Tema:

Fato:

Fonte:
