

UFRRJ
INSTITUTO DE AGRONOMIA
PROGRAMA DE PÓS-GRADUAÇÃO EM
EDUCAÇÃO AGRÍCOLA

DISSERTAÇÃO

PRODUÇÃO DE JOGOS DIGITAIS E PENSAMENTO
COMPUTACIONAL: UMA ABORDAGEM CONSTRUCIONISTA DE
APRENDIZAGEM

DEIVID DO VALE NASCIMENTO

2024



**UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO DE AGRONOMIA
PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO AGRÍCOLA**

**PRODUÇÃO DE JOGOS DIGITAIS E PENSAMENTO
COMPUTACIONAL: UMA ABORDAGEM CONSTRUCIONISTA DE
APRENDIZAGEM**

DEIVID DO VALE NASCIMENTO

Sob a Orientação do Professor
Gabriel de Araújo Santos

Dissertação submetida como requisito parcial para a obtenção do grau de **Mestre em Educação**, no Programa de Pós-Graduação em Educação Agrícola, Área de Concentração em Educação Agrícola

Seropédica /RJ
Fevereiro de 2024

Universidade Federal Rural do Rio de Janeiro
Biblioteca Central / Seção de Processamento Técnico

Ficha catalográfica elaborada
com os dados fornecidos pelo(a) autor(a)

N244p

NASCIMENTO, DEIVID DO VALE , 1993DEIVID DO VALE
NASCIMENTO-
PRODUÇÃO DE JOGOS DIGITAIS E PENSAMENTO
COMPUTACIONAL: UMA ABORDAGEM CONSTRUCIONISTA DE
APRENDIZAGEM / DEIVID DO VALE NASCIMENTO. -
Seropédica, 2024.
114 f.: il.

Orientador: Gabriel de Araújo Santos.
Dissertação (Mestrado). -- Universidade Federal Rural
do Rio de Janeiro, Programa de Pós-Graduação em Educação
Agrícola, 2024.

1. Construcionismo. 2. Pensamento Computacional.
3. Robótica. I. Santos, Gabriel de Araújo , 1949-,
orient. II Universidade Federal Rural do Rio de
Janeiro. Programa de Pós-Graduação em Educação Agrícola
III. Título.

“O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001”. “This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001”



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO AGRÍCOLA



HOMOLOGAÇÃO DE DISSERTAÇÃO DE MESTRADO Nº 35 / 2024 - PPGEA (11.39.49)

Nº do Protocolo: 23083.022713/2024-32

Seropédica-RJ, 09 de maio de 2024.

**UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE AGRONOMIA
PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO AGRÍCOLA**

DEIVID DO VALE NASCIMENTO

Dissertação submetida como requisito parcial para obtenção do grau de Mestre em Educação, no Programa de Pós-Graduação em Educação Agrícola, Área de Concentração em Educação Agrícola.

DISSERTAÇÃO APROVADA EM: 15/03/2024

Dr. GABRIEL DE ARAUJO SANTOS - UFRRJ
Orientador

Ms. FRANCELINA DE QUEIROZ FELIPE DA CRUZ - UFRRJ
Membro interno

Dr. MARCOS BACIS CEDDIA - UFRRJ
Membro interno

Dra. CRISTIANE DE FÁTIMA DOS SANTOS CARDOSO - IFgoiano
Membro externo

(Assinado digitalmente em 09/05/2024 15:21)
FRANCELINA DE QUEIROZ FELIPE DA CRUZ
PROFESSOR MAGISTERIO SUPERIOR-SUBSTITUTO
DeptTPE (12.28.01.00.00.00.24)
Matrícula: 1035885

(Assinado digitalmente em 09/05/2024 10:18)
GABRIEL DE ARAUJO SANTOS
PROFESSOR DO MAGISTERIO SUPERIOR
DATS (11.39.00.35)
Matrícula: 6385676

(Assinado digitalmente em 13/05/2024 10:42)
MARCOS BACIS CEDDIA
PROFESSOR DO MAGISTERIO SUPERIOR
DATS (11.39.00.35)
Matrícula: 1220296

(Assinado digitalmente em 10/05/2024 08:41)
CRISTIANE DE FÁTIMA DOS SANTOS CARDOSO
ASSINANTE EXTERNO
CPF: 920.268.421-91

Visualize o documento original em <https://sipac.ufrj.br/public/documentos/index.jsp> informando seu número: **35**, ano: **2024**, tipo: **HOMOLOGAÇÃO DE DISSERTAÇÃO DE MESTRADO**, data de emissão: **09/05/2024** e o código de verificação: **50319867c1**

DEDICATÓRIA

Dedico este trabalho à minha mãe, Venilma Maria do Nascimento, cujo apoio incansável, inspiração constante e incentivo motivador foram fundamentais para a realização deste marco em minha carreira profissional. Sua presença e orientação foram essenciais em cada etapa deste percurso, possibilitando-me alcançar este importante objetivo.

AGRADECIMENTOS

Expresso minha profunda gratidão ao Prof. Greiton Toledo de Azevedo, cujos insights valiosos foram essenciais no desenvolvimento deste trabalho, além de me acolher e guiar no projeto Mattics. Agradeço também ao Prof. Gabriel pela orientação perspicaz e compreensiva, criando um ambiente favorável à conclusão deste estudo. Meu sincero agradecimento a Franceline Neta Coutinho e Vilson Antonio de Amorim, amigos que me incentivaram e acreditaram em meu potencial durante o mestrado. Aos professores da UFRRJ, agradeço por enriquecerem minha jornada com seus conhecimentos e experiências, contribuindo significativamente para meu crescimento pessoal e profissional. Minha gratidão ao IF Goiano – Campus Avançado Ipameri, especialmente à Prof^a. Juliana, pela permissão para realizar esta pesquisa em suas instalações. Ao Hospital Dia do Idoso em Anápolis, referência no tratamento da doença de Parkinson, estendo meus agradecimentos a todos os profissionais e pacientes que se tornaram parte desta história. Finalmente, agradeço aos colegas docentes e discentes que dedicaram tempo e esforço para tornar esta pesquisa possível.

RESUMO

NASCIMENTO, Deivid do Vale. **Produção De Jogos Digitais E Pensamento Computacional: Uma Abordagem Construcionista De Aprendizagem.** 2024. 114f. Dissertação (Mestrado em Educação Agrícola) - Universidade Federal Rural do Rio de Janeiro, Seropédica, RJ. 2024.

Essa dissertação apresenta uma análise aprofundada da criação de jogos digitais e do design de dispositivos robóticos, realizados por estudantes do ensino médio com o objetivo de promover o alívio dos sintomas da doença de Parkinson. A investigação tem como objetivo discernir o impacto dessas atividades inovadoras no aprimoramento do Pensamento Computacional. A metodologia adotada envolve a observação e análise de três projetos distintos. Cada um incorporando conceitos, práticas e perspectivas de Pensamento Computacional, conforme delineado por Resnick (2012). Esses projetos não só promoveram o engajamento dos alunos com princípios fundamentais da computação, como também permitiram a integração de conhecimentos interdisciplinares, evidenciando o potencial da tecnologia educacional para além das salas de aula. Por meio de uma abordagem qualitativa, os dados coletados revelaram a capacidade dos estudantes de entender e aplicar o Pensamento Computacional em contextos práticos, contribuindo para o desenvolvimento de habilidades cruciais como criatividade, colaboração, e resolução de problemas. Consequentemente, este estudo não apenas confirmou a relevância do Pensamento Computacional na educação contemporânea, mas também destacou sua aplicabilidade na criação de soluções inovadoras para desafios reais na área da saúde.

PALAVRAS-CHAVE: Construcionismo; Pensamento Computacional; Robótica;

ABSTRACT

NASCIMENTO, Deivid do Vale. **Digital Games Production and Computational Thinking: A Constructionist Approach to Learning**. 2024. 114f. Dissertation (Master's in Agricultural Education) - Federal Rural University of Rio de Janeiro, Seropédica, RJ. 2024.

This dissertation presents an in-depth analysis of the creation of digital games and robotic device design, conducted by high school students with the aim of promoting relief from Parkinson's disease symptoms. The research aims to discern the impact of these innovative activities on enhancing Computational Thinking. The methodology adopted involves the observation and analysis of three distinct projects, each incorporating concepts, practices, and perspectives of Computational Thinking as outlined by Resnick (2012). These projects not only fostered students' engagement with fundamental computing principles but also allowed the integration of interdisciplinary knowledge, showcasing the potential of educational technology beyond the classroom. Through a qualitative approach, collected data revealed students' ability to understand and apply Computational Thinking in practical contexts, contributing to the development of crucial skills such as creativity, collaboration, and problem-solving. Consequently, this study not only confirmed the relevance of Computational Thinking in contemporary education but also highlighted its applicability in creating innovative solutions for real challenges in the healthcare field.

KEYWORDS: Constructionism; Computational Thinking; Robotic;

LISTA DE FIGURAS E ILUSTRAÇÕES

Figura 1 - Palavras mais usadas nas definições de pensamento computacional.....	18
Figura 2 - Polígono irregular.....	20
Figura 3 - Sintomas de AVC.....	22
Figura 4 - Coleta Seletiva.....	23
Figura 5 - Localização geográfica do Estado de Goiás com destaque para as unidades do IF Goiano.....	30
Figura 6 - Vista aérea do Campus Avançado Ipameri.....	30
Figura 7 - Estudantes Engajados no Laboratório de Matemática: Uma Cena de Colaboração Ativa onde Estudantes Analisam, Identificam e Corrigem Erros, Além de Desenvolverem Novas Funcionalidades para os Jogos Propostos.....	37
Figura 8 - Estudantes explicando seções do código do jogo Labirinto.....	37
Figura 9 - União e Tecnologia: Estudantes Utilizando a Placa Makey-Makey em Conjunto para Jogar Labirinto, Uma Experiência de Aprendizado Colaborativo Onde a Conexão Física Entre os Alunos Completa o Circuito do Joystick.....	38
Figura 10 - Estudantes explicando seções do código do jogo Pássaro.....	38
Figura 11 - Estudante explicando a função exponencial do jogo Paraquedas.....	39
Figura 12 - Divisão dos grupos responsáveis pelos jogos.....	39
Figura 13 - Sessão de Videoconferência Interativa: Alunos em Diálogo com a Fisioterapeuta Bruna Cândido.....	40
Figura 14 - Estudantes apresentando o dispositivo robótico “Asa”.....	41
Figura 15 - Estudantes apresentando os dispositivos robóticos “Leme” e “Regador”.....	41
Figura 16 - Estudantes refinando e testando os jogos e dispositivos robóticos.....	42
Figura 17 - Seção de fisioterapia no Hospital Dia do Idoso.....	43
Ilustração 1: Iterações Multifásicas: Uma Análise Visual do Desenvolvimento do 'Bird in The Sky' (dados da pesquisa).....	49
Figura 18 - Espectrograma de uma das músicas da biblioteca Scratch utilizada como tema do jogo Bird In The Sky (dados da pesquisa).....	53
Figura 19 - Trecho de código do jogo Bird In The Sky (dados da pesquisa).....	54
Figura 20 - Dispositivo robótico adaptado (dados da pesquisa).....	55
Figura 21 - Uso prático do dispositivo robótico Asa (dados da pesquisa).....	56
Ilustração 3: Iterações Multifásicas: Uma Análise Visual do Desenvolvimento do Flower (dados da pesquisa).....	61
Ilustração 4 - Materiais de baixo custo usados na confecção do dispositivo robótico Regador... 67	
Figura 22- Apresentação e Brainstorming sobre a concepção do Game Flower (dados da pesquisa).....	68
Ilustração 5: Iterações Multifásicas: Uma Análise Visual do Desenvolvimento do Way To The Garden (dados da pesquisa).....	77
Ilustração 6 - Materiais utilizados na construção do dispositivo robótico volante (dados da pesquisa).....	83

LISTA DE QUADROS

Quadro 1 - Recorte dos trabalhos deferidos na revisão de literatura (Dados da Pesquisa).....	7
Quadro 2 - Softwares utilizados nos trabalhos deferidos na revisão de literatura da dissertação (Dados da Pesquisa).....	9
Quadro 3 - Definição do PC segundo diversos autores.....	16
Quadro 4 - Dimensões do PC segundo diversos autores.....	17
Quadro 5 - Habilidades do PC segundo diversos autores.....	19
Quadro 6 - Ferramentas utilizadas no projeto Mattics.....	34
Quadro 7 - Analisando o jogo Bird in the Sky (Dados da pesquisa) ‘continua’.....	50
Quadro 8 - Conceitos do PC identificados no trecho de código de Bird In The Sky.....	52
Quadro 9 - Trechos de código do jogo Bird In The Sky e suas funcionalidades (dados da pesquisa) ‘continua’.....	57
Quadro 10 - Habilidades do pensamento computacional percebidas nos trechos de código....	59
Quadro 11 - Analisando o jogo Flower (dados da pesquisa) ‘continua’.....	62
Quadro 12 - Conceitos do PC identificados no trecho de código do jogo Flower.....	64
Quadro 13 - Leiaute e fragmento de código do jogo (ator6) Flower responsável pela tela de finalização.....	65
Quadro 14 - Trajetória sinusoidal do personagem Ator1 ‘continua’.....	69
Quadro 15 - Elementos da abstrações identificados no trecho de código Ator1.....	71
Quadro 16 - Elementos da decomposição identificados no trecho de código Ator1.....	72
Quadro 17 - Elementos de reconhecimento de padrões identificados no código Ator1.....	74
Quadro 18 - Elementos de algoritmo identificados no trecho de código Ator1.....	75
Quadro 19 - Analisando o jogo Way To The Garden (dados da pesquisa) ‘continua’.....	79
Quadro 20 - Conceitos do PC utilizados no trecho de código Carro 1.....	81
Quadro 21 - Personagem Tronco de Madeira do jogo Way To The Garden (dados da pesquisa).....	85
Quadro 22 - Elementos da abstração identificados no trecho de código do jogo Way To The Garden.....	86
Quadro 23 - Elementos da decomposição identificados no trecho de código do jogo Way To The Garden ‘continua’.....	87
Quadro 24 - Elementos do reconhecimento de padrões identificados no trecho de código do jogo Way To The Garden.....	89
Quadro 25 - Elementos de algoritmos identificados no trecho de código do jogo Way To The Garden ‘continua’.....	90

LISTA DE FLUXOGRAMAS, GRÁFICOS E INFOGRÁFICOS

Fluxograma 1 - Percurso de revisão bibliográfica [organização, fluxo e deferimento] (Dados da Pesquisa).....	6
Gráfico 1 – Número de publicações por ano (Dados da pesquisa).....	8
Infográfico 1 - Visão Geral dos Episódios Analíticos.....	45
Infográfico 2 - Categorias de Análise.....	46

LISTA DE SIGLAS

BNCC	Base Nacional Comum Curricular
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CEP	Comitê de Ética em Pesquisa
PC	Pensamento Computacional
TALE	Termo de Assentimento Livre e Esclarecido
TCLE	Termo de Consentimento Livre e Esclarecido

SUMÁRIO

1 INTRODUÇÃO.....	1
1.1 REVISÃO DA LITERATURA.....	5
1.2 JUSTIFICATIVA.....	10
2 REFERENCIAL TEÓRICO.....	12
2.1 MODELOS TRADICIONAIS DE ENSINO E O CONSTRUCIONISMO DE PAPERT.....	12
2.2 A ABORDAGEM DO PENSAMENTO COMPUTACIONAL E SUAS DIMENSÕES.....	15
2.3 HABILIDADES DO PENSAMENTO COMPUTACIONAL.....	17
2.4 DIMENSÕES DO PENSAMENTO COMPUTACIONAL SEGUNDO MITCHEL RESNICK - CONCEITOS, PRÁTICAS E PERSPECTIVAS.....	24
2.4.1 Conceitos de Pensamento Computacional.....	24
2.4.2 Práticas do Pensamento Computacional.....	26
2.4.3 Perspectivas do Pensamento Computacional.....	27
3 PERCURSO METODOLÓGICO.....	28
3.1 PESQUISA QUALITATIVA.....	28
3.2 CONTEXTO DE PESQUISA.....	29
3.2.1. Perfil do Estudante - Amostra.....	32
3.3 JOGOS DIGITAIS E DISPOSITIVOS ROBÓTICOS: UMA ABORDAGEM CONSTRUCIONISTA.....	33
3.3.1 Scratch.....	35
3.4 A ELABORAÇÃO DE UMA PROPOSTA PEDAGÓGICA CONSTRUCIONISTA NO ENSINO MÉDIO INTEGRADO.....	36
3.5 ORGANIZAÇÃO DOS DADOS DA PESQUISA.....	43
3.5.1 Episódios Analíticos e Categorias de Análise.....	45
4 APRESENTAÇÃO E ANÁLISE QUALITATIVA DOS DADOS.....	48
4.1 E1 - BIRD IN THE SKY.....	48
4.1.1 E1: Etapas De Desenvolvimento.....	49
4.1.2 E1: Dimensões Do Pensamento Computacional.....	50
4.1.3 E1: Habilidades Do Pensamento Computacional.....	57
4.2 E2: FLOWER.....	60
4.2.1 E2: Etapas De Desenvolvimento.....	61
4.2.2 E2: Dimensões Do Pensamento Computacional.....	62
4.2.3 E2: Habilidades Do Pensamento Computacional.....	69
4.3 E3: WAY TO THE GARDEN.....	76
4.3.1 E3: Etapas De Desenvolvimento.....	77
4.3.2 E3: Dimensões Do Pensamento Computacional.....	78
4.3.3 E3: Habilidades Do Pensamento Computacional.....	84
4.4 AS IMPLICAÇÕES DA CONSTRUÇÃO DE JOGOS DIGITAIS E DISPOSITIVOS ROBÓTICOS PARA O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL DE ESTUDANTES DO ENSINO MÉDIO.....	92
5 CONSIDERAÇÕES FINAIS.....	95
6 REFERÊNCIAS BIBLIOGRÁFICAS.....	97
ANEXOS.....	103

1 INTRODUÇÃO

No âmbito educacional, este estudo se insere no contexto do desenvolvimento de competências nos currículos escolares, com um enfoque especial na promoção das habilidades de resolução de problemas, fundamentadas na Ciência da Computação.

Os estudantes do século XXI, inseridos em uma sociedade do conhecimento, demandam um olhar do educador focado na compreensão dos processos de aprendizagem e na promoção deles por meio de uma nova concepção de como eles ocorrem, independentemente de quem é o sujeito e das suas condições circundantes (Moran, 2018).

Contudo, mesmo nos dias de hoje, em meio à era da cibercultura, muitas escolas no Brasil ainda adotam um modelo de ensino fundamental que se baseia predominantemente na memorização e na simples transmissão de conhecimentos, muitas vezes segmentados em disciplinas isoladas (Pretto; Pinto, 2006; Paula; Valente, 2016).

Nesse cenário, ressurgem as conversas relacionadas ao avanço do Pensamento Computacional (PC), que ganhou notoriedade a partir de 2006 através do trabalho de Jeannette Wing. Ela realça o potencial dos princípios da computação como uma metodologia para solucionar problemas, a qual pode ser empregada em variados âmbitos e graus de ensino.

Nos últimos anos, o pensamento computacional emergiu como uma habilidade crucial na educação básica, transformando a maneira como os alunos aprendem e abordam os desafios acadêmicos. De acordo com Wing (2006), o pensamento computacional, originalmente desenvolvido como uma habilidade fundamental para cientistas da computação e engenheiros de *software*, transcendeu suas origens e agora desempenha um papel vital em uma ampla variedade de disciplinas e níveis de ensino.

A introdução do pensamento computacional nas salas de aula da educação básica tem se revelado uma mudança significativa e benéfica no processo de aprendizagem. Segundo Wing (2006), essa abordagem não se limita apenas ao ensino de programação de computadores, mas se estende a uma mentalidade analítica que promove a resolução de problemas, a lógica, a criatividade e a capacidade de decompor tarefas complexas em etapas mais simples.

Nessa perspectiva, instituições brasileiras têm se esforçado para que as escolas adotem os modernos padrões educacionais que já se consolidaram em outras nações. Focando em um contexto mais atual, merecem destaque os pareceres emitidos pelo Conselho Nacional de Educação (CNE): o Parecer CNE/CP nº 15, datado de 21 de dezembro de 2017, e a Resolução CNE/CP nº 2, de 22 de dezembro de 2017. Essas diretrizes estabelecem normas específicas

para a integração da Computação na Base Nacional Comum Curricular (BNCC) no contexto da Educação Básica, abrangendo tanto a Educação Infantil quanto o Ensino Fundamental, conforme delineado no capítulo V, artigo 22.

Da mesma forma, cabe destacar a Resolução do Conselho Nacional de Educação - CNE/CP nº 4, datada de 17 de dezembro de 2018, que representa um complemento crucial à BNCC no contexto do Ensino Médio. O artigo 18 dessa resolução reafirma a importância de diretrizes suplementares relacionadas aos "Conteúdos e processos pertinentes à aprendizagem de Computação na Educação Básica". Como resultado dessa consideração e em resposta à deliberação da Câmara de Educação Básica (CEB), materializada na Indicação CNE/CEB nº 3/2019, foi estabelecida a Portaria CNE/CEB nº 9, em 11 de dezembro de 2019, com o propósito de criar uma comissão encarregada de elaborar normas específicas relativas ao ensino da Computação.

Como resultado dessa série de iniciativas, tem se à elaboração das Normas sobre Computação na Educação Básica, as quais servem como um complemento essencial à BNCC e foram recentemente aprovadas como Resolução Nº 1, em 4 de outubro de 2022 (Brasil e, 2022). Esta proposta representa o produto de um esforço dedicado e de longa duração de educadores de diversas áreas da Computação, em várias unidades federativas do Brasil, que se dedicam ao ensino e pesquisa da Computação na Educação Básica. A concepção dessa proposta se baseou nos três pilares delineados no documento intitulado 'Diretrizes para o Ensino de Computação na Educação Básica' (SBC, 2019), que compreendem os seguintes eixos: Pensamento Computacional, Mundo Digital e Cultura Digital.

Ainda assim, o ensino de computação nas escolas brasileiras é carente, especialmente no que diz respeito à introdução do PC. A ausência de um planejamento eficiente e de uma estrutura adequada emerge como um obstáculo significativo para a efetiva implementação da computação como ciência nas escolas públicas. De acordo com Soares Neto et al. (2013), 40% das instituições escolares são categorizadas como possuindo infraestrutura básica, caracterizada pela presença de elementos essenciais como água, sanitários, energia, esgoto, cozinha, sala de diretoria, além de equipamentos como TV, DVD, computadores e impressora. Contrastando com essa realidade, apenas 15,5% das escolas brasileiras desfrutam de características mais avançadas, indicando uma infraestrutura escolar considerada adequada e sofisticada, um ambiente propício para o desenvolvimento de atividades relacionadas à computação. Essa disparidade ressalta a necessidade urgente de investimentos e planejamento estratégico para proporcionar a todas as escolas condições propícias ao ensino efetivo da ciência da computação.

Para Silva (2016), outro fator que limita a integração do ensino de computação nas escolas é a escassez de professores devidamente capacitados e confiantes no uso efetivo da computação como ferramenta educacional. A falta de preparo docente nesse contexto é uma barreira, visto que um ensino de computação bem aplicado não apenas abre espaço para a interdisciplinaridade, mas também permite uma contextualização na resolução de problemas, proporcionando relevância prática para o cotidiano do aluno, tanto em atividades escolares quanto em iniciativas extracurriculares. A superação desse desafio exige investimentos significativos em formação docente, incentivando o desenvolvimento de habilidades que habilitem os professores a incorporar efetivamente a computação no processo educacional (Silva, 2016).

Em consonância com as diretrizes estabelecidas para a Educação Básica, particularmente no que diz respeito ao ensino de Computação, o estudo conduzido por Azevedo (2022) se concentra na formação matemática, aplicação do Pensamento Computacional (PC) utilizando estratégias construcionistas no contexto do desenvolvimento de jogos digitais e dispositivos robóticos sustentáveis e de baixo custo para o tratamento dos sintomas da doença de Parkinson. Ainda de acordo com Azevedo (2022), o projeto de extensão denominado Mattics¹, utilizou ferramentas computacionais de programação e robótica como recursos pedagógicos para orientar os estudantes na concepção e desenvolvimento de jogos digitais, simultaneamente promovendo o pensamento matemático e o pensamento computacional. À medida que os alunos colaboram em projetos, eles "não apenas constroem redes de conceitos, mas também adquirem habilidades - unindo ideias e criando soluções para problemas e para comunicar ideias" (Resnick, 2017, p. 54, tradução nossa).

Considerando as demandas educacionais do século XXI e a necessidade de alinhar os métodos de ensino com a realidade contemporânea, Papert (2008) sugere uma estratégia construcionista que envolve a utilização de ferramentas computacionais para promover a criação colaborativa de jogos digitais. De acordo com suas premissas, a produção de jogos digitais com objetivos educacionais bem definidos visa proporcionar diversas vantagens, incluindo a promoção da descoberta, do pensamento crítico, da curiosidade e da autonomia dos alunos. Além disso, busca fomentar uma maior integração e colaboração entre professores e estudantes, bem como entre os próprios estudantes, no processo de construção de

¹ "Projeto Mattics - Facebook." https://www.facebook.com/matticsoficial/?locale=pt_BR. Acessado em 15 jan.. 2024.

conhecimento. Essa abordagem também visa criar um ambiente de aprendizado participativo e dinâmico.

É fundamental ressaltar que essa dinâmica de ensino não apenas catalisa o desenvolvimento das habilidades matemáticas e do pensamento computacional dos estudantes, mas também oferece uma perspectiva de aprendizado inovadora e envolvente, proporcionando uma visão mais ampla e interconectada das disciplinas estudadas.

Para Azevedo (2017) algumas características em favor do fortalecimento ao contexto de formação em matemática são desenvolvidas ao se trabalhar com invenções e soluções científico-tecnológicas.

[...] à medida que os alunos desenvolvem invenções e soluções científico-tecnológicas, algumas características se remontam em perspectivas potenciais de fortalecimento ao contexto de formação em Matemática, tais como: Independência formativa, imprevisibilidade de respostas cabais, aprendizagem centrada à compreensão-investigação, conexão entre distintas áreas (interdisciplinaridade), responsabilidade e colaboração mútua [...]. (Azevedo, 2018, p. 163)

Wing (2006), afirma que essas e outras características também podem ser percebidas em favor da formação do pensamento computacional, em específico ao se utilizar linguagens de programação. Diante disso, a nossa pergunta diretriz que norteia a investigação desta obra é: Quais as implicações da construção de jogos digitais e dispositivos robóticos para o desenvolvimento do pensamento computacional de estudantes do ensino médio? Procuramos responder a esse questionamento realizando a participação e análise das atividades desenvolvidas por uma turma de estudantes do projeto Mattics do IF Goiano - Câmpus Avançado Ipameri no ano de 2023 com objetivo geral de examinar o processo de desenvolvimento do Pensamento Computacional de estudantes do ensino médio, em um ambiente construcionista, quando constroem jogos digitais e desenvolvem dispositivos robóticos sustentáveis e de baixo custo. Para atingir o objetivo geral, definimos os seguintes objetivos específicos:

1. Identificar as dimensões (estrutura e práticas) do Pensamento Computacional, conforme definido por Resnick em 2012, utilizadas no projeto de extensão Mattics;
2. Analisar a construção dos algoritmos computacionais dos jogos digitais de estudantes de ensino médio;
3. Evidenciar as habilidades do Pensamento Computacional desenvolvidas pelos estudantes durante o processo;

Para contextualizar esta pesquisa e destacar sua relevância na área do Pensamento Computacional, apresentamos a seção de revisão a seguir.

1.1 REVISÃO DA LITERATURA

A revisão de literatura realizada para esta dissertação teve como objetivo identificar trabalhos com impacto social relacionados ao desenvolvimento do Pensamento Computacional de estudantes do Ensino Médio, especificamente no contexto de criação de inovações científico-tecnológicas de baixo custo e sustentáveis, incluindo jogos digitais e dispositivos robóticos.

Para conduzir esta pesquisa de literatura, optamos por seguir o método *Preferred Reporting Items for Systematic Reviews and Meta-Analyses* (PRISMA)². Esse método visa identificar e analisar pesquisas potencialmente relevantes que ofereçam suporte a decisões, ao mesmo tempo que evidenciam lacunas no objeto de estudo, no presente contexto, o Desenvolvimento do Pensamento Computacional. Para alcançar esse objetivo, delineamos as seguintes etapas de mapeamento e tabulação: (i) período considerado; (ii) línguas abordadas; (iii) bases de dados pesquisadas; (iv) sentenças lógicas utilizadas; (v) critérios de inclusão; (vi) critérios de exclusão; (vii) trabalhos selecionados; (viii) análises ou adaptações realizadas com base nos resultados obtidos. Esse rigoroso protocolo metodológico visa assegurar uma abordagem sistemática e transparente na identificação e seleção das fontes, contribuindo para uma revisão abrangente e fundamentada sobre o tema em questão.

Nesta revisão pesquisamos este tema em outros idiomas (Inglês e Espanhol) e utilizamos operadores booleanos para a busca em diferentes bases de dados sendo eles: “Pensamento Computacional AND Jogos Digitais”; “Pensamento Computacional AND Jogos Digitais AND Ensino Médio”; “Pensamento Computacional AND Robótica”; “Pensamento Computacional AND Robótica AND Ensino Médio”; “Pensamento Computacional AND Impacto Social”.

As plataformas de mapeamento utilizadas incluíram o Catálogo de Dissertações e Teses da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e Dissertações e Teses da ERIC - Education Resources Information Center. Além disso, foram considerados periódicos indexados à área do Pensamento Computacional listados pela Biblioteca Digital da Sociedade Brasileira de Computação (SBC OPENLIB) tais como o periódico "Computação na Escola" e a "Revista Brasileira de Informática na Educação".

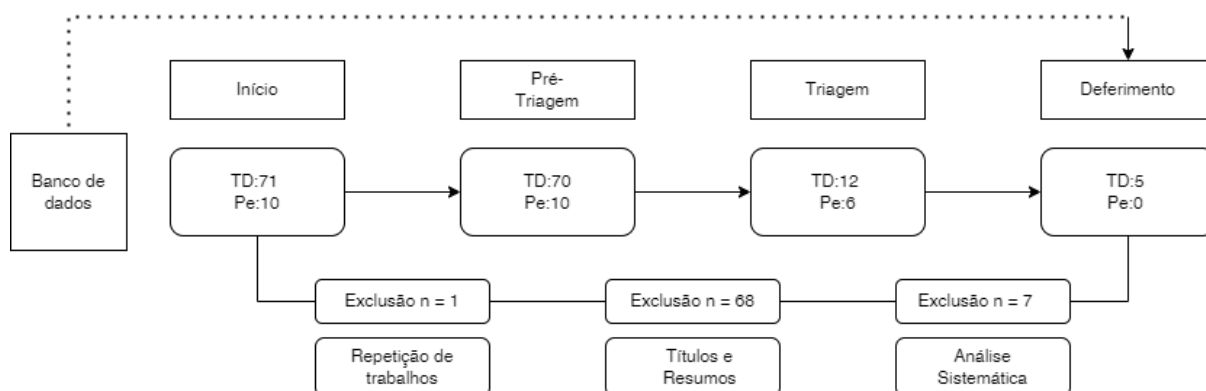
Para determinar a elegibilidade dos trabalhos, foram estabelecidos critérios específicos, como a concentração na área do Pensamento Computacional, a abordagem de

² Cf.: Disponível em: < <https://www.prisma-statement.org/> >. Acesso em outubro de 2023

invenções robóticas, audiência-alvo restrita aos estudantes do Ensino Médio e impacto social. A seleção dos trabalhos passou por quatro etapas principais: Início, Pré-Triagem, Triagem e Deferimento.

Durante a fase de Pré-Triagem, foram analisados os títulos e resumos dos trabalhos relacionados para determinar sua relevância em relação ao tema da pesquisa. Na etapa de Triagem, os trabalhos selecionados na pré-triagem passaram por uma leitura sistemática mais detalhada. Apenas aqueles que discutiam a produção ou invenção de jogos digitais com linguagem de programação e a invenção robótica no contexto do Pensamento Computacional na educação básica foram considerados. Foi feita uma exclusão cuidadosa de pesquisas que não se enquadraram nos critérios estabelecidos, incluindo aquelas que se concentravam na formação de professores, bem como resumos, ensaios, pesquisas incompletas, resenhas de livros, relatos de experiências e repetições de trabalhos já identificados nas bases de dados.

O processo de seleção pode ser visualizado no Fluxograma 1.



Fluxograma 1 - Percurso de revisão bibliográfica [organização, fluxo e deferimento] (Dados da Pesquisa).

Fonte: elaborado pelo autor

A partir da análise dos 70 trabalhos (teses e dissertações) obtidos durante a Pré-Triagem, realizamos um refinamento por meio da leitura dos títulos, resumos e verificação dos trabalhos repetidos. Como resultado desse processo, catalogamos 12 obras potenciais para estudo. Após uma análise sistemática seguindo critérios de elegibilidade, identificamos cinco trabalhos relevantes: duas teses e três dissertações, todos abordando o desenvolvimento do Pensamento Computacional de estudantes do Ensino Médio com foco em robótica ou programação de jogos digitais. Destacamos que os 6 periódicos inicialmente considerados foram excluídos por não atenderem aos critérios de elegibilidade estabelecidos. Importante observar que dentre os trabalhos selecionados, encontramos apenas um trabalho relacionado a inovações voltadas para resolver problemas reais na sociedade.

Dos 76 trabalhos excluídos, grande parte abordou aspectos da computação desplugada³, enquanto outra parte tratava de estudantes do Ensino Fundamental. Além disso, alguns dos trabalhos focalizaram outras áreas de conhecimento, como Engenharia.

Os trabalhos selecionados para análise estão disponíveis para visualização no Quadro 1 a seguir.

Quadro 1 - Recorte dos trabalhos deferidos na revisão de literatura (Dados da Pesquisa).

Títulos	Tipo	Autor (Ano)	Abordagem
Processo formativo em Matemática: invenções robóticas para o Parkinson	Tese	Azevedo (2022)	Qualitativa
Relações entre o Pensamento Computacional e a Matemática em atividades didáticas de construção de jogos digitais	Tese	Barcelos (2014)	Qualitativa
Pensamento Computacional: o uso do Scratch no ensino de Ciências	Dissertação	Campos (2021)	Qualitativa
Do Desplugado Ao Plugado: Uma Proposta Para O Desenvolvimento Do Pensamento Computacional E Do Pensamento Matemático Avançado Em Aulas Do Ensino Médio	Dissertação	Lopes (2022)	Qualitativa
Injecting Computational Thinking into Computing Activities for Middle School Girls	Dissertação	Webb (2013)	Qualitativa

Fonte: elaborado pelo autor

No que tange à data de publicação, as teses e dissertações identificadas foram publicadas a partir de 2013. Conforme evidenciado no Gráfico 1, os anos de 2013, 2014, 2021 e 2022 se destacam pelo maior volume de artigos, enquanto os anos de 2015 a 2020 se caracterizam pela ausência de trabalhos publicados sobre o tema que estejam em conformidade com os critérios de investigação estabelecidos.

³ Computação Desplugada é uma coleção de atividades livres e gratuitas que trazem conceitos e problemas do mundo da computação para a Educação Básica sem utilizar nenhum computador ou equipamento eletrônico. <<https://desplugada.ime.unicamp.br/>>. Acessado em 15 jan.. 2024.

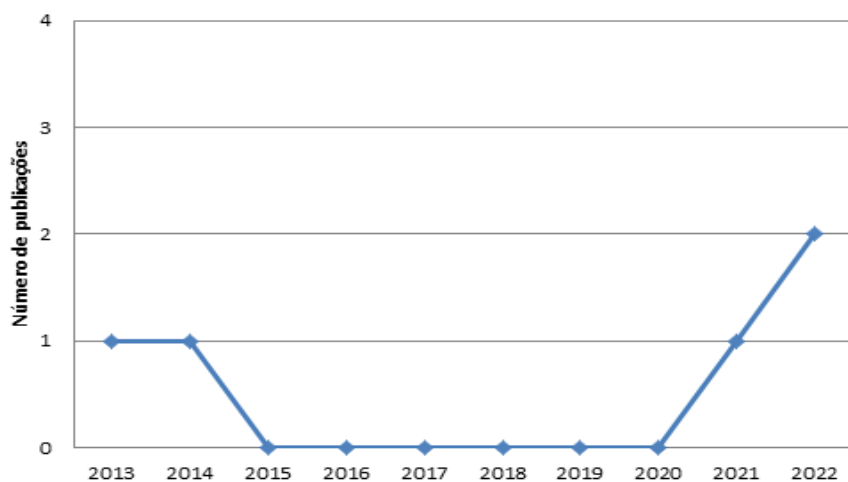


Gráfico 1 – Número de publicações por ano (Dados da pesquisa).
Fonte: elaborado pelo autor.

Entre os trabalhos selecionados, Azevedo (2022) explora o processo formativo em Matemática durante a construção de jogos digitais voltados para o tratamento de Parkinson, enquanto Barcelos (2014) investiga as relações entre Pensamento Computacional e Matemática em atividades de construção de jogos digitais. Campos (2021) destaca a importância do Pensamento Computacional no uso de Tecnologias Digitais de Informação e Comunicação (TDIC) na educação. Lopes (2022) enfatiza a necessidade de incorporar o Pensamento Computacional nas práticas pedagógicas de professores de Matemática, seguindo a Base Nacional Comum Curricular. Finalmente, a dissertação de Webb (2013), aborda especificamente a integração de habilidades de computação e Pensamento Computacional em atividades de aprendizagem de informática voltadas para meninas do ensino médio.

Os estudos citados empregam uma variedade de *softwares* e recursos de robótica em suas investigações. Azevedo (2022) adotou placas BBC micro:bit⁴ e MakeyMakey⁵, permitindo que os alunos desenvolvessem seus próprios *joysticks*. Além disso, ele empregou *softwares* como GeoGebra⁶ para elucidar conceitos matemáticos, e o programa Scratch⁷ para a elaboração de jogos. Tanto Barcelos (2014) quanto Campos (2021) optaram pelo uso do Scratch em suas pesquisas. Lopes (2022) propôs a aplicação dos *softwares* Maze⁸ e Scratch

⁴ "Microsoft MakeCode for micro:bit." <<https://makecode.microbit.org/>>. Acessado em 15 jan.. 2024.

⁵ "Makey Makey – Joylabz Official Makey Makey Store." <<https://makeymakey.com/>>. Acessado em 15 jan.. 2024.

⁶ "Aplicativos matemáticos gratuitos - Usado por mais de ... - GeoGebra." <<https://www.geogebra.org/?lang=pt>>. Acessado em 15 jan.. 2024.

⁷ "Scratch - Imagine, Program, Share." <<https://scratch.mit.edu/>>. Acessado em 15 jan.. 2024.

⁸ "Maze Game | Software - TechTudo." <<https://www.techtudo.com.br/tudo-sobre/maze-game/>>. Acessado em 15 jan.. 2024.

em sua abordagem. Por sua vez, Webb (2013), utilizou Scratch e Alice⁹ em suas atividades. O Quadro 2 exemplifica as informações.

Quadro 2 - *Softwares* utilizados nos trabalhos deferidos na revisão de literatura da dissertação (Dados da Pesquisa).

Autor/Ano	Softwares Utilizados
Azevedo (2022)	BBC micro:bit, MakeyMakey, GeoGebra, Scratch
Lopes (2022)	Maze, Scratch
Campos (2021)	Scratch
Barcelos (2014)	Scratch
Webb, Heidi Cornelia (2013)	Scratch, Alice

Fonte: elaborado pelo autor.

Em síntese, a revisão bibliográfica revela que, embora os trabalhos analisados tenham proporcionado avanços significativos na formação do Pensamento Computacional de estudantes do Ensino Médio por meio do desenvolvimento de eletrônicos educacionais, há uma notável lacuna quando se trata da integração simultânea da construção de jogos e dispositivos robóticos sustentáveis e de baixo custo. Quatro dos cinco estudos resultantes da busca enfocaram principalmente o desenvolvimento do Pensamento Computacional em contextos específicos, como a criação de jogos digitais e dispositivos robóticos, mas não abordaram de maneira conjunta essas duas vertentes.

É importante destacar que, apenas o trabalho de Azevedo (2022) apresenta uma abordagem de invenções científico-tecnológicas destinadas ao impacto social. Essa lacuna revela uma oportunidade significativa para futuras pesquisas e práticas educacionais, especialmente considerando a crescente importância de desenvolver habilidades computacionais em estudantes do Ensino Médio.

Ao considerar o cenário atual da educação, reforçamos a importância de identificar novos modos de formação do Pensamento Computacional que incorporem invenções científico-tecnológicas sustentáveis e de baixo custo. A necessidade de preencher essa lacuna torna-se ainda mais relevante quando voltamos nossa atenção para o potencial impacto social dessas inovações. Ao buscar soluções educacionais inovadoras, sustentáveis e socialmente relevantes, podemos não apenas fortalecer as habilidades dos estudantes, mas também contribuir para melhorias tangíveis nas comunidades.

⁹ "Alice para Windows - Baixe gratuitamente na Uptodown." <<https://alice.br.uptodown.com/windows>>. Acessado em 15 jan.. 2024.

Portanto, diante desse panorama, encorajamos futuras investigações a explorar e desenvolver estratégias pedagógicas que integrem de maneira abrangente a construção de jogos, dispositivos robóticos e invenções científico-tecnológicas. Essa abordagem holística pode não apenas enriquecer a formação do Pensamento Computacional, mas também capacitar os estudantes a aplicar essas habilidades na resolução de problemas do mundo real, promovendo impactos sociais positivos e duradouros.

Na próxima seção, enfatizaremos a importância deste trabalho, fundamentando-o na realidade tecnológica contemporânea que permeia a experiência educacional dos estudantes. Essa abordagem não apenas justifica a relevância do estudo, mas também o contextualiza dentro do cenário digital em que os alunos estão imersos.

1.2 JUSTIFICATIVA

A BNCC apresenta dez competências gerais¹⁰, que deverão ser desenvolvidas por estudantes da educação infantil e básica, notamos que dentre elas existe a competência “Cultura digital” que propõe a utilização e criação de tecnologias digitais de forma crítica significativa e ética para comunicar-se, acessar e produzir informações e conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva. Ela também define diversas competências e habilidades que permitem aos educandos a apropriação do conhecimento e a autonomia, de modo que eles consigam, por exemplo:

Utilizar, propor e/ou implementar soluções (processos e produtos) envolvendo diferentes tecnologias, para identificar, analisar, modelar e solucionar problemas complexos em diversas áreas da vida cotidiana, explorando de forma efetiva o raciocínio lógico, o pensamento computacional, o espírito de investigação e a criatividade. (BNCC, 2018, p.475)

De acordo com WING (2006), Universidades e escolas ao redor do mundo têm vindo a reconhecer a importância do Pensamento Computacional como uma competência essencial para preparar os alunos para os desafios da era digital. Como resultado, muitas instituições estão incorporando a disciplina de Pensamento Computacional em suas grades curriculares. Na escola, o pensamento computacional faz com que o estudante, apresente um maior desenvolvimento, quando desde cedo, começa a desenvolver esse pensamento, aprendendo a

¹⁰ "EDUCAÇÃO É A BASE - BNCC."

<http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf>. p.9-10. Acessado em 17 jan.. 2024.

construir uma forma de pensar logicamente, fazendo com que a facilidade de resolver outros desafios de outras disciplinas acabe sendo maior.

A realidade tecnológica em que vivemos faz com que seja primordial que a criança ou o adolescente tenha uma base sólida sobre habilidades que venham a ajudar no seu desenvolvimento em momentos de tomadas de decisões. Pois assim, a criança estará preparada para resolver problemas, sejam eles relacionados a sua vida pessoal ou profissional, sabendo qual a ferramenta correta a se utilizar.

Esta pesquisa justifica-se devido à crescente importância do pensamento computacional na sociedade contemporânea. O estudo desempenha um papel crucial no aprimoramento dos métodos de ensino, em consonância com as atuais diretrizes curriculares, além de fortalecer o projeto de extensão Mattics no intuito de beneficiar a comunidade, a pesquisa amplia e contribui para o avanço do conhecimento científico na área, destacando-se como um esforço significativo para preencher lacunas educacionais existentes. No contexto atual, a ausência de uma abordagem adequada ao ensino de Computação, especialmente na introdução do Pensamento Computacional nas escolas, ressalta a urgência de iniciativas educacionais inovadoras. Este estudo visa suprir essa carência ao investigar estratégias eficazes para incorporar o Pensamento Computacional de maneira efetiva no ambiente escolar. Ao fazê-lo, não apenas atende à demanda crescente por habilidades computacionais, mas também promove uma educação alinhada com as exigências contemporâneas, preparando os estudantes para os desafios e oportunidades do mundo digital em constante transformação.

Como um incentivo a mais que ratifica a importância desta pesquisa, observamos que o projeto Mattics, analisado neste estudo, é referência no Campus Avançado Ipameri, na abordagem do estudante do IF como construtor da tecnologia e não apenas consumidor, o que pode incentivar outros docentes e discentes a explorarem mais o tema, assim como os demais ligados ao Pensamento Computacional.

No próximo capítulo, exploraremos diferentes perspectivas educacionais, incluindo o método tradicional de ensino, a abordagem construcionista, e ofereceremos um resumo conciso sobre a definição do pensamento computacional, suas dimensões e os princípios fundamentais que o embasam.

2 REFERENCIAL TEÓRICO

2.1 MODELOS TRADICIONAIS DE ENSINO E O CONSTRUCIONISMO DE PAPERT

Autores conceituados, sociólogos e pensadores modernos da educação como Paulo Freire, Seymour Papert, Michel Serres consideram o modo tradicional de educar como sendo ineficiente. Paulo Freire, educador e filósofo brasileiro, possui sua própria concepção sobre os métodos tradicionais de educação, Freire (1987) denomina como sendo educação “bancária” o modo como os professores, por meio da narração, “depositam” os saberes nos alunos como se estes fossem meros recipientes vazios a serem enchidos, alienando-os a medida em que o conhecimento é guardado e arquivado.

Educador e educandos se arquivam na medida em que, nesta distorcida visão da educação, não há criatividade, não há transformação, não há saber. Só existe saber na invenção, na reinvenção, na busca inquieta, impaciente, permanente, que os homens fazem no mundo, com o mundo e com os outros. Busca esperançosa também. (Freire. 1987, p.33).

Paulo Freire preconizava ações e reflexões acerca de questões como o diálogo, a emancipação, a cultura e a liberdade dentro de uma proposta de educação libertadora. Na proposta de Educação Libertadora, Freire (1996) assume que “ensinar não é transferir conhecimento, mas criar as possibilidades para a sua própria produção ou a sua construção” (FREIRE, 1996, p.25), uma conduta do professor que preza a interação e o diálogo entre aluno e professor e por meio da qual ambos constroem conhecimentos.

Conforme destacado por Freire (2014),

[...] se nosso objetivo é alcançar a verdadeira emancipação das pessoas, não devemos iniciar por aliená-las ou mantê-las em estado de alienação. A autêntica libertação, que envolve o processo de humanização, não é algo a ser simplesmente transmitido aos indivíduos. Ela não é apenas um acréscimo vazio e mitificante de palavras. Trata-se de prática (práxis), que envolve a ação e reflexão dos homens sobre o mundo para transformá-lo (Freire, 2014, p. 52).

Dessa forma, na visão Freiriana, retoma-se a compreensão de que a aprendizagem se refere a um procedimento construtivista em que o educando, a partir da sua autonomia de fazer, atue de forma efetiva na construção de novos saberes, contando com o apoio do docente, como facilitador, enquanto interage com o acadêmico, o que inclui o respeito pelas suas opiniões e o aproveitamento dos seus saberes internalizados.

Indo de encontro com as idéias de Paulo Freire de uma educação libertadora, Papert (1988) discorre sobre o propagação de uma cultura de computadores, em uma escola distinta

da atual, como maneira de quebrar círculos viciosos da educação formal. “Não se pode pensar seriamente sobre o pensamento sem pensar sobre pensar em alguma coisa .. ” (PAPERT, 1988, p. 24). Papert (1988) julga as formas como computadores vêm sendo usados na educação, como apenas máquinas para fornecer informações (instrução assistida por computador ou CAI); mesmo respeitando-se características individuais e ritmos e provendo atividades dentro de um nível apropriado de complexidade," ... é o computador programando a criança.. "(PAPERT, 1988, p. 35). Sette (1999) afirma que o *software* é apenas uma ferramenta a serviço do professor, dependendo exclusivamente da sua postura crítico-reflexiva para ser inserido com sucesso nas práticas pedagógicas.

Papert (1988) cunhou o termo "Instrucionismo" para descrever a abordagem de ensino em que o computador é utilizado simplesmente para transmitir informações. Esse termo foi criado em contraposição ao "Construcionismo", uma teoria educacional que ele próprio desenvolveu e que se aproxima de outra teoria denominada Construtivismo, conforme definido por Piaget.

Enquanto o construtivismo delimita a construção de estruturas de conhecimentos por intermédio da internalização progressiva de ações, o construcionismo acrescenta que isso ocorre de maneira mais eficaz quando o aprendiz está em um contexto consciente e quando pode contribuir suas ideias e representá-las no mundo real (Campos, 2019, p.80).

De acordo com Papert (1980) a introdução do conceito de Instrucionismo serviu como meio de destacar e aprofundar as ideias fundamentais do Construcionismo.

Na sua obra *A Família em Rede*, Papert explica a diferença entre essas abordagens, a partir do exemplo dos fabricantes de programas de computadores. Na abordagem instrucionista, o fabricante busca criar um jogo que ensina. É característico da aprendizagem escolar tradicional. O professor ensina o passo a passo “Grande parte dos programas virados para a transmissão de conhecimentos parecem jogos que empurram a criança para um papel reactivo. A máquina coloca uma pergunta, a criança responde” (Papert, 1997, p. 78).

Já na abordagem construcionista, o fabricante recomenda: faça você mesmo o jogo. Os estudantes aprendem os pormenores técnicos do jogo, ao programarem o computador. O mais essencial dentro dessa forma de visão, é que os estudantes vão expandir uma percepção do seu eu e um senso de controle.

Dessa maneira, a principal característica dos ambientes de aprendizagem que se baseiam no construcionismo é a possibilidade de desenvolvimento tanto emocional quanto cognitivo dos estudantes por meio do uso da programação, com a construção de significados

mediante a interação do educando com o computador (Papert, 1988), sendo sugerida como a forma ideal para uso do computador no processo educativo (Valente, 1997).

No construcionismo de Papert (1986) o professor tem o papel de tutor e desafiador. “Cabe ao professor, momento histórico em que se vive, o trabalho de apresentar, aos educandos, conhecimentos ou informações sob a forma de problemas a resolver, situando-os num contexto e colocando-os em perspectiva, de modo que possam estabelecer ligação entre a sua solução e outras mais abrangentes” (Delors, 1998 p. 34).

Papert (1996) conduziu uma série de estudos que culminaram na definição de cinco dimensões fundamentais para a criação de ambientes de aprendizagem baseados no Construcionismo. Cada uma dessas dimensões desempenha um papel crucial na promoção da aprendizagem significativa. Abaixo, apresentamos uma descrição de cada uma das dimensões propostas:

- **Dimensão Pragmática:** Nessa dimensão, destaca-se a importância de utilizar o conteúdo de maneira prática, de modo que o aprendiz crie artefatos com aplicação imediata. Isso proporciona ao aprendiz a sensação de que está desenvolvendo algo com utilidade real em um curto espaço de tempo.
- **Dimensão Sintônica:** A dimensão sintônica promove a conexão entre o aprendiz e o conteúdo, permitindo que o aprendiz escolha tópicos que sejam relevantes e significativos para ele. Essa abordagem aumenta a relevância do projeto, torna a aprendizagem mais envolvente e melhora a assimilação do conteúdo.
- **Dimensão Sintática:** Nesta dimensão, destaca-se a acessibilidade do ambiente educativo, permitindo que o aprendiz avance em seus estudos sem a necessidade de pré-requisitos complexos. A ênfase está na facilidade de navegação e no desenvolvimento cognitivo do aprendiz.
- **Dimensão Semântica:** Essa dimensão enfatiza a interação do aprendiz com elementos que tenham significado pessoal, não se limitando a abordagens formais de aprendizagem. Isso permite que o aprendiz se relacione de maneira mais profunda com o conteúdo.
- **Dimensão Social:** A dimensão social integra a atividade de aprendizagem ao contexto cotidiano do aprendiz, garantindo que o conteúdo esteja alinhado com a cultura do ambiente em que a aprendizagem ocorre. Isso promove uma conexão mais relevante e significativa com o conteúdo.

Ao desenvolver atividades nas quais as dimensões propostas por Papert sejam estimuladas, são aumentadas consideravelmente as chances de que o conhecimento seja construído corretamente pelo aprendiz, desde que haja boa elaboração de atividades, para que possam abranger todas as dimensões propostas (Maltempi, 2000).

2.2 A ABORDAGEM DO PENSAMENTO COMPUTACIONAL E SUAS DIMENSÕES

O conceito de pensamento computacional remonta a um período não recente; desde 1972, Papert já sublinhava os benefícios da incorporação do computador na educação, como evidenciado pelo sistema Logo¹¹. Ainda que o termo tenha sido introduzido por Seymour Papert em 1980, foi com o artigo de Wing (2006) que ele adquiriu maior destaque. Nesse artigo, Wing definiu o pensamento computacional como uma competência fundamental, relevante não apenas para cientistas da computação, mas para todos os indivíduos.

Jeannette Wing apresenta diversas definições de Pensamento Computacional durante o período de 2006 a 2014. De acordo com a autora, o Pensamento Computacional abrange os processos mentais envolvidos na elaboração de um problema e na apresentação de soluções eficazes, de maneira que tanto uma máquina quanto um ser humano possam compreender e executar (Wing, 2016).

Além disso, Wing ressalta que a essência do Pensamento Computacional no contexto educacional não reside apenas na focalização do algoritmo em si, nem na codificação ou nos procedimentos relacionados, como sintaxe, algoritmos ou compilação. Ao contrário, o enfoque recai sobre a habilidade de desenvolver a lógica sequencial, estimular a curiosidade, elaborar estratégias, identificar e concatenar ideias para resolver os desafios emergentes na sociedade (Denning, 2017; Wing, 2006; 2016). Portanto, o pensamento computacional não se limita a enviar e-mails, dominar a navegação na internet, criar blogs, operar processadores de texto ou planilhas eletrônicas. Não se resume a conceitos computacionais para resolver problemas em sua forma mais básica; engloba igualmente práticas de conceber sistemas, compreender o comportamento humano e estimular o pensamento crítico (Wing, 2006).

¹¹ "Logo – Wikipédia, a enciclopédia livre." <https://pt.wikipedia.org/wiki/Logo>. Acessado em 15 jan.. 2024.

Diversos autores propuseram várias outras definições de Pensamento Computacional. Entre esses, destacamos no Quadro 3:

Quadro 3 - Definição do PC segundo diversos autores.

Autor/Ano	Definição
Brackmann (2017)	O Pensamento Computacional é uma distinta capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da Computação, nas mais diversas áreas do conhecimento, com a finalidade de identificar e resolver problemas, de maneira individual ou colaborativa, através de passos claros, de tal forma que uma pessoa ou uma máquina possam executá-los eficazmente.
Google for Education (2015)	Uma abordagem para solucionar problemas utilizando o conhecimento em Computação.
Liukas (2015)	Pensar nos problemas de maneira que possam ser resolvidos por um computador. Inclui pensamento lógico, reconhecimento de padrões, algoritmos, decomposição e abstração.
Furber (2012)	O processo de reconhecer aspectos da computação no ambiente ao nosso redor e aplicar ferramentas da Ciência da Computação para entender e analisar sistemas naturais e artificiais.
McMaster et al (2010)	Uma prática fundamental da Ciência da Computação, que não é exclusiva da área e não abrange completamente o campo.

Fonte: Brackmann (2007). Adaptado pelo autor.

Diversos autores têm contribuído para a compreensão das estruturas do pensamento computacional. Esses autores têm abordagens ligeiramente diferentes para as dimensões do pensamento computacional, mas em geral, suas contribuições ajudaram a moldar o entendimento desse conceito essencial no campo da educação e da informática. Wing (2006) é conhecida por sua definição amplamente aceita de pensamento computacional. Ela descreveu quatro dimensões: decomposição, reconhecimento de padrões, abstração e algoritmos. Grover e Pea (2013) propuseram uma estrutura que inclui quatro dimensões do pensamento computacional: ações de computação, pensamento algorítmico, pensamento sobre sistemas e práticas de resolução de problemas. Harel (1990) desenvolveu um modelo de pensamento computacional com quatro dimensões: ação computacional, pensamento algorítmico, pensamento sistêmico e pensamento inovador. Guzdial (2008) propõe um modelo que destaca as dimensões de resolução de problemas, expressão criativa e compreensão de sistemas, juntamente com as dimensões conceituais do pensamento computacional.

No Quadro 4 estão as dimensões do pensamento computacional citadas pelos seus autores.

Quadro 4 - Dimensões do PC segundo diversos autores

Autor/Ano	Dimensões do Pensamento Computacional
Brennan e Resnick (2012)	- Conceitos - Práticas - Perspectivas
Shuchi Grover e Roy Pea (2013)	- Ações de Computação - Pensamento Algorítmico - Pensamento sobre Sistemas - Práticas de Resolução de Problemas
Mark Guzdial (2008)	- Resolução de Problemas - Expressão Criativa - Compreensão de Sistemas - Dimensões Conceituais do Pensamento Computacional
Jeannette Wing (2006)	- Decomposição - Reconhecimento de Padrões - Abstração - Algoritmos
Idit Harel (1990)	- Ação Computacional - Pensamento Algorítmico - Pensamento Sistêmico - Pensamento Inovador

Fonte: elaborado pelo autor

Dentre as diversas dimensões do Pensamento Computacional citadas por diferentes autores, para melhor compreendermos essas dimensões, na seção 2.4 focaremos no trabalho realizado por Brennan e Resnick (2012). Nesse estudo, eles se propuseram a definir o PC por meio de três dimensões distintas (conceitos, práticas e perspectivas) e a avaliar seu impacto utilizando três abordagens fundamentais (análise do portfólio de projetos, entrevistas baseadas nos artefatos e cenário de projetos). Sendo possivelmente o primeiro *framework*¹² desenvolvido para o Pensamento Computacional, esse trabalho tem sido amplamente referenciado. Para atender o primeiro objetivo específico desta dissertação, limitaremos nossa análise às dimensões do Pensamento Computacional proposto por Resnick (2012), sem abordar seu método de avaliação.

2.3 HABILIDADES DO PENSAMENTO COMPUTACIONAL

De acordo com diversos autores, incluindo Wing (2006), Riley e Hunt (2014), Krauss e Prottzman (2016), Gadanidis (2017), Marques et al. (2017), Barbosa (2019) e as diretrizes do ISTE/CSTA (2011), a aplicação adequada do Pensamento Computacional tem o potencial de capacitar os estudantes a desenvolver uma variedade de habilidades essenciais. Estas incluem, mas não se limitam a: Pensamento Algorítmico, Abstração, Decomposição, Aprendizagem Colaborativa, Resolução de Problemas, Criatividade, Raciocínio Lógico e Interpretação Textual. Consolidando as múltiplas perspectivas e definições do Pensamento

¹² "Framework - Saiba o que é | Glossário de produto da PM3."

<<https://www.cursospm3.com.br/glossario/framework/>>. Acessado em 16 jan.. 2024.

Computacional apresentadas pelos autores citados até o momento, podemos conceber as habilidades do Pensamento Computacional como um conjunto abrangente de competências cognitivas e técnicas. Estas competências capacitam indivíduos a abordar problemas e desafios de forma eficiente, lógica e criativa, empregando princípios e métodos provenientes da ciência da computação e da resolução algorítmica. Importante notar que essas habilidades transcendem o âmbito da programação e da tecnologia, abarcando uma diversidade de aptidões que podem ser aplicadas em variados contextos da vida cotidiana e profissional (Wing, 2006).

Mesmo após diversos estudos e quase uma década de esforços para definir PC, ainda existem críticas que sugerem que não sabemos o que PC significa ou sua forma de medir (Kurshan, 2016, tradução nossa).

Baseando-se nessa indefinição do termo Pensamento Computacional, Kalelioglu (2016) realiza uma revisão sistemática a partir de 125 artigos, com objetivo de tentar identificar uma definição para o termo, infelizmente não se chegou a uma definição, porém a partir desse trabalho ele gerou uma nuvem de palavras, destacando quais eram as palavras quem mais eram representadas nos artigos.



Figura 1 - Palavras mais usadas nas definições de pensamento computacional.
Fonte: Kalelioglu, Gulbahar, & Kukul, 2016.

Analisando a representação visual na Figura 1, é possível identificar que algumas das palavras em destaque estão relacionadas às práticas do Pensamento Computacional, conforme citado por Resnick (2012) e referenciadas na seção 3.4.2 deste estudo. No entanto, é fundamental ressaltar, conforme destacado pelo autor, que as práticas representam a maneira pela qual essas habilidades do Pensamento Computacional são concretizadas na prática, especialmente no contexto de criação de mídias interativas. Dessa forma, as habilidades do Pensamento Computacional constituem a base cognitiva, enquanto as práticas-chave refletem a aplicação concreta dessas habilidades em situações do mundo real, sobretudo em projetos

criativos e interativos. Ambos os elementos desempenham papéis fundamentais na compreensão integral e eficaz do Pensamento Computacional.

No Quadro 5 é possível observar a descrição de algumas das habilidades do pensamento computacional percebidas por diversos autores em seus estudos.

Quadro 5 - Habilidades do PC segundo diversos autores.

Habilidades do Pensamento Computacional	Descrição da Habilidade	Autor/Ano
Representação	Representar informações e dados de maneira clara e organizada	Brennan & Resnick (2012)
Automação	Automatizar tarefas para reduzir a possibilidade de erro humano	Brennan & Resnick (2012)
Teste e Depuração	Identificar e corrigir erros em um programa de computador	Barr & Stephenson (2011)
Pensamento sistêmico	Compreender como partes individuais de um sistema se relacionam e interagem	Grover & Pea (2013)
Simulação	Criar modelos para representar sistemas do mundo real	Brennan & Resnick (2012)
Pensamento criativo	Desenvolver soluções inovadoras para problemas existentes	Grover & Pea (2013)
Pensamento crítico	Avaliar informações e argumentos com ceticismo e julgamento informado	Barr & Stephenson (2011)
Colaboração	Trabalhar em equipe para alcançar um objetivo comum	Brennan & Resnick (2012)
Comunicação	Expressar ideias e informações de maneira clara e concisa	Grover & Pea (2013)
Fluência tecnológica	Usar a tecnologia de maneira eficaz e responsável	Wing (2006)
Projeto	Planejar e projetar soluções para problemas	Grover & Pea (2013)
Segurança	Conhecer e aplicar práticas de segurança cibernética	Grover & Pea (2013)
Análise de dados	Coletar, organizar e analisar informações usando ferramentas de TI	Brennan & Resnick (2012)
Pensamento probabilístico	Entender e utilizar conceitos de probabilidade e estatística para resolver problemas	Wing (2006)
Pensamento espacial	Visualizar e manipular objetos em um espaço tridimensional	Barr & Stephenson (2011)

Fonte: elaborado pelo autor.

Para uma análise mais aprofundada das habilidades do Pensamento Computacional, exploraremos exemplos práticos que ilustram os quatro pilares fundamentais do Pensamento Computacional baseado nos exemplos de Rodrigo (2021). Esse aprofundamento teórico nos fornecerá uma base sólida para abordar o terceiro objetivo específico deste estudo. De acordo com o site BBC Bitesize (2015), o Pensamento Computacional é estruturado em quatro

pilares essenciais: Decomposição, Reconhecimento de Padrões, Abstração e Algoritmo. Essas habilidades do Pensamento Computacional têm sido amplamente discutidas na literatura acadêmica e são também reconhecidas por Jeannette Wing como princípios-chave do Pensamento Computacional.

No âmbito do Pensamento Computacional, Jeannette Wing identifica a Decomposição (Decomposition) como um dos quatro pilares fundamentais em seu artigo "Computational Thinking" de 2006. A Decomposição é definida como a habilidade de desmembrar um problema complexo em componentes menores e mais gerenciáveis. Essa abordagem simplifica o processo de desenvolvimento, resolução e administração de problemas, permitindo a resolução de desafios complexos de maneira mais acessível. Além disso, a Decomposição aprimora a compreensão de novos cenários e viabiliza a criação de sistemas de grande porte (Wing, 2006). A Figura 2 exemplifica como é possível fragmentar um problema maior em partes menores para facilitar sua resolução.

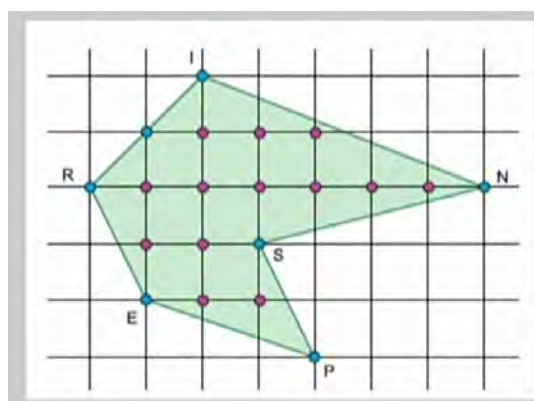


Figura 2 - Polígono irregular.

Fonte:

<<https://www.passeidireto.com/arquivo/92858429/univesp-pensamento-computacional-aula-2-pilares-do-pc-final-ok>>. Acesso em 20 de junho de 2022.

Utilizando o exemplo de Rodrigo (2021), para ilustrar o princípio da Decomposição no contexto do Pensamento Computacional, considere o exemplo de como calcular a área de um polígono irregular, ou seja, um polígono que não possui uma forma conhecida.

Suponha que saibamos como calcular a área de um polígono regular, como um quadrado ou um triângulo. Para calcular a área de um polígono irregular, podemos aplicar o princípio da Decomposição da seguinte maneira:

1. Dividir o Polígono Irregular: Primeiro, dividimos o polígono irregular em partes menores, como triângulos, retângulos ou outras formas regulares que conhecemos como calcular.
2. Calcular as Áreas das Partes: Em seguida, calculamos a área de cada uma das partes menores que resultaram da divisão.
3. Somar as Áreas: Por fim, somamos as áreas das partes menores calculadas. A soma das áreas das partes resulta na área total do polígono irregular.

Esse processo exemplifica a Decomposição, pois quebramos um problema complexo (calcular a área de um polígono irregular) em partes menores e mais simples (calcular áreas de polígonos regulares), tornando o problema mais gerenciável e passível de solução.

Outro pilar do PC é o reconhecimento de padrões. O Reconhecimento de Padrões é a capacidade de identificar tendências, regularidades e características em dados, informações ou situações com base em observações e análises. “Essa habilidade permite a detecção de elementos repetitivos ou previsíveis em um conjunto de dados, o que, por sua vez, possibilita a aplicação de soluções conhecidas a novos contextos com base na identificação de padrões familiares” (Haykin, 2009, p. 70, tradução nossa). O hábito de identificar padrões nos acompanha desde a infância, é uma construção continuada e o nosso repertório de padrões não para de crescer e de se reconstruir. Quanto mais padrões encontramos, mais fácil e rápida será a nossa tarefa geral de solução de problemas.

Associado ao pensamento computacional, cumpre salientar a importância dos algoritmos e de seus fluxogramas, que podem ser objetos de estudo nas aulas de Matemática. “[...] Outra habilidade relativa à álgebra que mantém estreita relação com o pensamento computacional é a identificação de padrões para se estabelecer generalizações, propriedades e algoritmos.” (Brasil, 2017)

As aplicações do Reconhecimento de Padrões são variadas, abrangendo diversas situações. Por exemplo:

- Ao analisar uma sequência de números, é possível prever o próximo número na sequência.
- A partir da posição do sol no céu, podemos estimar a hora do dia.
- Observando a direção em que os galhos de uma árvore se movem, é possível identificar o sentido do vento.
- Com base nas configurações de chuva, podemos antecipar a possibilidade de chuva.

- No campo da medicina, o Reconhecimento de Padrões é crucial para diagnosticar doenças, levando em consideração sintomas, aparência e comportamento do paciente.

Um exemplo notável é a capacidade de reconhecer quando uma pessoa está sofrendo um AVC (Acidente Vascular Cerebral) com base em padrões observados. Figura 3.



Figura 3 - Sintomas de AVC.

Fonte: < <https://www.tuasaude.com/sintomas-de-avc-acidente-vascular-cerebral/> >. Acesso em 20 de junho de 2022.

A abstração se apresenta como uma das ferramentas mais importantes do PC, ela corresponde a capacidade de extrair informações relevantes de contextos genéricos, sendo que é possível apresentar essa representação abstrata dos conceitos de maneira compreensível através da codificação, que é escrever um processo de maneira que possa ser executável por um computador (Wing, 2008; Gadanidis, 2017; Sawaya, 1999).

Para Riley e Hunt (2014), a abstração está incluída no conceito de resolução de problemas, sendo definida como uma capacidade que permite focalizar nas características mais importantes do problema, de maneira a desprezar detalhes não tão importantes, o que Krauss e Prottzman (2016) e Barbosa (2019) complementam dizendo que essa ação funciona para alcançar uma solução de um problema mais geral, dando significado para essa solução, de modo a validá-la para diferentes problemas.

Rodrigo (2021) coloca esse pilar como sendo o de maior importância para o pensamento computacional, por ser utilizado em diversos momentos, por exemplo:

- Na escrita de perguntas,
- Na escrita do algoritmo e de suas iterações,
- Na compreensão e organização de módulos em um sistema,
- Na seleção dos dados importantes,
- Na natureza de um indivíduo, se comparando a um robô, etc...

Rodrigo (2021) ainda cita o exemplo da coleta seletiva: Figura 4



Figura 4 - Coleta Seletiva.

Fonte: <<https://www.projetopiracanjuba.org.br/-coleta-seletiva-e-reciclagem---entenda-a-diferenca/>>. Acesso em 20 de junho de 2022.

Em relação a figura acima, não conseguimos exibir todos os objetos que podem ser de plástico, mas conseguimos apresentar um único material que engloba todos esses objetos, por isso, na lata de lixo vermelha, só jogamos objetos de plástico. Essa abstração facilita o bom uso desse recurso de separação de lixo, uma vez que se tivéssemos que escrever todos os objetos de plástico, estaríamos despendendo um número enorme de recursos, o que inviabilizaria a coleta seletiva.

O pilar Algoritmo é o pilar que agrega os demais pilares (Wing, 2014). É uma sucessão finita de etapas, cada qual executável em um tempo limitado, por um agente computacional, natural ou sintético. A idealização de um algoritmo computacional passa pelo procedimento de decomposição, reconhecimento de padrões e abstração.

Sendo assim, percebemos que os pilares do pensamento computacional têm impactos significativos em nossa capacidade de resolver problemas, proporcionando eficiência na decomposição, compreensão profunda por meio do reconhecimento de padrões, simplificação via abstração e eficácia na criação de algoritmos. Essa abordagem transformadora não apenas redefine a resolução de desafios, mas também impulsiona a inovação, deixando um impacto duradouro em nossa capacidade de enfrentar os desafios contemporâneos de maneira eficaz e criativa. Em nosso trabalho, buscamos identificar essas habilidades (objetivo três) e descrever seus impactos nos estudantes do ensino médio ao desenvolverem jogos digitais e dispositivos robóticos.

2.4 DIMENSÕES DO PENSAMENTO COMPUTACIONAL SEGUNDO MITCHEL RESNICK - CONCEITOS, PRÁTICAS E PERSPECTIVAS

Para articular a estrutura do pensamento computacional neste trabalho, utilizaremos os conceitos desenvolvidos por Resnick (2012). Conforme destacado por Resnick (2012), nos últimos anos, ao estudar as atividades na comunidade online do Scratch e ao participar de workshops do Scratch, foi possível desenvolver uma definição ampla de pensamento computacional, composta por três dimensões principais: conceitos computacionais (os princípios aplicados pelos designers ao programar), práticas computacionais (os métodos desenvolvidos pelos designers durante a programação) e perspectivas computacionais (a visão que os designers adquirem sobre o mundo ao seu redor e sobre si mesmos). As observações e entrevistas desempenharam um papel fundamental na compreensão do desenvolvimento ao longo do tempo dos criadores, com suas participações e portfólios de projetos que abrangem semanas a vários anos. Nas seções a seguir, apresentaremos detalhadamente as dimensões citadas por Resnick (2012).

2.4.1 Conceitos de Pensamento Computacional

Seguindo as observações de Resnick (2012), à medida que os jovens se dedicam à criação de mídias interativas através do *software* Scratch, eles exploram um conjunto de conceitos computacionais que são universais em várias linguagens de programação. Resnick (2012) identificou sete conceitos particularmente valiosos em uma ampla gama de projetos do Scratch, que podem ser aplicados em diferentes contextos, tanto relacionados à programação quanto em outras áreas. Esses conceitos incluem:

- ★ **Sequências** - Um conceito fundamental na programação envolve a representação de uma atividade ou tarefa específica como uma sequência de passos ou instruções individuais que o computador pode executar.
- ★ **Loops** - Loops representam um mecanismo que possibilita a repetição da mesma sequência de ações várias vezes.
- ★ **Eventos** - Segundo Resnick, o conceito de eventos envolve a ideia de que ações específicas ou condições desencadeiam reações ou respostas em um programa ou sistema, permitindo que os computadores respondam a estímulos do ambiente ou

interações do usuário. Esses eventos podem desencadear a execução de blocos de código ou acionar ações específicas, tornando os programas mais interativos e sensíveis ao contexto.

- ★ **Paralelismo** - O conceito de paralelismo refere-se à capacidade de executar múltiplas ações ou tarefas simultaneamente em um programa de computador. Isso implica que diferentes partes do programa podem funcionar de forma independente e ao mesmo tempo, o que permite que um programa realize várias operações de forma concorrente. O paralelismo é uma ideia fundamental na programação e no Pensamento Computacional, uma vez que permite que os programas sejam mais eficientes e capazes de lidar com tarefas complexas de maneira mais eficaz.
- ★ **Condicionais** - Outro conceito essencial em ambientes interativos é o uso de condicionais, que se refere à habilidade de tomar decisões com base em condições específicas, possibilitando a expressão de múltiplos resultados.
- ★ **Operadores** - Os operadores desempenham um papel fundamental ao possibilitar expressões matemáticas, lógicas e de strings, capacitando o programador a efetuar manipulações em números e sequências de caracteres. O Scratch oferece suporte para uma variedade de operações matemáticas, que abrangem desde adições, subtrações, multiplicação e divisão, até funções trigonométricas como o seno e operações de exponenciação. Além disso, o Scratch também permite operações com strings, incluindo a concatenação e a determinação do comprimento das sequências de caracteres.
- ★ **Dados** - Os dados englobam o processo de armazenar, recuperar e atualizar valores. No ambiente do Scratch, há atualmente dois tipos de contêineres para dados: variáveis (que têm a capacidade de armazenar um único número ou sequência de caracteres) e listas (que permitem a retenção de uma coleção de números ou sequências de caracteres). O uso de variáveis frequentemente se manifesta como um incentivo significativo para jovens criadores, especialmente ao rastrear e exibir pontuações em jogos.

2.4.2 Práticas do Pensamento Computacional

Conforme apontado por Resnick (2012), a partir das entrevistas e observações realizadas com jovens designers, tornou-se evidente que a delimitação do pensamento computacional unicamente em torno de conceitos revelava-se insuficiente para abordar outros elementos do processo de aprendizado e da participação desses designers. Para articular a estrutura de pensamento computacional é necessário descrever os processos de construção, as práticas de design durante a criação de projetos computacionais. De acordo com Resnick (2012), as práticas computacionais enfocam o processo de pensamento e aprendizado, abordando não apenas o que está sendo aprendido, mas também o modo como ocorre o aprendizado. Nas palavras do autor: ...as práticas computacionais concentram-se no processo de pensar e aprender, indo além *do que* você está aprendendo, até *como* você está aprendendo (Resnick, 2012, p. 6, tradução nossa).

Resnick (2012) também identificou um conjunto de práticas que são comuns quando jovens estão envolvidos no desenvolvimento de mídias interativas. Estas práticas-chave incluem: ser incremental e iterativo, realizar testes e depuração, reutilizar e remixar, e aplicar abstração e modularização.

- ★ **Ser incremental e iterativo:** O design é um processo adaptativo que envolve a progressão gradual e a experimentação, com ciclos iterativos de imaginação e construção, permitindo o aprimoramento contínuo do projeto.
- ★ **Testar e depurar:** Problemas e desafios inesperados são uma parte intrínseca do processo de design. Os designers desenvolvem práticas de teste e depuração para lidar com problemas, incluindo identificação, leitura de scripts, experimentação, reescrita de scripts e consulta a outras pessoas.
- ★ **Reutilizar e remixar:** A prática de construir sobre o trabalho de outros é uma tradição na programação, estimulada pelo ambiente online do Scratch. Isso apoia o desenvolvimento de habilidades de leitura de códigos e levanta questões sobre colaboração, propriedade e autoria.
- ★ **Abstrair e modularizar:** Os designers empregam abstração e modularização em vários níveis, desde a concepção inicial do problema até a tradução de conceitos em sprites individuais e pilhas de código. Isso permite a criação de projetos complexos ao reunir peças menores e facilita a organização, teste e depuração.

2.4.3 Perspectivas do Pensamento Computacional

As perspectivas do pensamento computacional, conforme descritas por Resnick (2012), referem-se a diferentes maneiras pelas quais os jovens designers, ao trabalharem com a plataforma Scratch, desenvolvem uma compreensão mais ampla de si mesmos, de suas relações com os outros e do mundo tecnológico ao seu redor. As três perspectivas específicas mencionadas por ele são:

- ★ **Expressando** - Refere-se à mudança na visão dos jovens em relação ao uso da computação. Em vez de apenas consumir tecnologia, os pensadores computacionais veem a computação como um meio de expressão e design. Eles reconhecem a capacidade de criar e expressar suas ideias por meio da programação e design de projetos interativos.
- ★ **Conectando** - Destaca a importância da interação social na prática criativa de um "Scratcher" (usuário do Scratch). A criatividade e aprendizagem são enriquecidas por interações face a face ou online. Os jovens reconhecem o valor de criar com outros, colaborar em projetos, obter ajuda e inspiração, bem como criar para outros, apreciando um público autêntico para suas criações.
- ★ **Questionando** - Refere-se à capacidade dos jovens de questionar e compreender o mundo tecnológico que os cerca. Os pensadores computacionais desenvolvem uma mentalidade de questionamento, buscando entender as tecnologias complexas e sentindo-se capacitados para fazer perguntas sobre como essas tecnologias funcionam. Além disso, eles podem questionar as limitações de ferramentas computacionais, participando ativamente no design e modificação de ambientes como o Scratch.

Essas perspectivas ampliam o entendimento do pensamento computacional além de habilidades técnicas específicas, destacando a importância da expressão criativa, interação social e questionamento crítico no desenvolvimento de competências computacionais.

Em resumo, o pensamento computacional e o Construcionismo de Papert são abordagens educacionais complementares que incentivam a experimentação, a criatividade e a resolução de problemas. Ao integrar essas abordagens na educação, os alunos podem aprender de forma mais significativa e natural, desenvolvendo habilidades essenciais para o século XXI e preparando-se para os desafios do mundo moderno.

3 PERCURSO METODOLÓGICO

Na seção a seguir, o método utilizado está detalhado.

3.1 PESQUISA QUALITATIVA

A natureza ou abordagem deste trabalho é a pesquisa qualitativa, pois de forma subjetiva busca interpretar e analisar os dados verificados baseados na análise interpretativa. De acordo com Bogdan e Biklen (1982), na pesquisa qualitativa, o processo é mais enfatizado que o produto e, portanto, a preocupação está em retratar a perspectiva dos participantes em interação.

A escolha do viés qualitativo permite uma análise minuciosa das experiências e percepções dos estudantes. Conforme observado por Lankshear & Knobel (2004), proporciona insights valiosos sobre como o PC se desenvolve em contextos educacionais específicos. Além disso, essa abordagem capacita a exploração de nuances e fatores contextuais que influenciam o desenvolvimento do PC durante o processo de construção de jogos e dispositivos robóticos.

A pesquisa qualitativa também se destaca na revelação de aspectos subjetivos e sociais do processo de construção, bem como na compreensão do papel da interação entre estudantes, professores e ambientes educacionais (Triviños, 1987 ; Bogdan E Biklen, 1994).

Ela ainda oferece flexibilidade metodológica, permitindo que os pesquisadores escolham e adaptem técnicas apropriadas, como entrevistas em profundidade, observação participante e análise de conteúdo, conforme discutido por Gil (2002) e Bardin (2016).

Portanto, adotamos os pressupostos da pesquisa qualitativa, pois buscamos “[...] atingir aspectos humanos sem passar pelos crivos da mensuração, sem partir de métodos previamente definidos e, por conseguinte, sem ficar presos a quantificadores e aos cálculos recorrentes” (Bicudo, 2006, p. 107). Negamos a neutralidade do pesquisador durante todo processo investigativo, pois consideramos que há sempre um aspecto subjetivo a ser considerado (Triviños, 2009; Bogdan; Biklen, 1994). Nessa perspectiva, torna-se indissociável a convivência com a ideologia em nossa travessia de investigação, não obstante, é possível controlá-la crítico e intencionalmente de modo a fazer ciência (Demo, 2011).

Em relação aos objetivos, este estudo pode ser classificado como sendo uma observação participante, pois de acordo com Bogdan e Biklen, ela procura a resposta para o objetivo geral e envolve a participação ativa do pesquisador no campo com um grupo seletivo de pessoas, assim a natureza qualitativa desta pesquisa se alinha ao objetivo central de examinar o processo de desenvolvimento do Pensamento Computacional de estudantes do ensino médio ao construírem jogos digitais.

3.2 CONTEXTO DE PESQUISA

Com o objetivo de examinar o processo de desenvolvimento do Pensamento Computacional de estudantes do ensino médio, em um ambiente construcionista, quando constroem jogos digitais, essa pesquisa utilizou-se do trabalho realizado nas atividades do projeto Mattics, no Instituto Federal Goiano - Campus Avançado Ipameri.

O IF Goiano foi criado por meio da Lei 11.892, de 29 de dezembro de 2008. Atualmente, é composto pela Reitoria e pelos campi: Campos Belos, Ceres, Cristalina, Iporá, Morrinhos, Posse, Rio Verde, Trindade, Urutaí; os campi avançados: Catalão, Hidrolândia e Ipameri e o Polo de Inovação Rio Verde. A Figura 5 apresenta a localização geográfica de cada Campi no estado de Goiás.



Figura 5 - Localização geográfica do Estado de Goiás com destaque para as unidades do IF Goiano.

Fonte: <<https://www.ifgoiano.edu.br/home/index.php/ultimas-noticias/1579-campus-avancados-estarao-vinculados-a-reitoria-a-partir-de-janeiro>>. Acesso em 22 de junho de 2022.

O Campus Avançado Ipameri, conforme o PDI 2019-2023, inicialmente vinculado ao Campus Urutaí, iniciou suas atividades em fevereiro de 2014, após receber as instalações do Campus IV da PUC-Goiás. Atualmente, oferece cursos técnicos e de graduação, com planos de expansão para cursos na área de Engenharia, com foco em Engenharia Civil e Arquitetura.

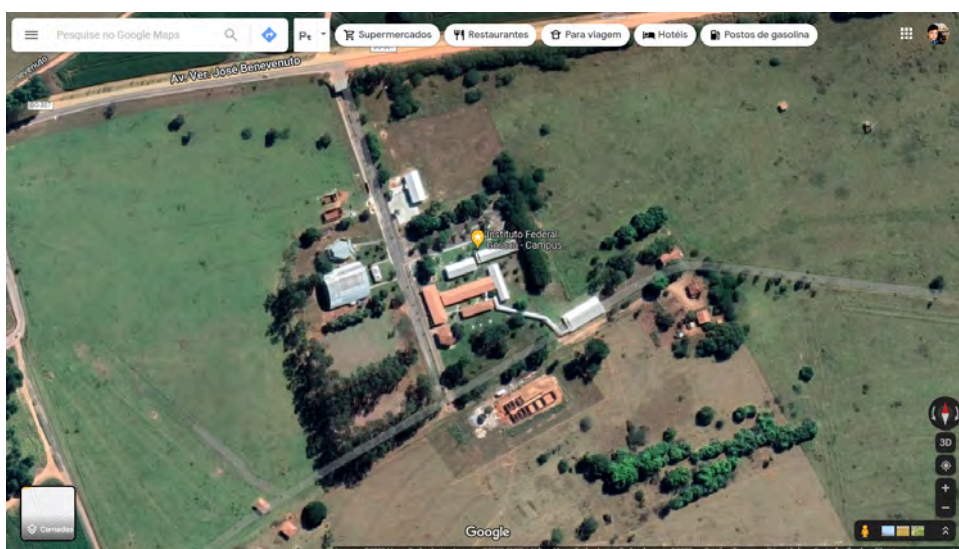


Figura 6 - Vista aérea do Campus Avançado Ipameri.

Fonte: Google Maps. Acesso em 22 de junho de 2022.

Quanto à infraestrutura, o campus conta com diversas salas de aula, laboratórios de informática, laboratórios específicos, biblioteca, auditório e quadras esportivas. O corpo docente é composto por 23 professores, incluindo especialistas, mestres e doutores. Há também 12 servidores técnico-administrativos em educação. O campus possui diversos projetos de ensino e extensão em andamento.

O município de Ipameri está localizado na região sudeste de Goiás, a uma distância de 193 km da capital do estado e 250 km do Distrito Federal. Em 2022, a população estimada de Ipameri era de 25.548 habitantes, ocupando cerca de 1% do território goiano.

A cidade limita-se com outros municípios, incluindo Campo Alegre, Catalão, Caldas Novas, Corumbáiba, Cristalina, Goiandira, Luziânia, Nova Aurora, Orizona, Paracatu, Pires do Rio e Urutaí. Em termos econômicos, o PIB do município em 2015 foi de \$ 875.626,45 (em escala de milhar), com destaque para os setores de Agropecuária, Indústria, Serviços, Administração, Defesa e Saúde Públicas e Impostos líquidos sobre produtos.

Inicialmente o projeto Mattics foi desenvolvido em um colégio público municipal, localizado em um município da região metropolitana de Goiânia no segundo semestre de 2015, e permitiu a alunos do Ensino Fundamental unificar matemática e linguagem de programação para criar jogos digitais e compreender conceitos matemáticos de forma problematizada. Conforme Azevedo (2017), nos primeiros encontros do projeto, os alunos puderam compreender, por meio de atividades investigativas e exploratórias, os conceitos computacionais e as características do fazer matematicamente na construção de pequenas animações no *software* Scratch. Puderam também explorar e debater ideias uns com os outros. Ao longo do tempo, no projeto, eles foram desafiados a estabelecer temáticas, como, por exemplo, meio ambiente, a escola que queremos, etc., para construir seus próprios jogos.

Em meados de 2018, por meio de um trabalho de extensão, o projeto Mattics foi implementado pela primeira vez no Instituto Federal Goiano - Campus Avançado Ipameri. O projeto em questão, tinha como um de seus objetivos produzir jogos digitais e dispositivos robóticos adaptados que resultou em um trabalho realizado coletivamente entre docente e discentes no Hospital Dia do Idoso.

Atualmente, o Hospital Dia do Idoso, situado na cidade de Anápolis - Goiás é reconhecido em nível nacional por seu atendimento especializado e técnico aos idosos, destacando-se principalmente no tratamento da doença de Parkinson. É relevante observar que Anápolis foi fundada em 1907 e, na atualidade, abriga uma população de 398.869 mil habitantes, sendo que 56.489 possuem 60 anos ou mais, conforme dados do IBGE em 2022.

Desde 2018, cerca de 30 jogos digitais e 15 dispositivos robóticos (específicos ao tratamento Parkinsoniano) foram criados pelos estudantes com a mediação do professor, com o auxílio de profissionais da área da computação, engenharia e da área médica - beneficiando pacientes com o retardamento de sintomas da doença de Parkinson.

As atividades relatadas nesta pesquisa, aconteceram no primeiro e segundo semestre de 2023, elas aconteciam sempre no período vespertino, especificamente nas terças-feiras. Foram planejados e executados 10 encontros, cada um com a duração de 3 horas, sendo o último realizado no Hospital Dia do Idoso em Anápolis-GO. A concepção e organização das atividades foram lideradas pelo professor de Matemática do Instituto Federal Goiano - Campus Avançado Ipameri, Greiton Toledo de Azevedo, contando com contribuições de diversos profissionais das áreas de saúde e computação.

No dia 15 de setembro de 2023, os estudantes tiveram a oportunidade de acompanhar as sessões de fisioterapia no Hospital Dia do Idoso, onde puderam observar o tratamento de Parkinson em ação e colocar em prática os jogos que eles próprios desenvolveram ao longo do ano.

No decorrer do ano de 2023, sob a orientação do professor e com a colaboração de profissionais das áreas de Computação e Saúde, os estudantes criaram 3 jogos e 3 dispositivos robóticos que são o foco do presente trabalho.

Os detalhes referentes aos encontros, à instituição educacional, à turma que participou da pesquisa e ao software utilizado no desenvolvimento dos jogos digitais serão minuciosamente explorados na próxima seção.

3.2.1. Perfil do Estudante - Amostra

A amostra desta pesquisa consiste na turma de 18 estudantes que participam do projeto de extensão Mattics. Os membros do projeto Mattics são estudantes atualmente matriculados nos cursos técnicos integrados de Redes de Computadores e Comércio, com idades compreendidas entre 16 e 18 anos. Esta investigação se concentrará especificamente nos alunos que fazem parte da turma do projeto Mattics no ano de 2023.

A seleção dos participantes do projeto foi conduzida de acordo com critérios estabelecidos pelo professor Greiton. Os estudantes passaram por um processo de inscrição e entrevista, sendo que critérios como desempenho acadêmico em matemática não foram

considerados. O objetivo do professor era garantir uma amostra que representasse fielmente a realidade escolar. No que diz respeito à seleção dos participantes específicos para esta pesquisa, o pesquisador optou por escolher os estudantes que demonstraram maior assiduidade nos encontros.

Todos os participantes e colaboradores desta pesquisa receberam os termos de consentimento, anuência e autorização. É relevante observar que todos eles expressaram seu desejo de participar da pesquisa sem a preservação do anonimato de suas identidades. Além disso, é importante mencionar que o trabalho de pesquisa de mestrado foi aprovado na Plataforma Brasil pelo Comitê de Ética da Universidade Iguazu - UNIG.

3.2.2 Sujeitos da Pesquisa

Em 2023, conduzimos um estudo abrangente com o respaldo da gestão das instituições IF Goiano e Hospital Dia do Idoso. A colaboração essencial foi fornecida pela equipe pedagógica e administrativa do Campus Ipameri, a qual incluiu o professor de Matemática, Doutor Greiton Toledo de Azevedo. Esta pesquisa envolveu a participação de estudantes do curso Técnico em Redes de Computadores e Técnico em Comércio Integrados ao Ensino Médio do IF Goiano - Campus Avançado Ipameri e pacientes idosos de faixa etária média de 65 anos de idade. Esses pacientes incluídos no projeto foram acompanhados de perto pela fonoaudióloga e fisioterapeutas do referido hospital, participante da pesquisa.

3.3 JOGOS DIGITAIS E DISPOSITIVOS ROBÓTICOS: UMA ABORDAGEM CONSTRUCIONISTA

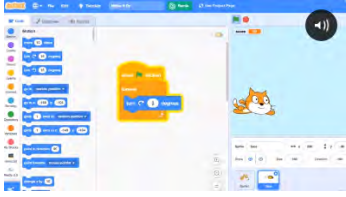
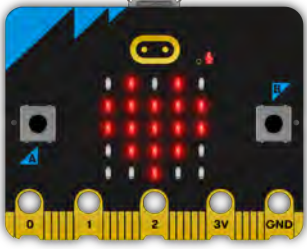
O projeto Mattics ocorria fora do horário normal das aulas, no tempo para as atividades extra-curriculares, de semana a semana, durante o ano letivo e, no fim do projeto, os estudantes puderam viajar até a cidade de Anápolis para acompanharem as sessões de fisioterapia no Hospital Dia do Idoso.


Professor e idealizador do projeto Mattics, Greiton Toledo de Azevedo, afirma que baseando-se na ideia construcionista do conhecimento desenvolvida por Papert, o aluno deixa de ser consumidor da tecnologia e passa a ser o construtor, discute e expressa ideias ao mesmo tempo que constrói conhecimento matemático.

Tal projeto, especialmente na dimensão de investigação, se organiza como estratégia para atender os alunos que apresentavam dificuldade em matemática nas aulas do professor-pesquisador, sem deixar de lado aqueles que apresentavam bons rendimentos. É uma investigação que se justifica ainda pelo interesse dos próprios estudantes em conhecer a estrutura do funcionamento dos jogos digitais e, principalmente, ter a oportunidade de construí-los. Por outro lado, o movimento dinâmico desse projeto idealiza-se não somente por satisfazer o gosto pessoal dos estudantes por tais jogos, nem tampouco prepará-los a seguir uma carreira profissional como programadores profissionais, mas, antes, foi pensado no sentido de incentivá-los, por meio da construção de tecnologias, a discutir ideias e expressá-las, além de construir o conhecimento matemático (Azevedo. G.T, 2017, p. 84).

No ano de 2023, no âmbito do projeto Mattics, uma variedade de ferramentas computacionais foi empregada, abrangendo robôs, sensores e *softwares*, a saber: "Materiais de Papelaria" que envolvem objetos recicláveis como papelão, plástico, metais, etc.; "Sucatas" que incluem rodas, paletes, regadores, entre outros; "Componentes Eletrônicos" como fios de cobre, pilhas, baterias, LEDs, entre outros; "*Software*" abrangendo Scratch e Geogebra; "Placas Robóticas" contemplando Makey-Makey e BBC: Micro-Bit. O Quadro 1 subsequente oferece a descrição e os dados dos principais recursos midiáticos utilizados.

Quadro 6 - Ferramentas utilizadas no projeto Mattics.

Ferramenta	Descrição
	<p>O Scratch é a maior comunidade do mundo de programação para crianças e uma linguagem de programação com uma interface visual simples que permite que os jovens criem histórias, jogos e animações digitais. Cf.: https://scratch.mit.edu/about</p>
	<p>GeoGebra é um software dinâmico de matemática para todos os níveis de educação que reúne geometria, álgebra, planilhas, gráficos, estatísticas e cálculos em uma única plataforma. Além disso, o GeoGebra oferece uma plataforma online com mais de 1 milhão de recursos gratuitos criados por nossa comunidade em vários idiomas. Cf.: https://www.geogebra.org/about</p>
	<p>O BBC micro:bit é um computador de bolso que mostra como <i>software</i> e hardware funcionam juntos. Possui display de luz LED, botões, sensores, alto-falante, microfone e muitos outros recursos de entrada/saída que, quando programados, permitem interagir com você e seu mundo. Cf.: https://microbit.org/buy/bbc-microbit-single/</p>

	<p>O MAKEY-MAKEY é uma placa robótica que possibilita desenvolver controles e dispositivos específicos que podem ser conectados a softwares, como o Scratch. Cf.: https://makeymakey.com/</p>
---	---

Fonte: elaborado pelo autor

Segundo o professor e idealizador do projeto Mattics, a seleção dos eletrônicos, por parte do professor, foi pautada na priorização de aplicativos acessíveis e de fácil utilização, com o objetivo de não limitar a capacidade dos alunos em desenvolver ideias e soluções para os problemas propostos. Além disso, buscou-se uma linguagem de programação dinâmica e intuitiva, evitando armadilhas técnicas e sintaxes desnecessárias como por exemplo pontos e vírgula. Optou-se, também, pelo uso de placas com características acessíveis e versáteis, facilitando a conexão entre condutores de corrente elétrica e materiais de suporte, promovendo a construção de circuitos elétricos por meio de alumínio, placas, cobre, baterias, entre outros materiais.

A concepção dos eletrônicos foi orientada para que fossem explorados pelos estudantes no papel de cientistas no ambiente de aprendizagem construcionista, evitando assim a mera reprodução de ideias mecânicas ou a repetição de modelos de prototipagem. Destacou-se a valorização das produções que empregaram eletrônicos, materiais de baixo custo e sustentáveis no contexto de formação financiados pelo professor e estudantes. Considerando os desafios específicos da escola pública, esses critérios foram considerados indispensáveis para a criação dos produtos robóticos voltados ao tratamento de Parkinson.

3.3.1 Scratch

Para a construção dos jogos digitais foi utilizada a plataforma Scratch. Scratch é uma linguagem de programação baseada nos princípios construcionistas dos aplicativos Logo e Etoys (Maloney et al., 2010). O *software* de criação de jogos Scratch é uma ferramenta didática idealizada para o trabalho com crianças e adolescentes. Foi desenvolvido pelo *Lifelong Kindergarten Group do MIT Media Lab* e é amplamente utilizado para ensinar

conceitos de programação e pensamento computacional a iniciantes. A principal característica do Scratch é sua interface de programação baseada em blocos. Em vez de escrever código tradicional, os usuários montam programas arrastando e encaixando blocos que representam comandos e ações. Isso torna a programação mais acessível, especialmente para aqueles que não têm experiência prévia em codificação. É uma das principais ferramentas para a utilização de programação em colégios, por ser didaticamente adequado e gratuito. Por conta desses fatores o *software* Scratch é a plataforma de construção de jogos utilizada no projeto Mattics.

O Scratch pode ser baixado para ser utilizado offline ou pode ser acessado pelo próprio navegador através do link <https://scratch.mit.edu/>. No mesmo link tutoriais sobre a utilização estão disponíveis.

3.4 A ELABORAÇÃO DE UMA PROPOSTA PEDAGÓGICA CONSTRUCIONISTA NO ENSINO MÉDIO INTEGRADO

Nessa seção será discutido em detalhes como aconteceram as etapas de criação de jogos durante o desenvolvimento das atividades do projeto Mattics realizadas entre os meses de março e outubro de 2023, no laboratório de Matemática, salas de aula do IF Goiano - Campus Avançado Ipameri e no Hospital Dia do Idoso, sob supervisão do professor da disciplina de Matemática.

O primeiro encontro foi realizado no dia 16 de março de 2023, no Laboratório de Matemática do Campus. Após coletarmos as assinaturas dos alunos dos Termos de Consentimento Livre e Esclarecido (TCLE) e de Assentimento Livre e Esclarecido (TALE), o professor explicou o funcionamento do projeto para os novos integrantes. No encontro, foram apresentados aos alunos 4 jogos: Labirinto, Mario, Ping Pong e Cadeirante. As atividades do encontro foram divididas em duas partes. Na primeira parte, os alunos foram divididos em grupos de dois a três estudantes por computador, e receberam a tarefa de escolher um dos jogos para analisar o código-fonte, corrigir possíveis erros (bugs) e adicionar novas funcionalidades. Em seguida, os estudantes foram desafiados a explicar uma seção específica do código do jogo escolhido pelo grupo ao qual pertenciam.



Figura 7 - Estudantes Engajados no Laboratório de Matemática: Uma Cena de Colaboração Ativa onde Estudantes Analisam, Identificam e Corrigem Erros, Além de Desenvolverem Novas Funcionalidades para os Jogos Propostos.

Fonte: acervo do autor



Figura 8 - Estudantes explicando seções do código do jogo Labirinto.

Fonte: acervo do autor

Na segunda parte da aula, orientados pelo professor, os alunos construíram um *joystick*¹³ usando copos cheios de água e uma placa robótica Makey-Makey, que seria utilizado coletivamente, para jogar o jogo Labirinto. Nessa atividade colaborativa, os estudantes precisaram se unir, dando as mãos, para que o fio negativo do *joystick* fechasse o contato e o *joystick* pudesse funcionar para controlar a direção do personagem dentro do jogo.

¹³ "Definição de joystick - Meu Dicionário." <https://www.meudicionario.org/joystick>. Acessado em 17 jan.. 2024.



Figura 9 - União e Tecnologia: Estudantes Utilizando a Placa Makey-Makey em Conjunto para Jogar Labirinto, Uma Experiência de Aprendizado Colaborativo Onde a Conexão Física Entre os Alunos Completa o Circuito do Joystick.

Fonte: acervo do autor

Durante a atividade, foi observado que os alunos já possuíam conhecimento prévio sobre a ferramenta Scratch. Além disso, os estudantes que enfrentaram dificuldades foram auxiliados pelos colegas de grupo, evidenciando uma colaboração entre os alunos.

O segundo encontro foi realizado no dia 30 de março de 2023, no Laboratório de Matemática do Campus. Como no primeiro encontro, este também foi dividido em duas partes. No início do encontro, os alunos receberam o jogo "Pássaro" para lerem e interpretarem o código-fonte. Em seguida, cada grupo composto por 4 alunos teve a tarefa de explicar trechos específicos do código do jogo.

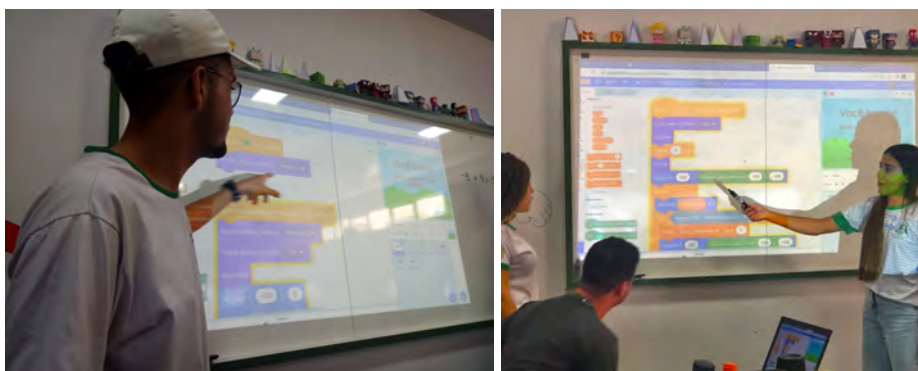


Figura 10 - Estudantes explicando seções do código do jogo Pássaro.

Fonte: acervo do autor

Na segunda parte do encontro, foi realizada uma explicação matemática sobre o funcionamento de uma função exponencial, com a função sendo escrita no quadro e representada em um plano cartesiano. Após a explicação, os alunos tiveram a oportunidade de experimentar a função exponencial em um exercício prático. Cada grupo recebeu um conjunto

de dados para analisar e, em seguida, foi solicitado que eles criassem um gráfico da função exponencial que melhor representasse esses dados utilizando o *software* GeoGebra. Os alunos trabalharam em conjunto para calcular os valores necessários e criaram gráficos precisos e interessantes.

Ao final do encontro, o código do jogo "Paraquedas" foi apresentado, e um dos integrantes do projeto explicou como a função exponencial pode ser aplicada para criar o movimento de curva de um avião dentro do jogo. A experiência permitiu que os alunos aplicassem os conceitos matemáticos aprendidos na prática, com a ajuda da programação.

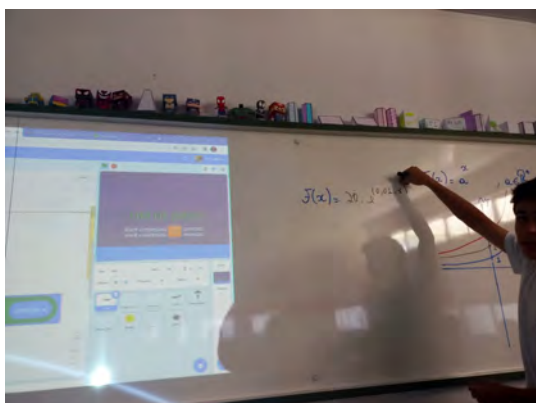


Figura 11 - Estudante explicando a função exponencial do jogo Paraquedas.
Fonte: acervo do autor

O terceiro encontro foi realizado em sala de aula no dia 13 de abril de 2023. Neste encontro, a turma foi dividida em 3 grupos, com o objetivo de reunir estudantes com diferentes habilidades artísticas, matemáticas e lógicas, visando a criação de 3 jogos e 3 dispositivos robóticos distintos que seriam apresentados no Hospital Dia do Idoso de Anápolis-GO. O professor propôs a utilização da ferramenta Scratch para a construção dos jogos e deixou a critério dos alunos a escolha dos materiais para a construção dos dispositivos robóticos.

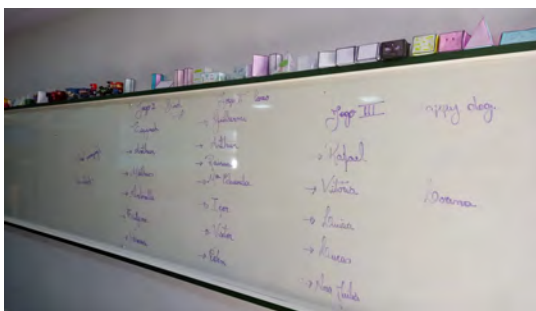


Figura 12 - Divisão dos grupos responsáveis pelos jogos.
Fonte: acervo do autor

Na segunda parte do encontro, os alunos participaram de uma videoconferência com a fisioterapeuta responsável pelo projeto no Hospital Dia do Idoso, promovendo um bate-papo enriquecedor.



Figura 13 - Sessão de Videoconferência Interativa: Alunos em Diálogo com a Fisioterapeuta Bruna Cândido.
Fonte: acervo do autor

No quarto encontro, que ocorreu em 27 de abril, os estudantes, já organizados em grupos preestabelecidos, tiveram a oportunidade de apresentar suas propostas iniciais de jogos a serem desenvolvidos. Os títulos sugeridos para os jogos foram "Bird in the Sky"¹⁴, "Flower"¹⁵ e "Way to the Garden"¹⁶.

No quinto encontro, realizado no dia 19 maio, e sexto encontro, realizado no dia 15 de junho, os alunos começaram a projetar e codificar os jogos. Nesses encontros os dispositivos robóticos também começaram a ser desenvolvidos. Os estudantes que tinham dificuldade com a plataforma de programação, foram auxiliados pelos colegas e pelo professor.

No sétimo encontro, em 22 de junho, os grupos de estudantes encararam o desafio de exibir os protótipos dos jogos, bem como os *joysticks* que foram desenvolvidos para acompanhá-los. Durante essa apresentação, receberam *feedback* construtivo por parte de colegas pertencentes a outros grupos.

Para o jogo "Bird in the Sky", por exemplo, uma abordagem inovadora foi adotada. Foi concebido um *joystick* com o formato de uma asa, que permite ao jogador imitar o movimento de bater asas ao erguer a asa do *joystick*. Tal gesto se traduz no jogo pelo movimento ascendente do personagem, um pássaro, como se estivesse batendo asas para ganhar altitude.

¹⁴ Bird In The Sky - Disponível em: <<https://scratch.mit.edu/projects/890745397/editor/>>.

¹⁵ Flower - Disponível em: <<https://scratch.mit.edu/projects/881032151/editor/>>.

¹⁶ Way To The Garden - Disponível em: <<https://scratch.mit.edu/projects/887979215/editor/>>.



Figura 14 - Estudantes apresentando o dispositivo robótico “Asa”.
Fonte: acervo do autor

Um segundo joystick utilizado foi o leme de navio, confeccionado a partir de materiais recicláveis. Esse dispositivo foi construído utilizando um aro de bicicleta, partes de um palete e segmentos de cabo de vassoura. O seu propósito é singular: proporcionar o controle preciso da direção da embarcação virtual dentro do jogo "Navegação". Futuramente o mesmo seria adaptado para o jogo Way to the Garden e seria chamado de “Volante”. Um dispositivo chamado regador, criado a partir de um regador de plantas, foi confeccionado para o jogo Flower.



Figura 15 - Estudantes apresentando os dispositivos robóticos “Leme” e “Regador”.
Fonte: acervo do autor

No oitavo e nono encontro os grupos de estudantes se focaram em concluir os jogos digitais e projetos robóticos para que pudessem ser apresentados as versões finais destes no próximo encontro.

O grupo encarregado do desenvolvimento do jogo "Bird In the Sky" decidiu abandonar a ideia de utilizar as asas de pássaro como um dispositivo robótico, optando, em vez disso, por incorporar um dispositivo semelhante a um guidão de bicicleta confeccionado com um tripé de câmera. Simultaneamente, o grupo responsável pelo jogo "Way to the

"Garden" não apenas testava e pintava o dispositivo robótico "Volante" para a condução do carro no jogo, mas também realizava os ajustes finais no código-fonte do projeto. Paralelamente, o grupo envolvido no jogo "Flower" também estava concentrado nos ajustes finais do dispositivo robótico "regador" e no refinamento do código fonte do jogo. Ambos os grupos demonstraram um notável comprometimento na conclusão bem-sucedida de seus projetos.



Figura 16 - Estudantes refinando e testando os jogos e dispositivos robóticos.
Fonte: acervo do autor

O décimo e último encontro teve lugar nas instalações do Hospital Dia do Idoso em Anápolis, Goiás. Nessa ocasião, os estudantes tiveram a oportunidade de colocar em prática seus jogos e dispositivos robóticos. Durante a manhã, os três jogos, "Bird in the Sky", "Flower" e "Way to the Garden", foram apresentados aos idosos, que tiveram a chance de jogá-los sob a supervisão atenta de fisioterapeutas. A atividade promoveu uma participação colaborativa entre estudantes, professor, profissionais da saúde, idosos e seus familiares, além do pesquisador.



Figura 17 - Seção de fisioterapia no Hospital Dia do Idoso.
Fonte: acervo do autor

3.5 ORGANIZAÇÃO DOS DADOS DA PESQUISA

Como apresentado na introdução desta pesquisa, o Pensamento Computacional foi selecionado como tema desta pesquisa devido a sua importância no desenvolvimento de habilidades para os estudantes do século XXI. A coleta e a análise de dados foi feita baseada nos métodos qualitativos expostos por Gil (2002) além de Bogdan e Biklen (1994).

A organização dos dados desta pesquisa se deu em 3 etapas, sendo elas Coleta, Análise e Triangulação dos Dados. Na fase de coleta de dados produzidos e organizados ao longo das etapas de criação dos jogos, foram utilizados 5 instrumentos, sendo eles: caderno de campo, registros fotográficos dos encontros, prints dos códigos fonte dos jogos, observação em sala de aula e questionário aberto aplicados via Google Forms. As descrições das

dimensões e habilidades do Pensamento Computacional foram incluídas no questionário para orientar os estudantes ao responderem as perguntas.

A etapa de análise de dados foi feita durante a pesquisa e de forma interpretativa com base nas observações do pesquisador, uma vez que se trata de uma pesquisa qualitativa, conforme Gil (2002).

Em sala de aula, observamos como o professor introduziu os conceitos de lógica de programação, como ele propôs os desafios a serem resolvidos pelos estudantes, o comportamento destes ao se depararem com os desafios, a interação entre eles, as ferramentas utilizadas, além de comportamentos e eventos significativos inesperados que puderam ocorrer durante todo o processo. Todos os eventos foram anotados em um diário/nota de campo. “O resultado bem sucedido de um estudo de observação participante em particular, mas também de outras formas de investigação qualitativa, baseia-se em notas de campo detalhadas, precisas e extensivas” (Bogdan e Biklen, 1994, p. 150).

Através de prints de trechos de códigos dos jogos elaborados pelos estudantes, questionários realizados após o desenvolvimento dos jogos e dispositivos robóticos, e observação atenta da concepção, planejamento e criação dos jogos, foram identificados conceitos, práticas e perspectivas do Pensamento Computacional.

Adotamos a precaução de identificar significados nos dados, evitando redundâncias de informações. Inicialmente, por meio de uma pré-análise abrangente, selecionamos e direcionamos nossa atenção exclusivamente para as partes que contribuíssem para o propósito central desta pesquisa. Em seguida, organizamos e estruturamos a compreensão do Processo de Desenvolvimento do Pensamento Computacional utilizando descrições, quadros e interpretações analíticas dos dados identificados.

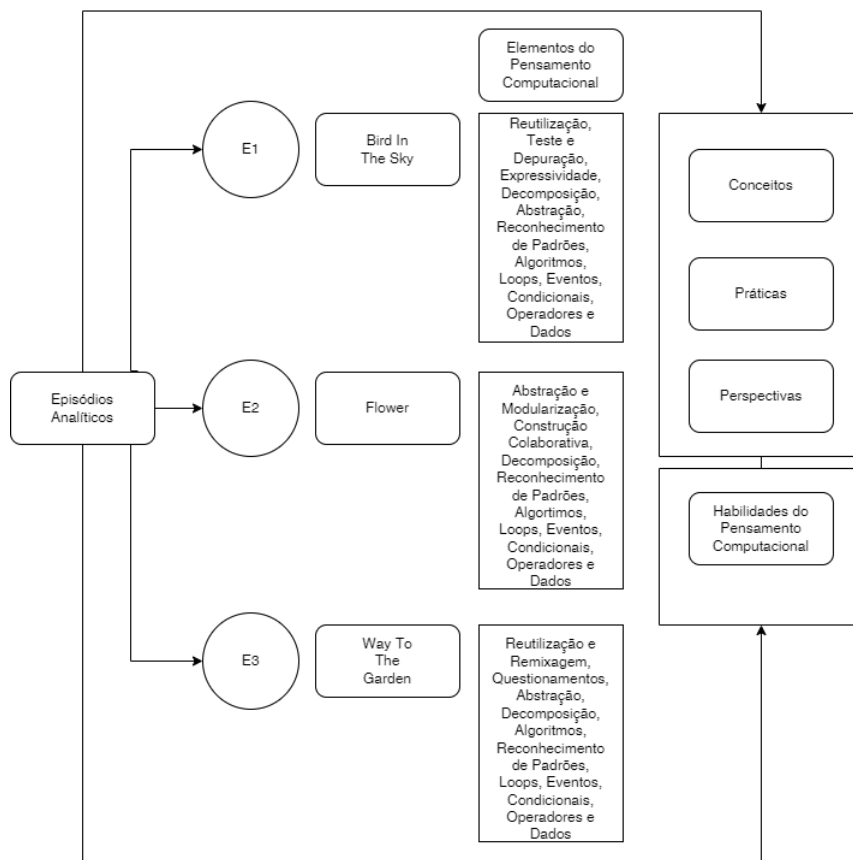
Denominamos o procedimento de elaboração de cada jogo digital como "Episódios Analíticos" (En). Através desses episódios, derivamos as "Categorias de Análise" (Cn) deste estudo e, conseqüentemente, os resultados da pesquisa. Toda a investigação segue a orientação da pergunta principal desta dissertação: "Quais são as implicações da construção de jogos digitais e dispositivos robóticos para o desenvolvimento do pensamento computacional de estudantes do ensino médio?".

Após essas etapas, fizemos a triangulação dos dados coletados. Patton (1999) cita a Triangulação de dados como uma abordagem metodológica que envolve a utilização de múltiplos métodos, fontes ou perspectivas na coleta e análise de dados, a fim de validar e enriquecer as conclusões de uma pesquisa. Essa estratégia é empregada para aumentar a confiabilidade e a validade dos resultados, proporcionando uma visão mais abrangente e

robusta do fenômeno estudado. Assim, a triangulação consistiu do uso de elementos extraídos dos trechos de códigos dos jogos, questionários e fotografias dos encontros.

3.5.1 Episódios Analíticos e Categorias de Análise

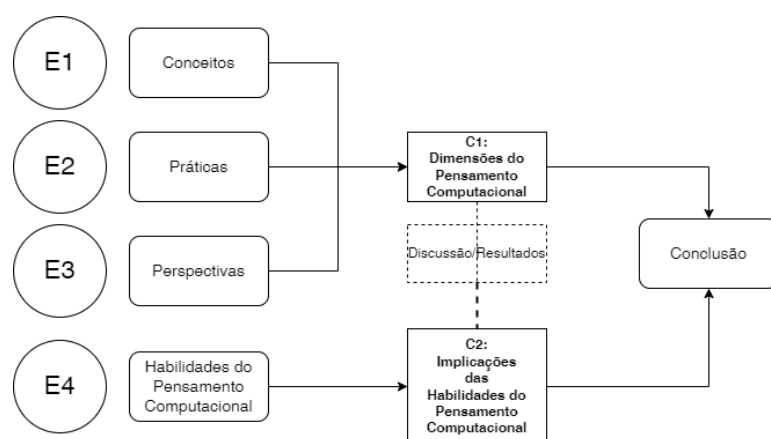
Apresentamos os resultados de nossa pesquisa, destacando a obtenção de três Episódios Analíticos (En), designados como E1, E2 e E3. Juntos, esses episódios compõem uma abrangente representação de todos os dados produzidos e organizados ao longo do estudo. Cada episódio recebeu o nome correspondente à sua respectiva inovação científico-tecnológica, sendo estas o jogo digital Bird In The Sky associado ao dispositivo robótico (E1), a criação Flower vinculada ao Regador-Robótico (E2) e a concepção Way To The Garden integrada ao volante (E3). Apresentamos, a seguir, um Infográfico com a síntese desses En.



Infográfico 1 - Visão Geral dos Episódios Analíticos.
Fonte: elaborado pelo autor

O infográfico apresenta de forma clara e concisa a estrutura dos Episódios Analíticos, destacando os elementos-chave do pensamento computacional empregados no desenvolvimento das três inovações científico-tecnológicas discutidas na dissertação. Cada um dos episódios, E1, E2 e E3, corresponde a um projeto distinto – o jogo "Bird In The Sky", a criação "Flower" e o jogo "Way To The Garden" – e demonstra a aplicação prática de habilidades e conceitos essenciais ao pensamento computacional. Essas habilidades incluem reutilização, teste, depuração, expressividade, decomposição, abstração, reconhecimento de padrões, desenvolvimento e implementação de algoritmos, loops, eventos, condicionais, operadores e manipulação de dados.

As Categorias de Análise que emergidas a partir dos Episódios Analíticos são fundamentais para uma compreensão mais profunda dos processos envolvidos no desenvolvimento do pensamento computacional dentro do projeto. As categorias identificadas são: C1: Dimensões do Pensamento Computacional e C2: Implicações das Habilidades do Pensamento Computacional. As categorias podem ser visualizadas, a seguir:



Infográfico 2 - Categorias de Análise.

Fonte: Elaborado pelo autor

O processo de construção das Cn se estabelece como um procedimento “[...] de organização de transcrições que foram sendo acumulados, com o objetivo de aumentar a sua própria compreensão desses mesmos materiais” (Bogdan, Biklen, 1994, p. 205). A categoria "C1 Dimensões do Pensamento Computacional" enfoca as dimensões essenciais do Pensamento Computacional, conforme delineadas por Resnick (2012). Esta categoria decorre três aspectos cruciais: Conceitos, Práticas e Perspectivas. Estes elementos são vitais para aprofundar nossa compreensão sobre como o pensamento computacional influencia e enriquece o processo educacional dos alunos envolvidos nos projetos E1, E2 e E3. Ao explorar esta categoria, buscamos não apenas avaliar o conhecimento técnico adquirido pelos

estudantes, mas também entender como suas práticas e perspectivas evoluem ao longo da experiência de aprendizagem.

A categoria " C2 Implicações das Habilidades do Pensamento Computacional" aborda as repercussões práticas e teóricas das habilidades desenvolvidas pelos alunos nos projetos E1, E2 e E3. Esta categoria é crucial para compreender como o pensamento computacional transcende o mero conhecimento técnico, influenciando o raciocínio, a resolução de problemas e a criatividade dos estudantes.

Ao investigar as implicações dessas habilidades, enfocamos como os alunos aplicam o pensamento computacional em contextos variados, indo além da sala de aula. Isso inclui a análise de como essas habilidades melhoram a capacidade dos alunos para enfrentar desafios complexos, colaborar com eficiência e inovar em suas abordagens. Após compreender a estrutura das Cn, progredimos para o capítulo de análise na dissertação.

4 APRESENTAÇÃO E ANÁLISE QUALITATIVA DOS DADOS

Neste capítulo, apresentamos uma análise qualitativa dos dados coletados no contexto do projeto de extensão do IFGoiano. Organizamos e estruturamos a compreensão do Processo de Desenvolvimento do Pensamento Computacional através de descrições, tabulações e interpretações analíticas dos dados obtidos. Em outras palavras, buscamos harmonizar esses elementos e apresentá-los de maneira contextualizada. A integração desses diversos registros, sob uma abordagem qualitativa, pode ser percebida como uma forma de triangulação dos dados da pesquisa.

Para facilitar o entendimento, os fragmentos de código apresentados nas seções subsequentes serão mostrados tanto em blocos distintos quanto em formato textual descritivo. Essa conversão do código para a forma narrativa foi realizada utilizando a plataforma Rokcoders¹⁷.

4.1 E1 - BIRD IN THE SKY

Neste segmento da dissertação, apresentamos o Jogo do Bird in The Sky, um projeto dividido em quatro fases essenciais: "Pré-produção", "Confecção dos artefatos robóticos e códigos", "Discussão e apresentação" e "Jogo em prática". Durante o processo de elaboração dos jogos, diversas dimensões e habilidades do pensamento computacional foram identificadas, as quais serão abordadas com mais detalhes ao longo deste episódio.

¹⁷ "sb3-to-txt - RokCoder Scratch Projects." <https://www.rokcoder.com/sb3-to-txt/>. Acessado em 18 jan.. 2024.

4.1.1 E1: Etapas De Desenvolvimento


Pré-produção	Confeção dos artefatos robóticos e códigos
	
Discussão e apresentação	Jogo em prática
	

Ilustração 1: Iterações Multifásicas: Uma Análise Visual do Desenvolvimento do 'Bird in The Sky' (dados da pesquisa).

Fonte: Acervo do autor

Quanto às representações visuais (Ilustração 1), a primeira imagem, à esquerda na ilustração, exemplifica a dinâmica de argumentação, expressão e reflexão-discussão compartilhada entre os estudantes. O conhecimento “torna-se valorizado pelo próprio aluno por ser útil, por ser possível compartilhar com outras pessoas e por combinar com o estilo pessoal de cada indivíduo” (Papert, 2008, p. 173). Estes estão envolvidos em debates construtivos, desenvolvendo ideias e refinando conceitos para a concepção do jogo do pássaro. A segunda imagem oferece um vislumbre do processo criativo relacionado ao desenvolvimento do dispositivo robótico destinado ao jogo. Esta fase captura a essência da inovação, mostrando a materialização das ideias em um artefato palpável, unindo a criatividade com a prática técnica. A terceira imagem retrata o momento em que os alunos apresentam o jogo aos seus colegas. Durante essa exposição, eles elucidam segmentos cruciais do código-fonte, fornecendo uma visão detalhada do funcionamento interno do jogo. Simultaneamente, o grupo participa de discussões sobre possíveis aprimoramentos,

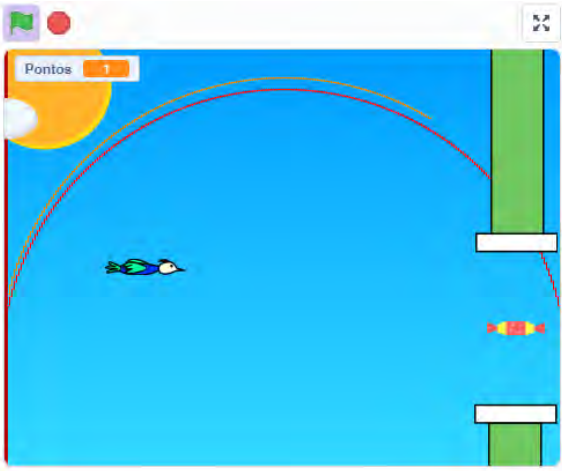
evidenciando um ambiente de colaboração e refinamento contínuo. Na quarta representação, é apresentada a implementação do jogo Bird in the Sky no Hospital do Idoso. Nesta fase, a validação da eficácia do jogo foi acompanhada por uma valiosa interação e feedback prático com os usuários reais.

Este ciclo de apresentação e diálogo não apenas promove a compreensão mútua entre os estudantes, mas também estimula a busca constante por melhorias e inovações no projeto.

4.1.2 E1: Dimensões Do Pensamento Computacional

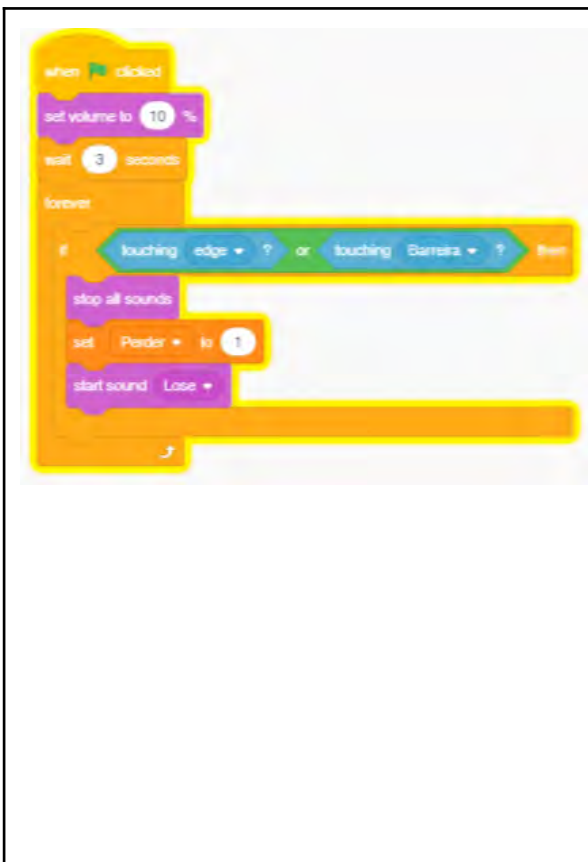
Com o objetivo de elucidar as dimensões do pensamento computacional empregadas no desenvolvimento do jogo 'Bird In The Sky', apresentamos a seguir um quadro detalhado. Este quadro descreve o funcionamento do jogo e inclui um segmento selecionado do seu código-fonte, proporcionando uma base sólida para nossa análise aprofundada:

Quadro 7 - Analisando o jogo Bird in the Sky (Dados da pesquisa) ‘continua’.

Leiaute e algoritmos	Descrição do jogo e do código
	<p>O jogo apresenta um pássaro que, o jogador controla por meio de um dispositivo robótico que utiliza Microbit. O objetivo é evitar colidir com barreiras verdes (objetos chamados "Barreira") enquanto mantém o pássaro em movimento. O jogo envolve elementos como gravidade, colisões, pontos e balas. Certas ações, como tocar uma barreira verde, podem afetar o jogo de maneiras específicas.</p>

Fonte: elaborado pelo autor

Quadro 7. Continuação



Este trecho do código do jogo do Bird in The Sky, referente ao objeto pássaro, lida com eventos que ocorrem quando a bandeira verde (iniciar) é clicada.

- O código ajusta o volume do som para 10. Isso pode ser feito para garantir uma experiência de áudio adequada no início do jogo.
- Após a configuração do volume, o jogo espera por 3 segundos antes de entrar em um loop. Esse intervalo pode ser utilizado para dar ao jogador um momento de preparação antes do início efetivo do jogo.
- A entrada em um loop infinito sugere que este bloco de código será executado continuamente durante todo o tempo de execução do jogo.
- Verifica constantemente se o pássaro está tocando nas bordas da tela ou em um objeto chamado "Barreira". Essa verificação é fundamental para detectar colisões no jogo.
- Se a condição de colisão for verdadeira, o código executa ações como parar todos os sons, sinalizar uma condição de perda (atribuindo 1 à variável "Perder") e reproduzir um som chamado "Lose". Essas ações indicam que o jogador perdeu o jogo.

Fonte: elaborado pelo autor

Indubitavelmente, todos os estudantes do projeto Mattics que participaram da codificação dos jogos digitais, em algum momento, empregaram os conceitos da dimensão do pensamento computacional. Isso se torna evidente no momento em que o professor opta por incorporar a plataforma Scratch como uma ferramenta para o desenvolvimento de jogos, o que automaticamente implica na aplicação de conceitos fundamentais, tais como loops, condicionais, operadores, entre outros. À medida que os jovens projetam mídias interativas com o Scratch, eles se envolvem com um conjunto de conceitos computacionais (mapeamento para blocos de programação do Scratch) que são comuns em muitas linguagens de programação (Resnick, 2012 ,p. 2, tradução nossa).

Ao analisar o trecho de código do jogo Bird In the Sky, podemos identificar vários desses conceitos fundamentais de programação (Quadro 8), alinhando-se ao desenvolvimento do Pensamento Computacional proposto por Resnick (2012):

Quadro 8 - Conceitos do PC identificados no trecho de código de Bird In The Sky

Pensamento Computacional: Conceitos segundo Resnick (2012)	Descrição
Sequência	O código inicia ajustando o volume do som (Sound.SetVolumeTo(10)), aguarda por 3 segundos (Control.Wait(3)) e, em seguida, entra em um loop infinito (Forever).
Loops (Forever)	A estrutura Forever cria um loop contínuo que monitora constantemente as condições do jogo. Isso demonstra a habilidade de pensar em termos de iterações repetitivas.
Eventos	O evento "Green Flag" é acionado quando a bandeira verde é clicada, iniciando o código. Isso ilustra a capacidade de responder a eventos específicos durante a execução do programa.
Condicionais	O bloco condicional If verifica se o objeto do jogo está tocando as bordas (<code>_edge_</code>) ou uma barreira (Barreira). Se essa condição for verdadeira, ações específicas são desencadeadas.
Operadores	O operador lógico Or é utilizado na condição, permitindo que o código responda caso uma das condições (toque nas bordas ou na barreira) seja verdadeira.
Dados (Variável "Perder")	A variável "Perder" é utilizada para controlar o estado do jogo. Se assumir o valor 1, indica que o jogador perdeu. Isso demonstra a manipulação de dados para controlar o fluxo do jogo.

Fonte: elaborado pelo autor

A utilização dos conceitos do pensamento computacional também puderam ser percebidas por meio das entrevistas realizadas com os estudantes do grupo responsável pelo desenvolvimento do jogo Bird In The Sky:

Pesquisador: “Poderia destacar uma funcionalidade específica do seu jogo em que você aplicou com sucesso algum conceito de loops, eventos, sequências, paralelismo, condicionais, operadores lógicos ou dados?”

Antonella: “Utilizamos condicionais no movimento do pássaro, tinha que ficar verificando o tempo todo se o pássaro bateu nos objetos”.

Esses elementos evidenciam a aplicação do Pensamento Computacional, conforme delineado por Resnick (2012), ao desenvolver habilidades de sequência, iteração, resposta a eventos, raciocínio condicional, uso de operadores lógicos e manipulação de dados para criar um jogo interativo e funcional. O código reflete a capacidade do programador de pensar de forma estruturada e lógica para resolver problemas específicos no contexto do jogo.

Por meio das observações em sala, o notamos que, no âmbito da aplicação do pensamento computacional voltado para a reutilização, durante a elaboração do jogo, identificamos a prática de reutilizar recursos da plataforma Scratch. Um exemplo concreto desse procedimento foi a utilização de áudios provenientes da plataforma (Figura 18).



Figura 18 - Espectrograma de uma das músicas da biblioteca Scratch utilizada como tema do jogo Bird In The Sky (dados da pesquisa).

Fonte: < <https://scratch.mit.edu/projects/890745397/editor/>>. Acesso em 10 de novembro de 2023

Quando questionados se reutilizaram códigos no jogo, a estudante Antonella respondeu:

Antonella: Sim. Por exemplo: as ondas do jogo Bird in the sky foram baseadas em um código de funções trigonométricas já existente.

A citação de Antonella sobre a utilização de funções trigonométricas existentes para criar as ondas no jogo Bird in the Sky destaca ainda mais a prática de reutilização de código. Isso indica que os alunos não apenas reutilizaram elementos visuais ou sonoros, mas também aproveitaram soluções de programação já existentes para atender às necessidades do jogo que estavam desenvolvendo. Essa abordagem não apenas agiliza o processo de desenvolvimento, mas também promove uma compreensão mais profunda dos conceitos de programação.

A prática de testar e depurar revela-se como um conceito significativo no desenvolvimento do pensamento computacional. Durante uma entrevista com a estudante Antonella, o pesquisador questionou sobre desafios enfrentados no desenvolvimento de jogos ou dispositivos robóticos, buscando entender como ela abordou tais questões.

Pesquisador: “Pode compartilhar um exemplo específico de um problema que você enfrentou durante o desenvolvimento de um jogo/dispositivo robótico e como lidou com isso?”

Antonella: “Foi um desafio desenvolver o código para o movimento adequado do pássaro. Busquei identificar os possíveis erros, para assim pensar em alguma solução para esses erros, baseada em técnicas aplicadas em outros jogos.”

Antonella enfatiza a identificação de erros no código, a aplicação de técnicas de outros jogos para encontrar soluções e destaca a atitude pró-ativa na resolução de problemas. A narrativa ressalta a natureza iterativa e experimental do processo de teste e depuração, destacando como esses elementos são cruciais para o desenvolvimento de habilidades essenciais no campo da computação.

Para enriquecer nossa análise acerca da prática de testes e depuração, ao examinar o trecho de código fonte na Figura 19, torna-se evidente que determinadas seções foram inicialmente desenvolvidas, submetidas a testes e, subsequentemente, retrabalhadas.

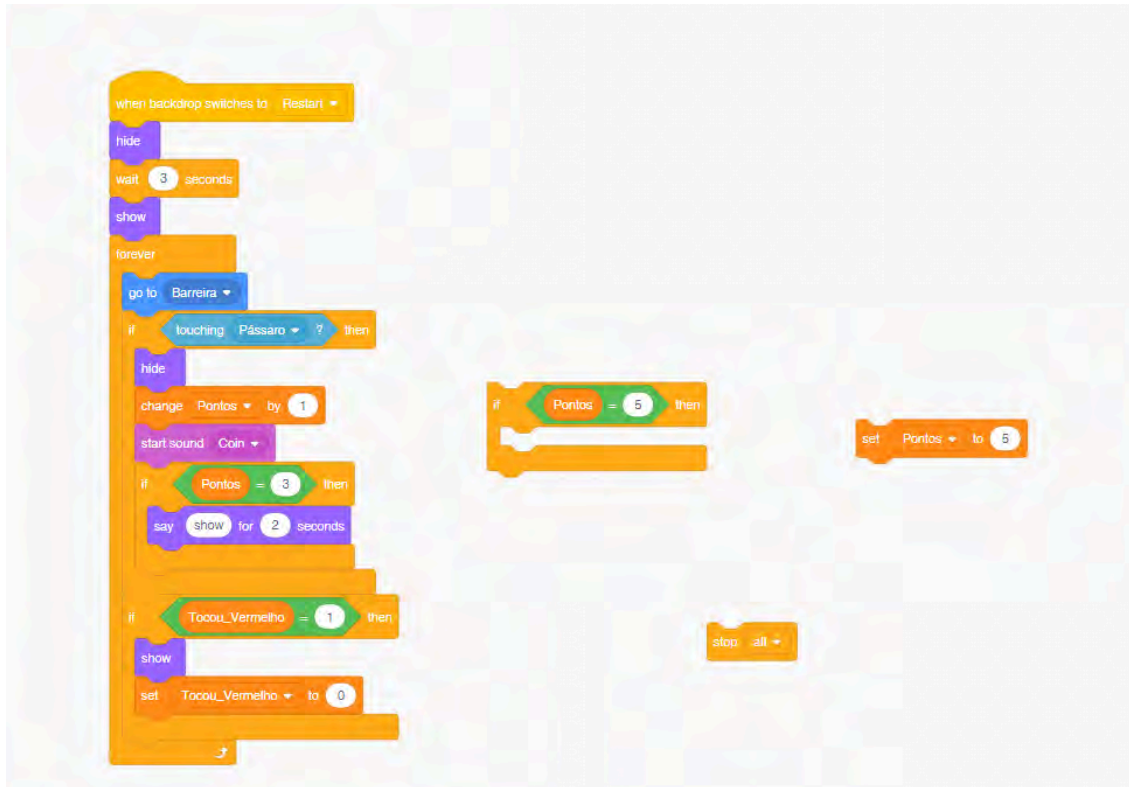


Figura 19 - Trecho de código do jogo Bird In The Sky (dados da pesquisa).

Fonte: <https://scratch.mit.edu/projects/890745397/editor/>. Acesso em 12 de novembro de 2023.

O segmento de código acima corresponde ao sistema de captação de "balas" do jogo. Quando o pássaro entra em contato com a bala (if (touching (Pássaro))), um ponto é incrementado (change (pontos) by 1), um som de moeda é reproduzido (start sound (Coin)), e se o jogador atingir 3 pontos (if (Pontos) = 3), a mensagem de feedback "show" é exibida por dois segundos (say "show" for 2 seconds).

No trecho de código no centro da figura, que não é inicializável, notamos que os estudantes inicialmente atribuíram o valor 5 a essa condição. No entanto, após testes, chegaram à conclusão de que o valor mais apropriado para o jogo seria 3. Nas observações do pesquisador, ficou evidente que esse tipo de parâmetro era constantemente testado e ajustado durante a elaboração do código.

A importância de conduzir testes e depuração durante o desenvolvimento de *software* é amplamente reconhecida na literatura especializada. Essas práticas desempenham um papel fundamental na identificação e correção de potenciais erros, contribuindo para a confiabilidade e eficiência do sistema (Myers, 2011). No caso específico do trecho de código

em questão, a transição do valor inicial de 5 para 3 revela a necessidade de ajustes com base na experiência prática.

Ao testar diferentes valores e condições, os estudantes puderam avaliar o impacto dessas escolhas no desempenho do jogo. No exemplo mencionado, a mudança para o valor 3 pode ter sido motivada por uma melhor adequação à dinâmica do jogo ou à preferência do jogador. Essa iteração contínua é fundamental para otimizar o código e proporcionar uma experiência mais sólida aos usuários.

Além disso, a depuração, que envolve a identificação e correção de erros no código, desempenha um papel crucial na eliminação de possíveis falhas que podem comprometer o funcionamento do sistema (Barnes, 2010). O fato de os estudantes terem percebido a necessidade de ajustar o valor do parâmetro indica uma abordagem proativa na resolução de problemas, contribuindo para um código mais robusto e confiável.

Ainda no que se refere a testes e depuração, durante a visita ao Hospital Dia do Idoso, ao iniciar a partida de Bird in the Sky, os alunos identificaram que a velocidade de ascensão e descida do pássaro não estava ideal. Em resposta a essa observação, eles realizaram ajustes imediatos nos parâmetros, buscando otimizar a jogabilidade do game.

Durante os testes finais do dispositivo robótico Asa, observados em sala de aula, o pesquisador constatou que os estudantes reconheceram a inadequação desse dispositivo para a sua utilização. Como resposta a essa percepção, os alunos optaram por desenvolver outro dispositivo robótico, utilizando uma placa BBC microbit e um tripé de câmera fotográfica, conforme ilustrado na Figura 20.



Figura 20 - Dispositivo robótico adaptado (dados da pesquisa).

Fonte: acervo do autor

Em resumo, a prática regular de testar e depurar não apenas aprimora a qualidade do código, mas também promove a eficiência no desenvolvimento, permitindo que os desenvolvedores respondam de forma ágil a desafios e refinem continuamente suas soluções.

Outra dimensão crucial para avaliar o progresso do pensamento computacional, conforme evidenciado nesta análise de projeto, é a perspectiva denominada "Expressando". Nessa abordagem, o estudante não se percebe apenas como um mero usuário de tecnologia, mas sim como um construtor ativo da mesma.

Pesquisador: Comparando a criação de jogos no Scratch com outras atividades online, como redes sociais, o que diferencia a experiência para você?

Igor: A proposta de criar algo do 0, já que nas redes sociais, normalmente é utilizado algo pronto e rápido. Na criação de jogos existe a probabilidade de usar sua criatividade desde o início, meio e fim do projeto.

A criatividade, mencionada por Igor, é fundamental para o desenvolvimento e a originalidade de projetos no âmbito do pensamento computacional. Essa habilidade criativa não apenas estimula a inovação, mas também desafia o estudante a pensar de forma abrangente, enfrentando os desafios do processo de criação de jogos. Ao criar algo do zero, o indivíduo não apenas aprende a aplicar conceitos técnicos, mas também desenvolve a capacidade de resolver problemas de maneira única e adaptativa.

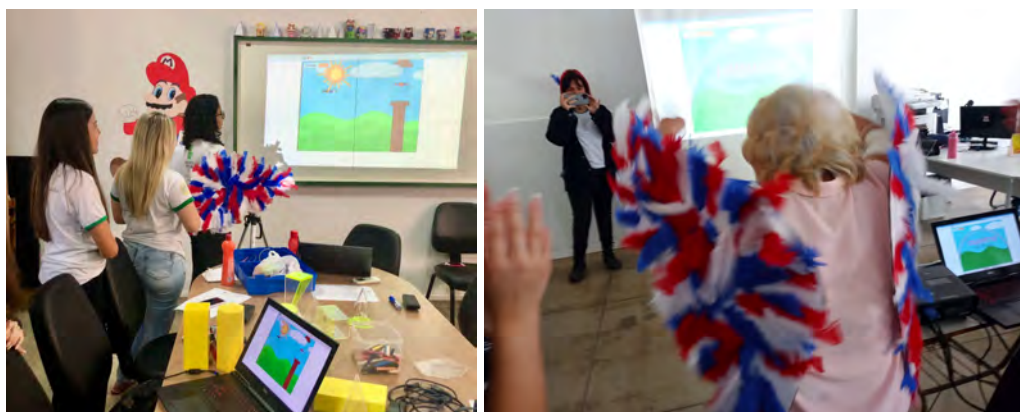


Figura 21 - Uso prático do dispositivo robótico Asa (dados da pesquisa).

Fonte: acervo do autor

Na Figura 21, visualiza-se o processo de desenvolvimento de um dos dispositivos robóticos utilizados no jogo "Bird In The Sky". Dado que o personagem principal é um pássaro, as estudantes Julia e Lorena empreenderam a tarefa de simular o movimento de voo da ave por meio deste dispositivo robótico, manifestando suas ideias por meio de uma abordagem inovadora. Na imagem à extrema direita, as estudantes puderam constatar que suas criações foram efetivamente aplicadas em um cenário prático. Uma das pacientes do

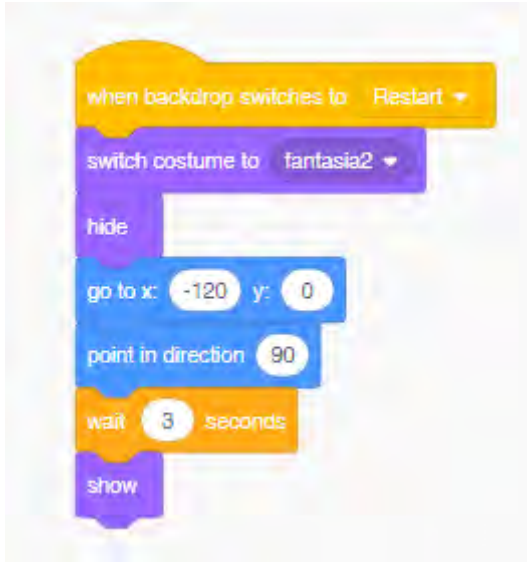
Hospital Dia do Idoso teve a oportunidade de utilizar o movimento de bater asas, desenvolvido pelas estudantes, como parte do tratamento para o mal de Parkinson. Esse impacto real evidencia a relevância e a aplicabilidade prática das soluções desenvolvidas no contexto do projeto.

A concepção desses algoritmos foi fundamentada em uma abordagem integrada do conhecimento. À medida que os estudantes se dedicaram a essa empreitada, estabeleceram conexões com conceitos previamente estudados e empenharam-se no desenvolvimento articulado de princípios do Pensamento Computacional.

4.1.3 E1: Habilidades Do Pensamento Computacional

No Quadro 9 detalhamos alguns trechos de código do jogo a fim de evidenciar habilidades do pensamento computacional percebidas no jogo Bird in The Sky.

Quadro 9 - Trechos de código do jogo Bird In The Sky e suas funcionalidades (dados da pesquisa) ‘continua’.

Trecho de código Scratch	Funcionalidade
	<p>Configuração Inicial do Cenário:</p> <ul style="list-style-type: none"> • Configura o cenário, posição e a aparência inicial do personagem quando o cenário é trocado para "Restart". • Oculta detalhes irrelevantes durante a configuração inicial.

Quadro 9 - Continuação.

	<p>Loop Principal:</p> <ul style="list-style-type: none">• Dentro de um loop infinito, controla a física do jogo enquanto o jogador não perde.• Mantém a gravidade e a movimentação vertical quando o jogador não perde.
	<p>Reset de Variáveis:</p> <ul style="list-style-type: none">• Reinicializa várias variáveis do jogo para seus valores iniciais.• Esconde detalhes de reinicialização desnecessários.

Fonte: elaborado pelo autor

Os trechos de código tratam da configuração inicial do jogo quando o cenário é trocado para "Restart". Ele posiciona o personagem, configura a aparência, inicia a física do jogo e reinicia variáveis essenciais. O loop principal controla a física do jogo, garantindo que a gravidade afete o personagem apenas enquanto o jogador não perdeu.

O código em questão demonstra uma abordagem de programação cuidadosamente estruturada, destacando-se por sua decomposição eficiente. Ao analisar as partes específicas dedicadas à reinicialização do jogo, observamos uma clara divisão de responsabilidades, abrangendo desde a configuração inicial do cenário até a gestão da física do jogo e a reinicialização de variáveis. Esse método de decomposição não apenas facilita a manutenção do código, mas também promove uma compreensão mais clara das diferentes facetas

envolvidas na reinicialização do jogo. Além disso, o reconhecimento de padrões é evidenciado pelo uso do bloco Forever, indicando um padrão recorrente no jogo, onde as instruções contidas nele são executadas repetidamente enquanto o jogo está em andamento. A abstração, por sua vez, destaca-se na ocultação de detalhes desnecessários, simplificando a leitura e compreensão do código ao esconder aspectos como a lógica específica de mudança de trajes e a aparência do personagem. Os algoritmos implementados, especialmente na lógica de reinicialização, demonstram uma abordagem técnica sólida, controlando aspectos cruciais como a configuração do cenário, a física do jogo, a reinicialização de variáveis, a gravidade e a movimentação vertical. O Quadro 10 resume o que foi discutido.

Quadro 10 - Habilidades do pensamento computacional percebidas nos trechos de código

Categoria	Descrição
Decomposição	A própria maneira de construir os jogos baseada em atores, em que cada um tem seu próprio código, força a decomposição.
Reconhecimento de Padrões	Os blocos são intuitivos, de forma que operadores são encaixados em operandos, e testes condicionais encaixam com operandos. A medida que o aluno tenta encaixar os blocos, ele começa a reconhecer e aplicar os padrões observados.
Abstração	Detalhes desnecessários são ocultos. Por exemplo, a lógica específica de mudança de trajes, a aparência do personagem e a espera controlada pelo bloco Control.Wait são abstraídos para facilitar a leitura e compreensão do código.
Algoritmos	O código implementa algoritmos específicos para reiniciar o jogo, incluindo a configuração inicial do cenário, a física do jogo e a reinicialização de variáveis. Algoritmos também controlam a gravidade e a movimentação vertical.

Fonte: elaborado pelo autor

Para reforçar nossa análise a respeito das habilidades desenvolvidas apresentamos os seguintes apontamentos feitos por uma estudante durante as entrevistas:

Antonella: “A decomposição é fundamental principalmente no ajuste e na identificação de erros nos algoritmos dos jogos, uma vez que pequenas alterações modificam e atrapalham o funcionamento ideal dos jogos. Para o desenvolvimento da função trigonométrica (onda do jogo Bird in the sky) foram dias revendo e alterando códigos para que assim chegássemos no ideal. “E esse processo de decomposição foi muito importante”. “Os padrões facilitam o desenvolvimento do comando, a solução de erros e a construção dos jogos. No jogo Bird in the sky, por exemplo, as condições de movimento do pássaro foram construídas por meio de padrões”. “Primeiramente pensávamos em como aquilo precisa acontecer, para depois desenvolver o algoritmo”. “O nosso jogo teve um processo de abstração bem claro, pois pegamos uma ideia bem simples para que todos conseguissem entender o jogo”.

O depoimento de Antonella fornece uma valiosa perspectiva sobre a importância das habilidades desenvolvidas, destacando a relevância da decomposição na identificação e ajuste de erros nos algoritmos de jogos. Sua experiência ao desenvolver a função trigonométrica

para a onda do jogo "Bird in the sky" ressalta como pequenas alterações podem impactar significativamente o funcionamento ideal do jogo, evidenciando a necessidade de uma abordagem meticulosa. A ênfase dada aos padrões como facilitadores no desenvolvimento de comandos, na solução de erros e na construção dos jogos destaca a aplicação prática desse conceito. O exemplo específico das condições de movimento do pássaro, construídas por meio de padrões, ilustra como a identificação e aplicação consistente desses padrões podem simplificar o processo de desenvolvimento. Além disso, a abordagem de Antonella em relação à abstração é notável, ao mencionar a clareza no processo de abstração do jogo "Bird in the sky" para garantir compreensão universal. Esse depoimento reforça a ideia de que habilidades como decomposição, reconhecimento de padrões e abstração são fundamentais para enfrentar desafios complexos no desenvolvimento de jogos, promovendo eficiência, compreensão e resolução eficaz de problemas.

4.2 E2: FLOWER

Nesta seção, vamos analisar o desenvolvimento do Jogo Flower, também dividido em quatro fases: "Pré-produção", "Confecção dos artefatos robóticos e códigos", "Discussão e apresentação" e "Jogo em prática". Ao longo do processo de criação dos jogos, identificamos e evidenciamos as dimensões e habilidades do pensamento computacional, que serão discutidas em detalhes ao longo do episódio.

4.2.1 E2: Etapas De Desenvolvimento





Pré-produção	Confeção dos artefatos robóticos e códigos
	
Discussão e apresentação	Jogo em prática
	

Ilustração 3: Iterações Multifásicas: Uma Análise Visual do Desenvolvimento do Flower (dados da pesquisa).

Fonte: Acervo do autor

A análise da Ilustração 3 apresenta uma sequência clara do processo de desenvolvimento do jogo Flower. Na primeira imagem à esquerda, é evidenciada a fase crucial de idealização, onde as estudantes dedicam tempo à discussão e teste de elementos gráficos para o jogo. Esta etapa é fundamental para estabelecer as bases conceituais que guiarão o projeto. Na segunda imagem, destacamos a etapa de confecção dos artefatos robóticos e códigos. Notamos o engenhoso dispositivo robótico "regador", construído a partir de materiais reciclados do lixo, juntamente com uma placa BBC, demonstrando uma abordagem inovadora e sustentável. A terceira fase, representada na terceira imagem à esquerda, envolve a Discussão e Apresentação, onde o grupo compartilha o jogo com a turma

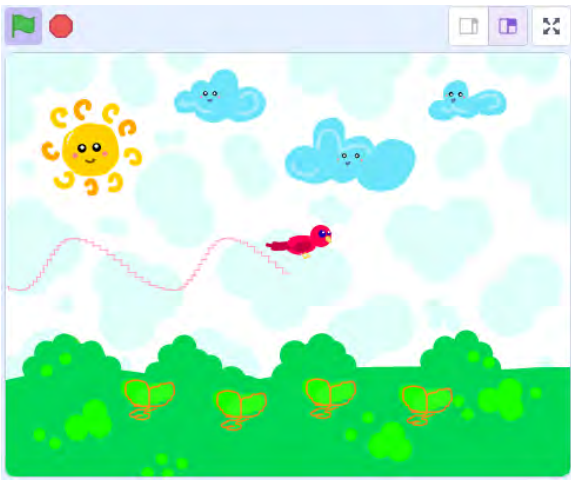
e promove uma discussão construtiva para identificar possíveis melhorias. Este processo colaborativo não apenas aprimora o projeto, mas também enriquece a experiência de aprendizado. Finalmente, na quarta e última etapa, denominada "Jogo em prática", as estudantes levam o jogo desenvolvido com o dispositivo robótico para uma sessão de fisioterapia no Hospital Dia do Idoso, envolvendo os idosos na experiência.

Assim, o processo de desenvolvimento do jogo Flower não apenas proporciona uma narrativa envolvente, mas também ilustra a integração sinérgica de criatividade, inovação, colaboração e aplicação prática, ressaltando a capacidade de projetos educacionais transcenderem as fronteiras da sala de aula e gerarem impacto significativo na comunidade e na vida cotidiana.

4.2.2 E2: Dimensões Do Pensamento Computacional


Com a finalidade de explorar as dimensões do pensamento computacional no desenvolvimento do 'Flower', apresentamos subsequente um quadro informativo (Quadro 11). Esse quadro ilustra os aspectos operacionais do jogo, complementada por uma seleção pertinente do código-fonte.

Quadro 11 - Analisando o jogo Flower (dados da pesquisa) ‘continua’.

Leiaute e algoritmos	Descrição do jogo e do código
	<p>O jogo envolve o crescimento de flores através da interação com a água do regador. Cada flor passa por diferentes estágios, representados por trajes e acompanhados por sons específicos. O objetivo é fazer com que todas as flores atinjam o estágio final para alcançar a vitória. A interação com eventos, troca de trajes, sons e movimentos contribui para a experiência do jogo. O código também inclui elementos de micro:bit, neste caso o micro:bit é usado no dispositivo robótico regador. O game possui uma celebração especial quando todas as flores atingem o estágio final.</p>

Quadro 11 - Continuação.

[BROTO]



```
when green flag clicked
  switch costume to semente
  go to x: -42 y: -153
  forever
    if <crecimento flor > 50 and <crecimento flor < 100 or <crecimento flor = 100 then
      go to x: -42 y: -153
      switch costume to fase2
      broadcast chuva
    if <crecimento flor > 100 and <crecimento flor < 200 or <crecimento flor = 200 then
      go to x: -42 y: -153
      switch costume to fase3
      broadcast chuva
    if <crecimento flor > 200 and <crecimento flor < 300 or <crecimento flor = 300 then
      go to x: -42 y: -153
      switch costume to fase4
      broadcast chuva
    if <crecimento flor > 300 and <crecimento flor < 400 or <crecimento flor = 400 then
      go to x: -42 y: -153
      switch costume to fase5
      broadcast chuva
    if <crecimento flor = 450 then
      go to x: -42 y: -153
      switch costume to fase final
      play sound brilho until done
      broadcast Ganhou
```

Este trecho de código representa a mecânica principal do jogo, onde o crescimento da flor é controlado pelo toque de "chuva". Conforme a flor cresce, ela passa por diferentes estágios, cada um associado a um traje específico. O jogo é vencido quando a flor atinge o estágio final. Analisando o código temos as seguintes instruções:

- Muda o traje para "semente" (Looks.SwitchCostumeTo(semente)).
- Move o objeto para as coordenadas (-42, -153) (Motion.GoToXY(-42, -153)).
- Inicia um loop infinito que continua durante toda a execução do jogo.
- Monitora o crescimento da flor, representado pela variável "crescimento flor".
- Se o crescimento da flor está entre 50 e 100 (inclusive), muda o traje para "fase2" e emite um broadcast chamado "chuva".
- Similarmente, há condições para outros estágios (fase3, fase4, fase5) com respectivos intervalos de crescimento.
- Se o crescimento da flor atinge 450, muda o traje para "fase final", reproduz um som ("brilho.wav") e emite um broadcast chamado "Ganhou".
- Fora do loop infinito, inicia outro bloco "WhenGreenFlagClicked", onde a variável "crescimento flor" é inicializada como 0.
- Dentro de um loop infinito, verifica se o objeto está tocando algo associado à "chuva" (Sensing.TouchingObject(chuva)) e, se sim, incrementa a variável "crescimento flor" em 1.

Fonte: elaborado pelo autor

Analisando esse trecho de código, destacamos no Quadro 12 os seguintes conceitos das Dimensões do Pensamento Computacional utilizados na construção do código:

Quadro 12 - Conceitos do PC identificados no trecho de código do jogo Flower.

Conceito	Utilização no Código	Exemplo
Sequência	Ordem de comandos para configuração inicial e comportamento do jogo	Instruções dentro do bloco WhenGreenFlagClicked
Loops	Uso de Forever para execução contínua e monitoramento do jogo	Forever usado para repetir a lógica principal do jogo
Eventos	WhenGreenFlagClicked para iniciar configurações iniciais	Configurações iniciais quando o botão verde é clicado
Condicionais	Estruturas If para verificar condições e tomar decisões	Verificação do crescimento da flor e mudança de estágio
Operadores	Uso de operadores lógicos como And e Or para criar condições	(crescimento flor > 50) And ((crescimento flor < 100) Or (crescimento flor == 100))
Dados	Utilização de variáveis (como crescimento flor) para armazenar informações essenciais no jogo	crescimento flor é uma variável que rastreia o crescimento da flor

Fonte: elaborado pelo autor


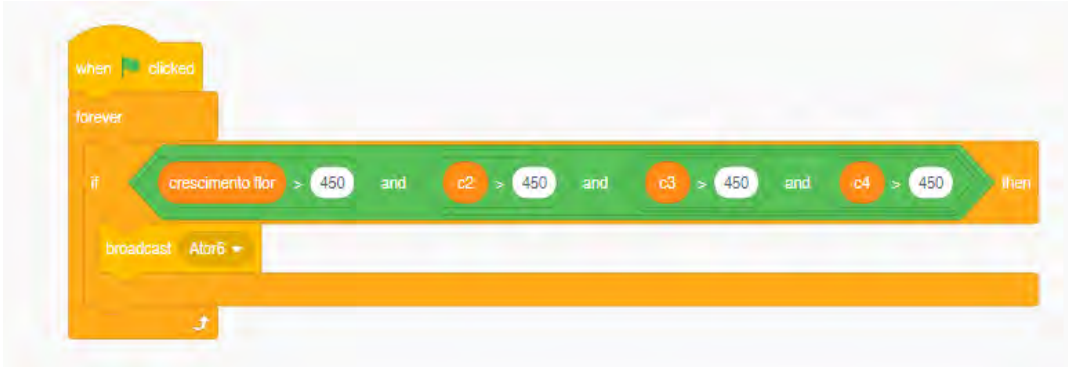
Através da fala da estudante Lorena, reafirmamos a nossa análise sobre os conceitos utilizados no jogo:

Pesquisador: Poderia destacar uma funcionalidade específica do seu jogo em que você aplicou com sucesso algum conceito de loops, eventos, sequências, paralelismo, condicionais, operadores lógicos ou dados?

Lorena: Design de fundo e personagens.

Ao engajar-se de maneira ativa na pesquisa, o pesquisador identifica que Lorena faz referência ao segmento de código denominado "ator6" ao abordar o design de fundo. O design de fundo e código em questão, "ator6", está disponível no Quadro 13:

Quadro 13 - Leiaute e fragmento de código do jogo (ator6) Flower responsável pela tela de finalização.

[ator6]


<p>O trecho de código apresenta um bloco "WhenGreenFlagClicked", que indica a execução de instruções quando o botão verde é clicado. Dentro desse bloco, há um loop infinito ("Forever") que continua a ser executado durante toda a execução do jogo.</p> <p>A condição dentro do loop verifica se o valor da variável "crescimento flor" é superior a 450 e se as variáveis "c2", "c3" e "c4" também são maiores que 450. Se essa condição for atendida, é emitido um broadcast com a mensagem "Ator6" usando o bloco "Event.Broadcast". Essa lógica indica que a emissão do broadcast ocorrerá quando todas as condições especificadas forem verdadeiras, sugerindo um evento significativo no jogo que envolve o crescimento das flores e outras condições relacionadas aos estágios do jogo.</p>

Fonte: elaborado pelo autor

O fragmento de código visto acima desempenha um papel crucial na elaboração da tela de encerramento do jogo, incorporando diversos conceitos do pensamento computacional, tais como loops e estruturas condicionais. A entrevista com Lorena revela não apenas a aplicação

de conceitos do pensamento computacional no código, mas também destaca a importância desses conceitos na criação de elementos visuais que enriquecem a experiência do jogador. O uso criativo de loops e condicionais no design de fundo e personagens demonstra uma compreensão sólida desses conceitos e sua aplicação prática na programação de jogos.

Prosseguindo com nossa análise das práticas do pensamento computacional, observamos que os estudantes não apenas aplicaram a abstração e modularização aos códigos do jogo, mas também a estenderam ao design de seus personagens, dividindo a construção deles em etapas distintas.

Vitória: Primeiro começamos com os desenhos, dos mais fáceis pros mais difíceis, assim como na programação.

Essa abordagem demonstra uma integração eficaz entre o processo criativo do design e os princípios fundamentais da programação. Ao dividir a construção dos personagens em etapas, os estudantes não apenas facilitam o desenvolvimento progressivo, mas também refletem uma compreensão avançada da importância da abstração e modularização em ambos os aspectos do projeto.

A analogia feita com a programação, onde começaram pelos desenhos mais simples e avançaram para os mais complexos, destaca a aplicação consistente de uma metodologia sistemática. Essa progressão ascendente não apenas proporciona uma experiência de aprendizado mais acessível, mas também ressalta a importância de estabelecer uma base sólida antes de abordar conceitos mais desafiadores.

A prática de dividir a criação de personagens em etapas revela não apenas competência técnica, mas também a capacidade de traduzir conceitos abstratos do pensamento computacional para a esfera visual e criativa. Esse enfoque integrado reflete uma compreensão holística do processo de desenvolvimento de jogos, onde a programação e o design se complementam harmoniosamente.

Uma dimensão essencial a ser explorada é a perspectiva "Conectando". De acordo com Resnick (2012), a criatividade e a aprendizagem são manifestações profundamente sociais. Dessa forma, o design de mídia computacional utilizando a plataforma Scratch é naturalmente enriquecido por interações com outras pessoas. Ao inquirir a estudante Maria Eduarda com a seguinte pergunta: "Você já percebeu benefícios ao criar colaborativamente com outros estudantes? Em caso afirmativo, como isso influenciou seus projetos?" Ela prontamente respondeu: "Sim, isso impactou na minha maneira de lidar com outras pessoas em situações de trabalho em equipe".

A resposta de Maria Eduarda sugere que a colaboração em projetos com colegas teve um impacto marcante em sua abordagem em contextos de trabalho colaborativo. Ao enfatizar que essa experiência "impactou na maneira como eu lido com outras pessoas em um trabalho em equipe", ela não apenas indica melhorias nos resultados práticos dos projetos, mas também ressalta a influência sobre sua dinâmica interpessoal.

Esta resposta está alinhada com os princípios discutidos por Resnick (2017), que sublinhou a natureza social intrínseca à criatividade e à aprendizagem, destacando como a interação com pessoas enriquece a experiência criativa. Maria Eduarda parece reconhecer não apenas os benefícios práticos da colaboração em projetos, mas também os impactos positivos que isso teve em suas habilidades interpessoais.

Assim, a resposta de Maria Eduarda destaca a relevância não apenas dos resultados palpáveis dos projetos colaborativos, mas também dos efeitos duradouros na sua abordagem social e na dinâmica de trabalho em equipe. Essa perspectiva está em sintonia com os fundamentos apresentados por Resnick sobre a interligação entre criatividade, aprendizagem e interações sociais.

Ao analisar a participação dos estudantes nos encontros na construção colaborativa do jogo "Flower", o pesquisador notou vividamente a expressão da criatividade em duas fases cruciais: a construção do dispositivo robótico e o processo de brainstorming para conceber o jogo. Essa observação destaca uma abordagem distintamente conectada de trabalho em conjunto, onde a troca de ideias desempenha um papel vital.

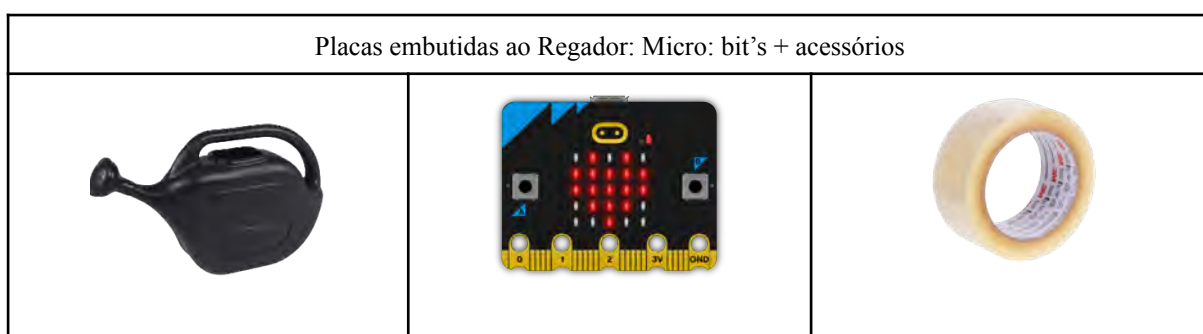


Ilustração 4 - Materiais de baixo custo usados na confecção do dispositivo robótico Regador.
Fonte: elaborado pelo autor

Durante a construção do dispositivo robótico, os estudantes não apenas colaboraram na execução de tarefas práticas, mas também contribuíram com ideias criativas. A escolha de materiais de baixo custo (Ilustração 4), o design do dispositivo e a resolução de desafios técnicos foram moldados pela interação e pela conexão de perspectivas criativas entre os membros da equipe, não se limitando às orientações do professor. Isso ocorre porque o

professor-pesquisador, atuando como facilitador do processo de aprendizagem, reconhece-se também como um aprendiz no processo conjunto de evolução com seu aluno. Nesse contexto, a abordagem não visa concentrar todas as ações no professor. Conforme destacado por Resnick (2017), o objetivo é fomentar vias que capacitem os estudantes a desenvolverem maior criatividade e autonomia ao longo de seu processo educativo, auxiliando-os na “maneira como compartilham, [refletem, desenvolvem] e colaboram [uns com outros em sala de aula], [e] fornecendo-lhes as melhores ferramentas, suportes e oportunidades para isso” (Resnick, 2017, p. 99, tradução nossa).



Figura 22- Apresentação e Brainstorming sobre a concepção do Game Flower (dados da pesquisa).
Fonte: acervo do autor

No processo de brainstorming (Figura 22) para a concepção do jogo "Flower", a criatividade emergiu de maneira proeminente. Os estudantes participaram ativamente da troca de ideias, incentivando a diversidade de abordagens e perspectivas. Esse intercâmbio criativo resultou em conceitos inovadores, demonstrando como a colaboração conecta mentes diversas e impulsiona a geração de soluções únicas.

A análise da participação dos estudantes sugere que a abordagem colaborativa vai além da simples execução de tarefas, destacando a importância intrínseca da troca de ideias. A criatividade não se limita à individualidade, mas floresce na sinergia conectada de mentes diversas, promovendo um ambiente de aprendizado dinâmico e estimulante. A criatividade, nesse contexto, não é apenas uma ferramenta para resolver problemas, mas um catalisador para a formação de ideias inovadoras por meio da conexão colaborativa. Essa abordagem ressalta como o trabalho conjunto e a troca contínua de ideias desempenham um papel vital na promoção da criatividade e no desenvolvimento de projetos significativos.

4.2.3 E2: Habilidades Do Pensamento Computacional

O trecho de código no Quadro 14 controla o comportamento de um personagem ("Ator1") que simula a animação do voo de um pássaro e desenha sua trajetória sinusoidal na tela. O personagem muda de figurino enquanto se move, e eventos de "land" são transmitidos quando a trajetória atinge uma posição específica na tela.

Quadro 14 - Trajetória sinusoidal do personagem Ator1 ‘continua’.

Representação visual do personagem “Ator1” e trajetória sinusoidal do voo	
	
Trecho do código Scratch	Funcionalidade

Quadro 14 - Continuação.



Limpeza Inicial:

- `Pen.Clear();` Limpa qualquer desenho anterior feito pelo personagem.
- `Pen.SetPenColorToColor(#ffa6c8);` Define a cor da caneta do personagem para uma tonalidade de rosa.
- `Looks.Show();` Mostra o personagem.

Loop "Repeat Until" com Condição de Posição Horizontal:

- `Motion.GoToXY(-400, Operator.Random(100, 180));` Move o personagem para uma posição inicial aleatória no eixo Y.
- `Looks.Show();` Mostra o personagem novamente.
- `Control.Wait(Operator.Random(1, 5));` Espera um tempo aleatório entre 1 e 5 segundos.
- `Repeat Until (Motion.XPosition() > 230);` Repete o bloco até que a posição horizontal do personagem seja maior que 230.

Ações no Loop "Repeat Until":

- `Pen.Down();` Abaixa a caneta, indicando que o personagem começará a desenhar.
- `y pos last time = Motion.YPosition();` Armazena a posição Y atual do personagem.
- `Motion.SetY((sin((Sensing.Timer() * wave frequency)) * amplitude));` Move o personagem verticalmente em uma trajetória sinusoidal com base no tempo e nas variáveis de frequência e amplitude.
- `If (Motion.YPosition() < y pos last time):` Verifica se o personagem está descendo.
 - `Looks.SwitchCostumeTo(fantasia1);` Troca o figurino para "fantasia1" se estiver descendo.
 - `Motion.ChangeXBy(forward step size);` Move o personagem para frente.
 - `Pen.Clear();` Limpa o desenho anterior.
- `Else: Se não estiver descendo (subindo):`
 - `Looks.NextCostume();` Avança para o próximo figurino.
 - `Motion.ChangeXBy((forward step size / upswing steepness));` Move o personagem para frente com um passo menor.
 - `Pen.Clear();` Limpa o desenho anterior.
- `Pen.Down();` Abaixa novamente a caneta.
- `Pen.Clear();` Limpa novamente o desenho.

Finalização do Loop "Repeat Until":

- `Looks.Hide();` Esconde o personagem.
- `Event.Broadcast("land");` Emite uma transmissão chamada "land".
- `Pen.Clear();` Limpa novamente o desenho.

Fonte: elaborado pelo autor

A abstração neste trecho de código se manifesta por meio da representação simplificada de conceitos complexos, seja por meio de variáveis, funções predefinidas, figurinos ou encapsulamento de comportamento. Essa abstração facilita a compreensão e a manipulação do código, tornando-o mais modular e flexível para futuras modificações ou extensões. O Quadro 15 detalha melhor os elementos de abstração no código:

Quadro 15 - Elementos da abstrações identificados no trecho de código Ator1.

Elemento	Representação de abstração
Variáveis como Abstrações de Conceitos	<ul style="list-style-type: none"> • O código utiliza diversas variáveis, como y pos last time, wave frequency, amplitude, forward step size, e upswing steepness. Cada uma dessas variáveis abstrai um conceito específico. • Exemplo: A variável wave frequency representa a frequência da trajetória sinusoidal, enquanto forward step size representa o tamanho do passo para frente do pássaro.
Uso de Funções Pré-definidas	<ul style="list-style-type: none"> • A plataforma de programação fornece funções predefinidas, como Motion.GoToXY(), Looks.Show(), Pen.Clear(), etc. Essas funções abstraem operações mais complexas, permitindo que o programador se concentre no comportamento desejado.
Figurinos como Abstrações Visuais	<ul style="list-style-type: none"> • O código utiliza figurinos para representar visualmente o pássaro em diferentes estados ou momentos. Cada figurino é uma abstração visual que comunica uma condição específica do personagem. • Exemplo: O figurino "fantasia1" representa um estado específico do pássaro durante o movimento.
Parâmetros de Movimento Abstraídos	<ul style="list-style-type: none"> • Os parâmetros relacionados ao movimento do pássaro, como posição inicial, direção, e velocidade, são abstraídos por meio de variáveis e funções específicas. Isso permite uma manipulação mais flexível e compreensível desses aspectos.
Encapsulamento de Comportamento	<ul style="list-style-type: none"> • O código encapsula o comportamento do pássaro em um bloco coeso de código. Detalhes complexos sobre como a trajetória sinusoidal é calculada ou como os figurinos são trocados são abstraídos, permitindo uma compreensão mais clara do comportamento geral.

Fonte: elaborado pelo autor.

Ao questionar a estudante Lorena sobre o processo de abstração em seu jogo "Flower", inquiri sobre a aplicação prática dessa técnica e como ela simplifica ou representa conceitos complexos para torná-los mais compreensíveis no contexto do jogo. Sua resposta foi: "O nosso jogo teve um processo de abstração bem claro, pois pegamos uma ideia bem simples para que todos conseguissem entender o jogo." A abordagem de Lorena de escolher uma ideia simples para garantir que todos pudessem entender o jogo é refletida no código, onde a

estrutura lógica e os elementos visuais são organizados de maneira a tornar o jogo acessível e compreensível para os jogadores.

Assim, a aplicação da abstração no trecho de código do jogo "Flower" parece estar alinhada com a intenção de criar uma experiência de jogo compreensível e acessível, reforçando a ideia de que a abstração é uma ferramenta essencial para simplificar conceitos complexos no contexto do desenvolvimento de jogos.

Outra habilidade que se destaca é a decomposição, evidenciada pela estrutura modular do código. Ao dividir o código em blocos lógicos, o programador adota uma abordagem de decomposição que torna cada parte do código mais gerenciável, facilitando a compreensão, manutenção e modificação futura do código. Essa decomposição é essencial para lidar com a complexidade de implementar comportamentos interativos em uma plataforma de programação visual. O Quadro 16 detalha como a decomposição é aplicada neste código específico:

Quadro 16 - Elementos da decomposição identificados no trecho de código Ator1.

Elemento	Representação de decomposição
Blocos Lógicos Distintos	O código é organizado em blocos lógicos distintos, cada um responsável por uma parte específica do comportamento do personagem. Esses blocos incluem a inicialização, o loop principal, o desenho da trajetória sinusoidal, a troca de figurinos, a ocultação do personagem e a emissão de uma transmissão.
Repetições Controladas	A estrutura de repetição Forever encapsula o comportamento contínuo do personagem, enquanto o loop Repeat Until detalha a lógica específica de desenho até que uma condição seja atendida (posição horizontal específica).
Manipulação Modular de Figurinos	A troca de figurinos é tratada de maneira modular, com blocos condicionais para decidir qual figurino usar com base na direção do movimento. Isso simplifica o entendimento do código relacionado à aparência visual do personagem.
Clareza na Inicialização e Finalização	A seção de inicialização e finalização do código é distinta, proporcionando uma visão clara sobre as etapas de configuração e limpeza do comportamento do personagem.

Fonte: elaborado pelo autor.

No que diz respeito à habilidade de pensamento computacional de decomposição neste trecho de código, ao questionar a estudante Maria Eduarda sobre como ela lida com desafios complexos na criação de jogos, a abordagem é explorar como ela divide esses problemas em partes menores mais gerenciáveis. A pergunta específica foi: "Ao enfrentar desafios complexos na criação de jogos, como você divide esses problemas em partes menores mais gerenciáveis? Pode fornecer um exemplo específico de como a decomposição foi útil em um de seus projetos?" A resposta de Maria Eduarda foi:

Maria Eduarda: "Normalmente, nós dividimos os problemas e conquistas em partes para tornar mais fácil e ágil a resolução, beneficiando assim outros projetos com essa habilidade."

A resposta da estudante Maria Eduarda indica que a decomposição é uma prática comum em seu trabalho de desenvolvimento de jogos digitais. Ao dividir problemas e conquistas em partes menores, ela e sua equipe buscam facilitar e agilizar a resolução, proporcionando benefícios não apenas para o projeto em questão, mas também para outros projetos futuros. A decomposição efetiva no código contribui para a clareza, manutenibilidade e entendimento modular do comportamento do personagem. Essa abordagem também se alinha à resposta de Maria Eduarda, destacando a utilidade da decomposição para enfrentar desafios complexos e agilizar a resolução de problemas.

A habilidade de reconhecimento de padrões no pensamento computacional também que compreende também por tomar uma solução (ou parte de) para um problema e generalizá-la para que possa ser aplicada a outros problemas e tarefas semelhantes (Bell; Witten; Fellows, 1998), é claramente demonstrada no trecho de código relacionado ao personagem "Ator1". Ao reconhecer padrões no movimento do pássaro e na trajetória sinusoidal, o código cria uma representação visual dinâmica e esteticamente agradável. A habilidade de reconhecimento de padrões é fundamental para criar comportamentos e aparências consistentes e previsíveis em ambientes de programação visual. O Quadro 17 detalha alguns dos elementos do reconhecimento de padrões encontrados no trecho do código.

Quadro 17 - Elementos de reconhecimento de padrões identificados no código Ator1.

Elemento	Representação de reconhecimento de padrões
Figurinos e Movimento Sincronizado	O código utiliza a troca de figurinos de maneira sincronizada com o movimento do pássaro. Quando o pássaro está descendo, o figurino é trocado para "fantasia1", e quando está subindo, o próximo figurino é selecionado. Essa sincronização cria um padrão visual que segue a trajetória do pássaro.
Lógica de Desenho Baseada em Condições	A lógica de desenho do pássaro é controlada por condições que verificam se a posição atual do pássaro é inferior à posição anterior ($\text{Motion.YPosition() < y pos last time}$). Essa verificação identifica o padrão do movimento do pássaro, distinguindo entre movimentos ascendentes e descendentes.
Loop de Movimento até uma Posição Específica:	A repetição Repeat Until é uma construção lógica que reconhece o padrão de movimento do pássaro até atingir uma posição horizontal específica (condição de saída do loop). Isso cria um padrão repetitivo de deslocamento e desenho até que o pássaro atinja a posição desejada.
Uso de Funções Matemáticas para Padrões Visuais:	A utilização da função matemática $\sin()$ para calcular a trajetória sinusoidal do pássaro introduz um padrão visual específico. O uso dessa função cria uma forma de onda que se repete ao longo do tempo, contribuindo para a aparência suave e cíclica do movimento.
Lógica de Troca de Figurinos em Resposta a Padrões de Movimento:	A lógica que decide qual figurino trocar em resposta ao movimento do pássaro é uma expressão de reconhecimento de padrões. A escolha de figurinos é determinada pela direção do movimento, proporcionando uma variação visual coerente.

Fonte: elaborado pelo autor.

Ao indagar a estudante Vitória com a seguinte pergunta: "Como o reconhecimento de padrões influencia suas decisões durante o desenvolvimento de jogos? Você poderia compartilhar uma situação em que identificar padrões foi fundamental para aprimorar a jogabilidade ou a experiência do usuário?" Ela prontamente respondeu: "Ao reconhecer que padrões acontecem no jogo, o seu entendimento fica bem mais fácil."

A resposta de Vitória sugere que o reconhecimento de padrões não apenas simplifica o entendimento do jogo, mas também pode ser crucial para melhorar a jogabilidade e a experiência do usuário. Identificar padrões permite que os desenvolvedores antecipem o comportamento do jogo, tomem decisões informadas e otimizem elementos visuais e interativos para criar uma experiência mais envolvente.

Na prática, o reconhecimento de padrões pode levar a ajustes refinados na lógica do jogo, na animação ou na resposta do usuário, contribuindo para uma experiência mais coesa e intuitiva. Nesse contexto, o código do personagem "Ator1" utiliza o reconhecimento de

padrões não apenas como uma ferramenta técnica, mas como um meio de criar uma estética visual e interativa consistente no jogo.

O código, também demonstra a habilidade de algoritmo ao definir uma sequência lógica de passos que controlam o comportamento do personagem. Esses passos são executados repetidamente, criando um padrão visual dinâmico enquanto o personagem se move e desenha na tela. O algoritmo delineado neste código contribui para a interatividade e estética do programa. O Quadro 18 detalha como o algoritmo é utilizado neste código específico:

Quadro 18 - Elementos de algoritmo identificados no trecho de código Ator1.

Elemento	Descrição
Fluxo Lógico Controlado por Eventos	O código é estruturado em torno do evento "WhenGreenFlagClicked", indicando que as instruções dentro desse bloco serão executadas quando a bandeira verde for clicada. Isso estabelece o ponto de partida e início do comportamento do personagem.
Loop Infinito "Forever"	O uso do bloco Forever cria um loop contínuo, representando o comportamento persistente do personagem enquanto o programa estiver em execução. Esse loop é interrompido apenas quando o programa é encerrado.
Inicialização e Configuração	No início do código, há instruções de inicialização que configuram o ambiente para o comportamento do personagem. Isso inclui a limpeza do desenho anterior (Pen.Clear()), a definição da cor da caneta (Pen.SetPenColorToColor(#ffa6c8)), e a exibição do personagem (Looks.Show()).
Movimento Aleatório e Aguardo	O código utiliza a função Motion.GoToXY() para posicionar o personagem em uma posição aleatória no eixo Y. Em seguida, há uma espera aleatória antes de começar o loop principal. Essa aleatoriedade contribui para a variabilidade do comportamento.
Loop Principal e Lógica de Desenho	O loop principal (Repeat Until) controla o movimento do personagem até atingir uma posição horizontal específica. Durante esse loop, há uma lógica de desenho que utiliza a função seno (sin()) para criar uma trajetória sinusoidal enquanto o personagem se move.
Troca de Figurinos e Limpeza do Desenho	Dentro do loop, há lógica para trocar os figurinos do personagem com base na direção do movimento. Além disso, o desenho é limpo em cada iteração para manter uma representação visual clara e suave da trajetória.
Ocultação do Personagem e Transmissão de Evento	Quando o loop atinge uma posição específica, o personagem é ocultado, uma transmissão chamada "land" é enviada e o desenho é limpo. Essa sequência indica a conclusão de um ciclo do comportamento do personagem.

Fonte: elaborado pelo autor.

Vitória descreve a elaboração de lógicas e sequências para as funcionalidades dos jogos da seguinte maneira:

Vitória: “Primeiramente pensávamos em como aquilo precisa acontecer, para depois desenvolver o algoritmo.”

A resposta de Vitória revela uma abordagem reflexiva e iterativa na criação de algoritmos para funcionalidades de jogos. Antes de iniciar a implementação, a equipe se dedica à análise detalhada de como a funcionalidade deve ocorrer, evidenciando uma compreensão atenta dos requisitos. Essa prática reflete uma tentativa consciente de traduzir a visão concebida para o jogo em lógica programática. No contexto do trecho de código do personagem "Ator1", percebemos a influência dessa mentalidade através da reflexão inicial sobre a mecânica do jogo, da visualização conceitual e da adaptação cuidadosa à lógica específica do jogo, aspectos todos moldados por essa abordagem reflexiva.

A análise do trecho de código do personagem "Ator1" revela habilidades significativas de pensamento computacional. Considerando que este código foi produzido por estudantes do ensino médio, é notável ver a aplicação de conceitos avançados de programação. A criatividade demonstrada na representação visual do movimento do personagem sugere um ambiente de aprendizado estimulante, onde os estudantes são incentivados a explorar tanto os aspectos técnicos quanto artísticos da programação.

4.3 E3: WAY TO THE GARDEN

Neste episódio, iremos explorar o evolução do Jogo Way To The Garden, estruturado em quatro etapas essenciais: "Pré-produção", "Reutilização de artefatos robóticos e codificação", "Discussão e Apresentação" e "Jogo em prática". Durante todo o ciclo de desenvolvimento, destacamos e aprofundamos nas diversas dimensões e habilidades do pensamento computacional, as quais serão minuciosamente abordadas ao longo deste episódio.

4.3.1 E3: Etapas De Desenvolvimento

A sequência de desenvolvimento do jogo Way To The Garden pode ser observada na Ilustração 5:



Ilustração 5: Iterações Multifásicas: Uma Análise Visual do Desenvolvimento do Way To The Garden (dados da pesquisa).

Fonte: Acervo do autor

Na análise da Ilustração 5, é possível observar o engajamento dos estudantes no processo de desenvolvimento do jogo. Na primeira imagem, destacamos a discussão aprofundada sobre a mecânica e o design, pautada pela utilização criteriosa de um jogo elaborado pela turma anterior do projeto como ponto de partida inspirador. Este método

reflexivo, baseado na análise crítica da criação precedente, demonstra um enfoque analítico e uma busca por referências sólidas no aprimoramento do projeto em questão.

Na segunda imagem, surge um artefato robótico volante que foi reaproveitado de uma iteração anterior do jogo. Os ajustes realizados pelos estudantes nesse dispositivo revelam uma abordagem proativa na modificação e refinamento da tecnologia existente. A pintura empregada no objeto não apenas evidencia o comprometimento estético, mas também sugere uma preocupação em conferir identidade visual única ao artefato.

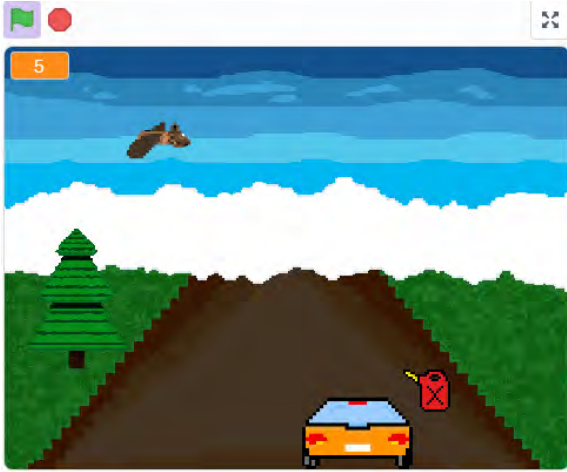

O ápice da narrativa se desenha na terceira imagem, quando os alunos compartilham o jogo finalizado com toda a turma. Esse momento não só representa a concretização palpável de esforços colaborativos, mas também estimula um diálogo construtivo em torno das potenciais melhorias a serem incorporadas. A análise crítica e a abertura para otimizações evidenciam a maturidade intrínseca do processo de criação, conferindo-lhe a característica marcante de ser um contínuo e evolutivo exercício de aprimoramento.

Na quarta imagem, a concretização do propósito do jogo se torna tangível ao observarmos uma paciente do hospital utilizando o volante modificado. Este cenário proporciona uma perspectiva pragmática do impacto positivo da iniciativa, não apenas como uma forma de entretenimento, mas também como uma ferramenta interativa benéfica para os pacientes.

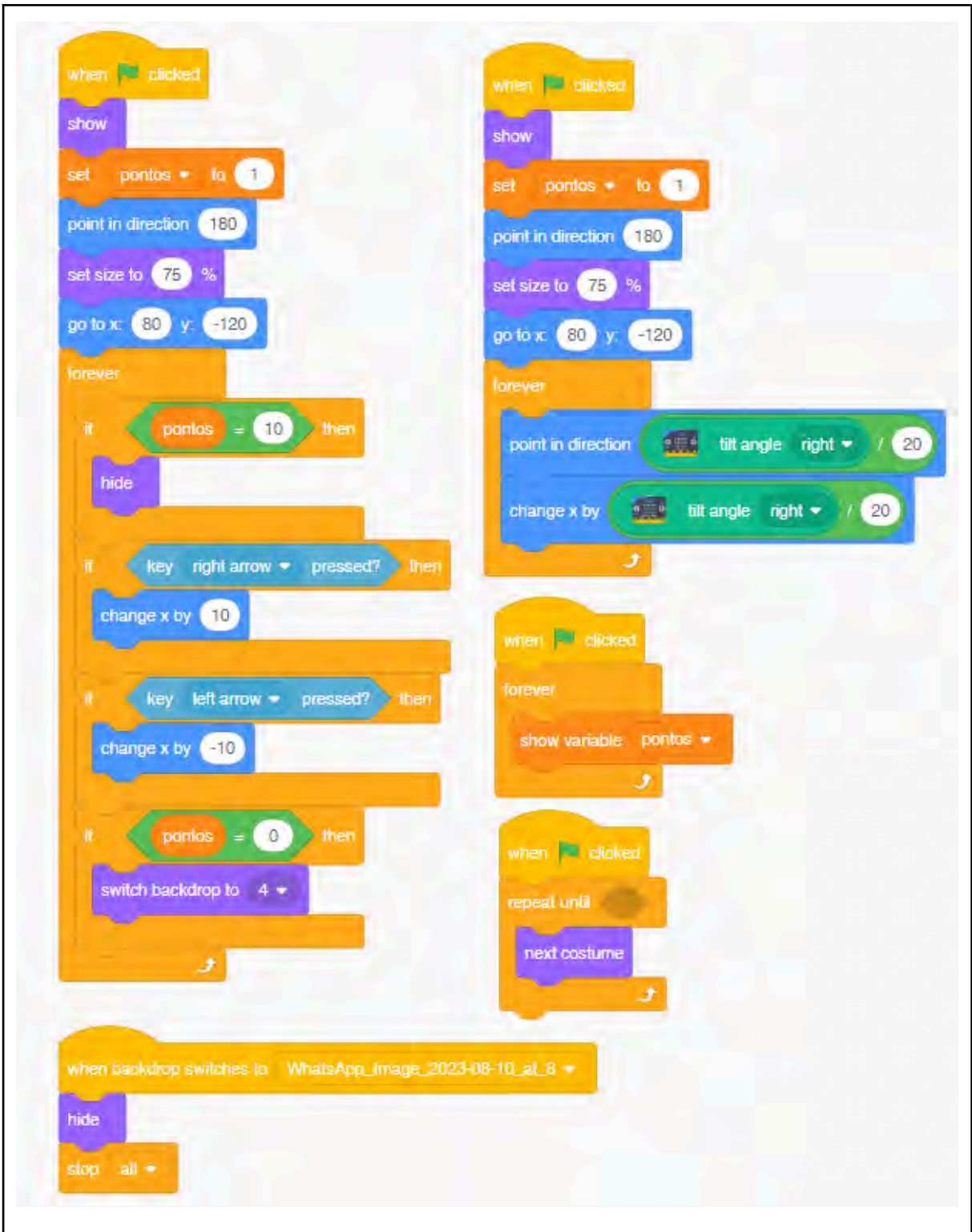
4.3.2 E3: Dimensões Do Pensamento Computacional

Para aprimorar a compreensão dos conceitos fundamentais do pensamento computacional, procederemos com uma análise detalhada do código a seguir.

Quadro 19 - Analisando o jogo Way To The Garden (dados da pesquisa) ‘continua’.

Leiaute e algoritmos	Descrição do jogo e do código
 A screenshot from the game 'Way To The Garden'. It shows a top-down view of a yellow car with a blue roof driving on a brown dirt road. To the right of the car is a red gas can with a yellow nozzle. The background features green grass, a blue sky with white clouds, and a brown bird flying in the sky. In the top left corner, there is a score of '5' in an orange box. The game window has standard OS window controls (minimize, maximize, close) in the top right.	<p>O jogo possui um cenário interativo simples onde o jogador controla um carro, ganha pontos coletando latas de gasolina e perde pontos colidindo com obstáculos. O objetivo é atingir 10 pontos para vencer o jogo. O jogo também inclui elementos visuais e auditivos para tornar a jogabilidade mais envolvente.</p>
[Carro 1]	
 A close-up, pixelated image of the yellow car. The car has a blue roof, red headlights, and a white license plate area. The body is yellow with black outlines for the wheels and bumper.	

Quadro 19 - Continuação.



O trecho de código é relativo ao objeto carro que move-se para direita e para a esquerda coletando e desviando-se de objetos. Analisando o código temos as seguintes instruções:

:

- Configura as propriedades iniciais do sprite quando a bandeira verde é clicada. Exibe o sprite, define a variável pontos para 1, ajusta direção, tamanho e posição. Inicia um loop infinito que reage à inclinação do dispositivo microbit;

Quadro 19 - Continuação.

- Verifica condições dentro de um loop: oculta o sprite se pontos atingir 10, movimenta o sprite em resposta às teclas de seta e troca o cenário se pontos for igual a 0;
- Exibe continuamente a variável pontos;
- Repete a troca de trajés do sprite até que uma condição seja atendida.
- Responde à mudança de cenário para "WhatsApp_Image_2023-08-10_at_8". Oculta o sprite e interrompe todos os scripts.

Fonte: elaborado pelo autor

Alguns exemplos de conceitos do pensamento computacional presentes nesta seção de código são ilustrados através de exemplos destacados no Quadro 20:

Quadro 20 - Conceitos do PC utilizados no trecho de código Carro 1.

Conceito	Exemplos no Código
Sequência	Configuração sequencial de propriedades iniciais do sprite no primeiro bloco WhenGreenFlagClicked().
Loops	Uso de um loop infinito Forever para reagir à inclinação do dispositivo no primeiro bloco WhenGreenFlagClicked().
Eventos	Utilização de eventos como WhenGreenFlagClicked() e WhenBackdropSwitchesTo(WhatsApp_Image_2023-08-10_at_8) para acionar a execução de blocos de código.
Condicionais	Verificação de condições como If (pontos == 10) para ocultar o sprite e If Sensing.KeyPressed(right arrow) para mover o sprite em resposta a teclas de seta.
Operadores	Uso de operadores como == (igual), ?? (coalescência nula) e += (incremento) em expressões como (microbit_getTiltAngle ?? - probably legacy function from Scratch 2.0 / 20).
Dados	Utilização de dados como variáveis (pontos, tamanho) e valores específicos (180, 75, 80, -120) para configurar propriedades do sprite e controlar o fluxo do programa.

Fonte: elaborado pelo autor.

Por meio das observações feitas pelos estudantes Igor e Guilherme, consolidamos nossa análise acerca dos conceitos empregados no jogo:

Igor: “Eu fiquei na parte da arte do jogo, entretanto, existe o looping de repetição das árvores, do morcego e da abelha, a criação dos eventos, tais como: Coletar Galões de Gasolina e Desviar dos Troncos de Árvore.”

Guilherme: “O conceito de loops, foi aplicado nos objetos de itens coletáveis e obstáculos, o de eventos foi na tela de morte ou de vencer o jogo.”

Como previamente abordado, na segunda fase do desenvolvimento do jogo, a prática de reutilização e remixagem se tornou evidente ao pesquisador ao constatar que os estudantes aproveitaram o dispositivo robótico "volante" concebido pela turma do ano anterior do projeto. Os conceitos do jogo também foram influenciados por um projeto anterior realizado por outra turma, conforme destacado na resposta do estudante Arthur:

Pesquisador: “Você já reutilizou ou remixou código de outros projetos em seus jogos? Se sim, pode compartilhar uma experiência específica e como isso contribuiu para o seu projeto?”

Arthur: “Utilizamos como referência o jogo de navio que foi produzido por uma turma anterior, mas nosso jogo foi praticamente criado do zero.”

A resposta de Arthur destaca a referência ao jogo de navio produzido por uma turma anterior, indicando uma prática de reutilização de código e conceitos para informar a criação do novo jogo. No entanto, é interessante notar que Arthur destaca que o jogo deles foi praticamente criado do zero, indicando uma abordagem mais personalizada e adaptativa.

Além disso, a prática de reutilização não se limita apenas ao *software*, mas também se estende ao hardware, como evidenciado pelo dispositivo robótico “volante”. Essa abordagem prática permite que os alunos apliquem conceitos aprendidos em projetos reais, promovendo uma compreensão mais profunda e duradoura.

O dispositivo robótico “volante” foi construído a partir de materiais reciclados e acessíveis, compreendendo um aro de bicicleta, um palete de madeira, um cabo de vassoura e uma placa BBC Microbit e foi projetado para o jogo Navio, conforme citado por Arthur. Os estudantes que optaram por reutilizar o dispositivo demonstraram cuidado com a estética do *joystick*, aplicando uma camada de pintura para aprimorá-lo visualmente.

O dispositivo robótico volante e os materiais utilizados na sua construção são apresentados na Ilustração 6:

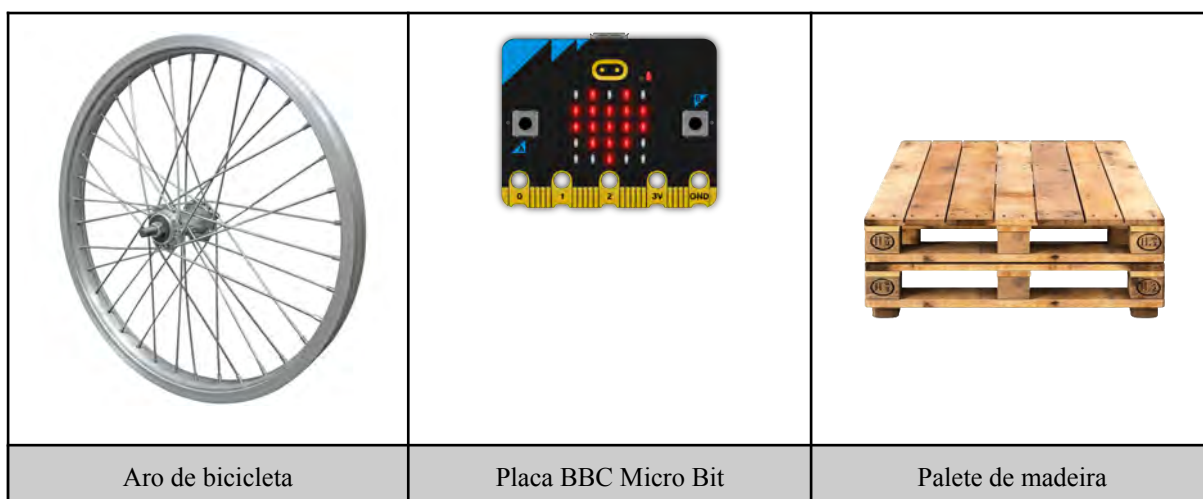




Ilustração 6 - Materiais utilizados na construção do dispositivo robótico volante (dados da pesquisa).
Fonte: acervo do autor.

A integração do dispositivo robótico de direção na narrativa adiciona uma dimensão física à prática de reutilização, mostrando como os conceitos e elementos de projetos anteriores podem transcender a esfera do código e influenciar até mesmo a construção de hardware. O cuidado dos estudantes com a estética do *joystick*, evidenciado pela aplicação de pintura, sugere uma valorização não apenas da funcionalidade, mas também da experiência visual e tátil do usuário.

Segundo Resnick (2012), ao adotar a abordagem computacional do questionamento, buscamos sinais de que os jovens não percebem uma desconexão entre as tecnologias ao seu redor e suas habilidades para interagir com as complexidades do mundo tecnológico. É crucial que os jovens se sintam capacitados para formular questionamentos sobre a tecnologia e interagir com ela de maneira significativa. Ao ser questionado sobre a importância de se questionar enquanto está criando jogos, Igor responde:

Igor: “Sim. É importante pois ajuda você a aprimorar sua ideia.”

Pesquisador: “Já pensou em mudar alguma coisa nas ferramentas que usa pra criar jogos?”

Igor: “Sim, já mudei de plataformas por falta de opções de estilo, falta de funcionalidade e etc.”

Nessa perspectiva de se questionar, a resposta de Igor nos revela que ele reconhece a importância de questionar durante o processo criativo, pois isso contribui para o aprimoramento de suas ideias. Ao mencionar a mudança de plataformas de criação de jogos, Igor evidencia uma postura crítica em relação às ferramentas que utiliza. Isso sugere que ele

não aceita passivamente as limitações impostas pelas tecnologias, mas busca alternativas que atendam melhor às suas necessidades e expectativas em termos de estilo e funcionalidade.


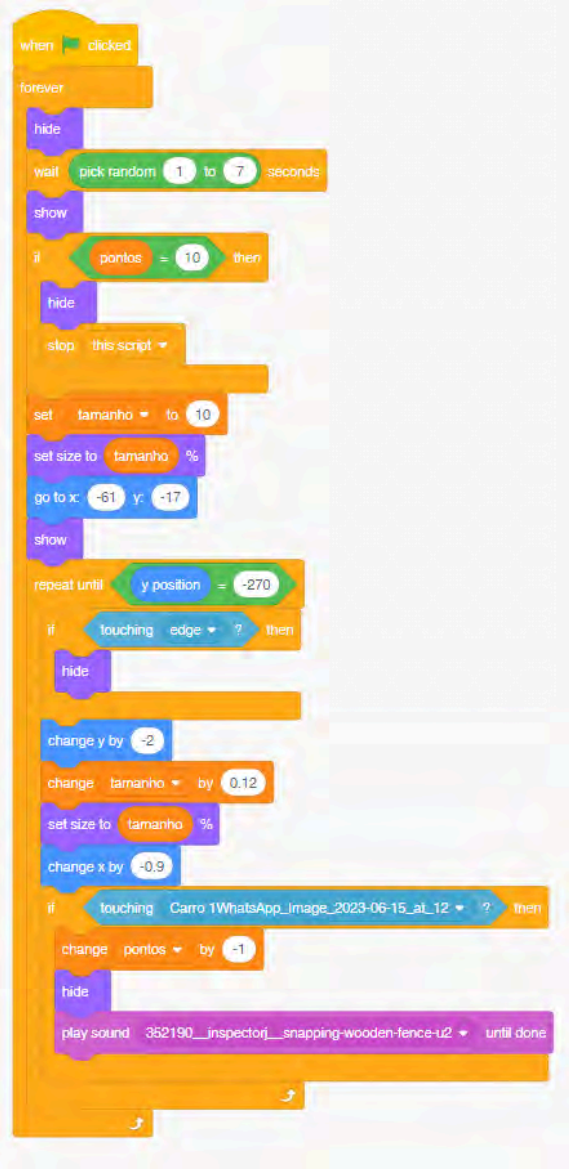
Essa atitude de questionamento e busca por melhorias nas ferramentas utilizadas revela uma conscientização por parte de Igor sobre a importância da interação significativa com a tecnologia. Ao reconhecer as limitações e buscar soluções, ele demonstra uma habilidade crucial para lidar com as complexidades do mundo tecnológico. Além disso, sua disposição para mudar de plataformas indica uma flexibilidade em adaptar-se a diferentes contextos e aprimorar suas habilidades em resposta às demandas do ambiente digital em constante evolução.

Assim, a resposta de Igor destaca a necessidade de os jovens não apenas utilizarem a tecnologia de maneira passiva, mas também de se envolverem ativamente no processo de criação e aprimoramento, questionando e adaptando as ferramentas disponíveis. Isso reforça a importância da abordagem computacional do questionamento, que busca identificar como os jovens percebem e interagem com as tecnologias ao seu redor, promovendo uma postura crítica e participativa diante do cenário tecnológico contemporâneo.

4.3.3 E3: Habilidades Do Pensamento Computacional

O código no Quadro 21 define o comportamento de um personagem chamado "Tronco de Madeira 1" no jogo. Esse personagem é projetado para aparecer e desaparecer em momentos específicos, movendo-se verticalmente para baixo em uma pista. Há a possibilidade de colisão com outro personagem denominado "Carro", resultando na subtração de um ponto do jogador em caso de impacto.

Quadro 21 - Personagem Tronco de Madeira do jogo Way To The Garden (dados da pesquisa).

Representação visual do personagem “Tronco de madeira”	
	
Trecho do código Scratch	Funcionalidade
	<p>Inicialização do Sprite:</p> <ul style="list-style-type: none"> O sprite é configurado para ser inicialmente escondido (Looks.Hide()). Um loop infinito (Forever) é iniciado para controlar o comportamento contínuo do sprite. <p>Exibição Aleatória e Condições de Ocultação:</p> <ul style="list-style-type: none"> O sprite é aleatoriamente exibido (Looks.Show()) após um intervalo de espera aleatório (Control.Wait(Operator.Random(1, 7))). Se a variável pontos atingir 10, o sprite é ocultado, e todos os scripts em execução são interrompidos (Stop(this script)). <p>Movimento do Sprite:</p> <ul style="list-style-type: none"> A posição e o tamanho do sprite são definidos (Motion.GoToXY(-61, -17), Looks.SetSizeTo(tamanho)). Um loop (Repeat Until) é iniciado até que a posição Y do sprite atinja -270. Dentro desse loop, o sprite se move para cima e aumenta de tamanho gradualmente. Se o sprite tocar a borda (edge), ele é ocultado (Looks.Hide()). <p>Interação com Outros Objetos:</p> <ul style="list-style-type: none"> Se o sprite tocar um objeto chamado "Carro 1WhatsApp_Image_2023-06-15_at_12", a variável pontos é decrementada por 1 (pontos += -1), o sprite é ocultado (Looks.Hide()), e um som é reproduzido (Sound.PlayUntilDone(352190__inspectorj__snapping-wooden-fence-u2)). <p>Condições de Ocultação Adicionais:</p> <ul style="list-style-type: none"> Fora do loop principal, outro loop infinito verifica se pontos é igual a 10 ou igual a 0. Se for o caso, o sprite é ocultado (Looks.Hide()). <p>Traje do Sprite (Costume):</p> <ul style="list-style-type: none"> O sprite utiliza o traje "Madeira 1.png".

Fonte: Elaborado pelo autor

A abstração neste trecho de código se manifesta por meio da simplificação e generalização de conceitos complexos relacionados ao comportamento do personagem "Tronco de Madeira 1". Isso é alcançado por meio de:

Quadro 22 - Elementos da abstração identificados no trecho de código do jogo Way To The Garden.

Elemento	Representação de abstração
Encapsulamento de Comportamento	<ul style="list-style-type: none"> ● O código encapsula o comportamento do personagem "Tronco de Madeira 1" em uma estrutura organizada, ocultando detalhes de implementação complexos. ● Os detalhes específicos de como o personagem é exibido, movido e interage com outros objetos são abstraídos para funções e blocos de código específicos.
Abstração de Interações	<ul style="list-style-type: none"> ● A lógica de interação entre o "Tronco de Madeira 1" e o "Carro" é abstraída para uma condição geral de colisão (<code>Sensing.TouchingObject(Carro1WhatsApp_Image_2023-06-15_at_12)</code>). ● A subtração de pontos é uma abstração de uma ação mais complexa que ocorre no contexto do jogo.
Abstração de Eventos	<ul style="list-style-type: none"> ● A aparência e desaparecimento do personagem são abstraídos para o uso de comandos como <code>Looks.Show()</code> e <code>Looks.Hide()</code>. ● A execução contínua do código é abstraída para loops (<code>Forever</code>) e eventos (<code>WhenGreenFlagClicked()</code>), isolando a lógica do jogo.
Abstração de Movimento	<ul style="list-style-type: none"> ● O movimento vertical do personagem é abstraído para comandos como <code>Motion.ChangeYBy(-2)</code>, <code>Motion.SetSizeTo(tamanho)</code>, etc.

Fonte: elaborado pelo autor.

Essas manifestações de abstração tornam o código mais legível, modular e fácil de entender, promovendo uma compreensão mais eficaz do comportamento do personagem e das interações dentro do contexto do jogo.

Ao indagar o estudante Guilherme sobre o processo de abstração em seu jogo Way To The Garden, questioneei: Como você descreveria a aplicação da abstração em sua prática? Em que medida você simplifica ou representa conceitos complexos para torná-los mais compreensíveis no contexto do seu jogo?" Sua resposta foi reveladora: "Normalmente, não percebo de imediato o que pode ser simplificado. A percepção ocorre, muitas vezes, de forma acidental, quando realizo alterações ou ao tentar corrigir algo."

A resposta de Guilherme revela uma abordagem mais intuitiva e reativa à habilidade de abstração no desenvolvimento de jogos digitais. Ele relata que geralmente não percebe o que pode ser simplificado até que acidentalmente faça alterações ou tente corrigir algo no código. Isso sugere uma falta de consciência prévia sobre oportunidades de simplificação durante o processo de design.

A abstração reativa, como descrita por Guilherme, ocorre após a detecção de problemas ou acidentes no código. Isso indica que, em vez de antecipar e identificar proativamente elementos que podem ser abstraídos, ele aborda a abstração como uma resposta a dificuldades encontradas durante o desenvolvimento.

A resposta também destaca a possibilidade de Guilherme estar em um estágio de desenvolvimento em que ainda está ganhando experiência. Conforme ele se familiariza mais com padrões e práticas de abstração, é provável que desenvolva uma consciência mais aguçada para identificar oportunidades de simplificação durante a fase inicial do desenvolvimento. Guilherme pode se beneficiar de uma abordagem mais reflexiva durante o processo de desenvolvimento, considerando ativamente como simplificar e abstrair conceitos complexos antes que problemas ocorram. Isso poderia resultar em código mais eficiente e fácil de entender desde o início.

Em última análise, a resposta destaca uma oportunidade contínua de aprendizado para Guilherme, encorajando-o a refletir proativamente sobre estratégias de abstração durante o desenvolvimento e aprimorar suas habilidades de pensamento computacional ao longo do tempo.

A habilidade de decomposição no pensamento computacional envolve dividir um problema complexo em partes menores e mais gerenciáveis. No contexto do trecho de código fornecido, podemos observar elementos de decomposição no Quadro 23:

Quadro 23 - Elementos da decomposição identificados no trecho de código do jogo Way To The Garden ‘continua’.

Elemento	Representação de decomposição
Funções Modularizadas	O código utiliza funções e blocos de código separados para diferentes funcionalidades, como <code>WhenGreenFlagClicked()</code> para configuração inicial, <code>Forever</code> para loops contínuos, e <code>WhenBackdropSwitchesTo()</code> para manipulação de eventos de troca de cenário. Essa modularização facilita a compreensão do código, pois cada bloco executa uma tarefa específica.
Loops e Controle de Fluxo	A decomposição é evidente na utilização de loops (<code>Forever</code> e <code>Repeat Until</code>) para repetir determinadas ações, como a movimentação do sprite e a verificação de condições específicas. Cada loop tem uma função específica, contribuindo para a organização e compreensão do código.
Condicionais para Tomada de Decisão:	A presença de estruturas condicionais (<code>If</code>) demonstra a decomposição de lógica condicional no código. Condições como <code>pontos == 10</code> ou <code>Sensing.TouchingObject(_edge_)</code> são avaliadas separadamente para controlar o comportamento do programa.

Quadro 23 - Continuação.

Atribuição e Manipulação de Variáveis:	A decomposição é percebida na atribuição e manipulação de variáveis, como pontos e tamanho. Cada variável desempenha um papel específico, e suas mudanças são controladas de maneira independente, contribuindo para uma compreensão mais clara.
Tratamento de Colisões	A detecção de colisões e as ações associadas (por exemplo, redução de pontos, reprodução de som) são decompostas em condições e ações específicas. Isso facilita a compreensão do que acontece quando ocorre uma colisão.
Coordenação de Movimento	A decomposição é evidente na coordenação do movimento do sprite, envolvendo mudanças nas coordenadas (<code>Motion.GoToXY()</code>) e ajustes de tamanho (<code>Looks.SetSizeTo()</code>). Cada aspecto do movimento é tratado separadamente.

Fonte: elaborado pelo autor.

A estudante Raíssa diz que “para o script dos troncos de árvores foi necessário passar por uma decomposição, desde aumento/diminuição da escala, movimentação dentro dos parâmetros da estrada.”.

A resposta da estudante Raíssa revela uma aplicação consciente e prática da habilidade de decomposição no pensamento computacional ao enfrentar desafios complexos na criação de jogos, com destaque para o script dos troncos de árvores. Ao abordar esse componente específico do projeto, Raíssa identifica diversas subtarefas, como controle do tamanho por meio do aumento/diminuição da escala e a gestão da movimentação dos troncos dentro dos parâmetros da estrada.

A abordagem modular adotada por Raíssa é clara ao mencionar aspectos específicos do comportamento dos troncos, como ajuste de escala e movimentação. Cada um desses elementos é tratado como uma unidade separada, indicando uma estratégia de decomposição que facilita a compreensão e a manipulação individual de cada componente.

A decisão de decompor o script dos troncos de árvores em partes menores demonstra a prática de Raíssa em simplificar o gerenciamento do código. Essa estratégia não apenas torna cada componente mais acessível para compreensão, teste e modificação, mas também contribui para um desenvolvimento mais eficiente do jogo como um todo.

A habilidade de reconhecimento de padrões é fundamental para entender a dinâmica do jogo, identificar comportamentos recorrentes e antecipar o impacto das interações. No contexto desse código, o estudante demonstra a capacidade de identificar e manipular padrões de movimento, visibilidade, tamanho e eventos, contribuindo para a construção de uma lógica coerente e interativa no jogo. Esses elementos se manifestam de maneira notável nas seguintes situações:

Quadro 24 - Elementos do reconhecimento de padrões identificados no trecho de código do jogo Way To The Garden.

Elemento	Representação do reconhecimento de padrões
Padrão de Movimento Vertical	O código apresenta um padrão consistente de movimento vertical do personagem. A instrução <code>Motion.ChangeYBy(-2)</code> sugere um deslocamento para cima em incrementos de 2 unidades, o que cria um padrão de movimento ascendente.
Padrão de Aparição e Desaparecimento	O uso de instruções <code>Looks.Hide()</code> e <code>Looks.Show()</code> dentro de um loop indica um padrão de aparência e desaparecimento do personagem. Isso pode ser interpretado como um mecanismo de alternância entre visibilidade e invisibilidade, criando um padrão de comportamento.
Padrão de Mudança de Tamanho	A variável tamanho é usada para controlar o tamanho do personagem, sendo incrementada por 0.12 em cada iteração. Isso sugere um padrão de aumento gradual de tamanho ao longo do tempo.
Padrão de Condições para Ocultar o Personagem	Existem condições que, quando atendidas, levam ao ocultamento do personagem. Por exemplo, se o personagem colidir com a borda (edge), ele é oculto. Essa lógica cria um padrão de resposta a eventos específicos.
Padrão de Pontuação e Condições de Encerramento	A pontuação (pontos) é utilizada como um indicador para determinar a visibilidade do personagem e encerrar o script quando atinge 10 pontos. Essa lógica estabelece um padrão de progressão e finalização do jogo.
Padrão de Som ao Colidir com Objeto Específico	Um padrão é estabelecido ao reproduzir um som específico (352190 __inspectorj__snapping-wooden-fence-u2) quando o personagem colide com o objeto "Carro 1WhatsApp_Image_2023-06-15_at_12".

Fonte: elaborado pelo autor.

No contexto do trecho de código analisado, a habilidade de reconhecimento de padrões é destacada durante a entrevista com o estudante Igor. Ao questionar como essa habilidade influencia suas decisões no desenvolvimento de jogos e pedir um exemplo prático, Igor compartilhou uma percepção valiosa relacionada ao aspecto visual do projeto.

Ele observou que, no decorrer do desenvolvimento, identificou que imagens com dimensões de 64x64 pixels eram eficazes para detalhes e permitiam economizar tempo. Essa observação prática, baseada no reconhecimento de padrões visuais, desempenhou um papel significativo no processo criativo. Igor mencionou que essa abordagem foi bem recebida pela equipe em 70-90% das vezes em que enviava uma imagem, contribuindo para uma apreciação consistente.

A resposta de Igor destaca como o reconhecimento de padrões é uma ferramenta valiosa durante o desenvolvimento de jogos, especificamente no contexto da escolha de imagens. A observação de que imagens de 64x64 pixels eram eficazes para detalhes e

economia de tempo evidencia um discernimento de padrões visuais que influenciou decisões de design.

O reconhecimento de que essa resolução específica era bem recebida pela equipe (70-90% das vezes) indica a aplicação prática desse padrão na produção de conteúdo. Ao seguir esse padrão, Igor não apenas poupa tempo, mas também estabelece um nível consistente de qualidade, contribuindo para a satisfação da equipe.

A resposta sugere que Igor utiliza o reconhecimento de padrões não apenas para otimizar o desenvolvimento, mas também para alinhar as preferências da equipe, indicando uma compreensão eficaz das convenções visuais que agradam ao grupo. Isso não só aprimora a eficiência, mas também fortalece a coesão da equipe em relação às escolhas visuais no desenvolvimento do jogo.

Em resumo, a resposta de Igor destaca como o reconhecimento de padrões, neste caso, visual, influenciou decisões práticas, resultando em eficiência, economia de tempo e alinhamento com as preferências da equipe, contribuindo para uma experiência de desenvolvimento mais suave e consistente.

No contexto do trecho de código fornecido, a habilidade do pensamento computacional relacionada à algoritmo é evidente na estruturação das instruções que controlam o comportamento do personagem ("Tronco de Madeira 1") no jogo. Vamos analisar alguns aspectos que destacam essa habilidade:

Quadro 25 - Elementos de algoritmos identificados no trecho de código do jogo Way To The Garden 'continua'.

Elemento	Representam de algoritmo
Sequencialidade e Controle de Fluxo	O código utiliza eventos específicos, como "WhenGreenFlagClicked", para indicar o início da execução do programa. Dentro desses eventos, há uma sequência clara de comandos que o personagem seguirá. Por exemplo, o uso de loops "Forever" indica a execução contínua de determinadas ações.
Condicionais	O trecho de código incorpora declarações condicionais ("If") para tomar decisões com base em certas condições. Por exemplo, verifica se a variável "pontos" é igual a 10 e, nesse caso, esconde o personagem e interrompe o script. Isso demonstra a capacidade de criar caminhos distintos com base em condições específicas.
Loops	O uso do bloco "Forever" e "Repeat Until" representa a aplicação da repetição controlada por condições. No caso, o movimento do personagem para baixo na tela é repetido até que sua posição Y alcance um valor específico.
Atribuição de Variáveis	A manipulação de variáveis, como "pontos" e "tamanho", é uma parte essencial de algoritmos. A atribuição e modificação dessas variáveis contribuem para o controle dinâmico do comportamento do jogo.

Quadro 25 - Continuação.

Interação com Eventos	O código responde a eventos específicos, como toques em objetos ("Sensing.TouchingObject"). Isso reflete a capacidade de criar algoritmos que reagem a eventos externos, tornando a interação do usuário uma parte crucial do jogo.
Controle de Aparência e Som	O código manipula a aparência e o som do personagem com blocos como "Looks.Show()", "Looks.Hide()", e "Sound.PlayUntilDone()". Isso demonstra a habilidade de integrar elementos visuais e auditivos no algoritmo, contribuindo para uma experiência de jogo mais envolvente.
Coordenação de Múltiplas Ações	A combinação de diferentes comandos, como alterar o tamanho, mover-se na tela e reagir a colisões, destaca a habilidade de coordenar múltiplas ações em um algoritmo coeso.

Fonte: elaborado pelo autor.

Quando questionado sobre a elaboração de lógicas e sequências para as funcionalidades do jogo, e como projeta e implementa algoritmos, Guilherme, o estudante, respondeu da seguinte forma:

Guilherme: “Eu começo imaginando como é a realidade e como é em programação e tento passar a realidade em programação na minha cabeça, depois tento implementar isso no projeto em si. Me lembro que estava tendo dificuldade em fazer um item coletável aparecer na tela então, me acalmei e pensei como era que eu queria e passei para programação”

A resposta fornecida por Guilherme oferece insights valiosos sobre a forma como ele aborda a criação de algoritmos no desenvolvimento de jogos. Sua abordagem é fundamentada em uma visualização inicial da realidade desejada, seguida por uma transição habilidosa para a implementação prática no ambiente de programação.

Ao mencionar a dificuldade específica relacionada à aparição de um item coletável na tela, Guilherme destaca sua capacidade de enfrentar desafios de forma metódica. A menção de acalmar-se antes de prosseguir com a resolução do problema sugere uma abordagem reflexiva e controlada diante de obstáculos, isso mostra uma mentalidade adaptativa e receptiva a ajustes contínuos durante o desenvolvimento.

A importância atribuída à experiência do usuário também é evidente em sua resposta. Ao pensar não apenas na funcionalidade técnica, mas também em como essa funcionalidade se traduz na experiência do jogador, Guilherme demonstra uma consideração holística no processo de criação.

Em última análise, a resposta de Guilherme destaca uma abordagem equilibrada entre a concepção inicial, a resolução prática de problemas, a consideração da experiência do usuário e a capacidade de aprendizado contínuo.

4.4 AS IMPLICAÇÕES DA CONSTRUÇÃO DE JOGOS DIGITAIS E DISPOSITIVOS ROBÓTICOS PARA O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL DE ESTUDANTES DO ENSINO MÉDIO

Os resultados discutidos nesta seção são interpretações e relações desenvolvidas a partir do conjunto de dados apresentados e analisados nas seções anteriores. Com foco nos Episódios Analíticos E1, E2 e E3, esta seção se concentra em explorar as dimensões e implicações do Pensamento Computacional, conforme identificado nos projetos analisados. Através deste exame, buscamos compreender as características evidenciadas pelos dados da pesquisa e seu impacto na experiência educacional dos alunos.

Nos Episódios Analíticos E1, E2 e E3, observamos a aplicação prática dos conceitos, práticas e perspectivas do Pensamento Computacional como delineado por Resnick (2012). Esses episódios revelam a interação dos estudantes com as dimensões do Pensamento Computacional e suas implicações no desenvolvimento de habilidades críticas.

No E1, a criação do jogo "Bird In The Sky" e seu dispositivo robótico correspondente demonstram uma aplicação significativa dos conceitos de programação. Os alunos mostraram habilidades em decomposição de problemas, abstração e reconhecimento de padrões, essenciais na codificação e design do jogo. Este episódio também revela práticas colaborativas e uma perspectiva inovadora, onde a tecnologia é vista como uma ferramenta para a criatividade.

O episódio E2, envolvendo o jogo "Flower" e o Regador-Robótico, ressalta a importância das práticas de design e desenvolvimento iterativo. A reutilização de conceitos e a adaptação de dispositivos robóticos preexistentes ilustram uma abordagem prática e econômica da aprendizagem, destacando a importância da resiliência e do pensamento adaptativo.

Em E3, o "Way to The Garden" integra a concepção de um volante reutilizado, enfatizando a interconexão entre o pensamento computacional e soluções físicas tangíveis. Este episódio destaca como os alunos aplicaram o pensamento computacional para resolver desafios práticos, evidenciando a conexão entre teoria e prática.

A pesquisa apontou que o desenvolvimento de jogos digitais tem um impacto substancial na evolução do pensamento computacional dos estudantes, particularmente nas dimensões de conceitos, práticas e perspectivas. Os dados ilustram como os estudantes, ao navegarem pelo processo de codificação, incorporaram habilidades cruciais de pensamento

computacional, incluindo a aplicação de conceitos fundamentais e a manipulação de elementos de programação.

Dentre as habilidades observadas, padrões de movimento vertical, aparição e desaparecimento de personagens, alteração progressiva de tamanho, condições para ocultação e revelação, bem como a gestão de pontuação e critérios de conclusão de etapas se destacaram. Esses padrões não só estruturaram a lógica do jogo, mas também fomentaram um entendimento mais profundo das dinâmicas de programação.

A pesquisa também destacou a importância de uma estrutura modularizada, na qual diferentes funções e blocos de código foram criados para tarefas distintas, promovendo a clareza e a eficiência do desenvolvimento. Loops e mecanismos de controle de fluxo possibilitaram a repetição controlada de ações, e a utilização de estruturas condicionais permitiu aos estudantes tomar decisões baseadas em situações específicas do jogo.

A habilidade de atribuir e manipular variáveis foi evidenciada pela capacidade dos estudantes de controlar dinamicamente aspectos como pontuação e características dos personagens. O tratamento inteligente de colisões e a coordenação harmoniosa de movimentos complexos refletem a aplicação efetiva de algoritmos.

O encapsulamento do comportamento, a abstração de interações, eventos e movimentos permitiram aos estudantes simplificar conceitos complexos, tornando-os mais acessíveis e gerenciáveis. Essa abordagem facilitou a compreensão do jogo como um todo e destacou a relevância da abstração no pensamento computacional.

Além disso, os dados coletados e analisados nas seções anteriores revelam várias outras características marcantes relacionadas ao desenvolvimento do Pensamento Computacional nos estudantes envolvidos nos Episódios Analíticos E1, E2 e E3. Estas características são fundamentais para entender o impacto e a eficácia da abordagem adotada no projeto.

1. Engajamento e Motivação

Uma das características mais evidentes é o alto nível de engajamento e motivação dos alunos. A oportunidade de trabalhar em projetos práticos, como o desenvolvimento de jogos e dispositivos robóticos, gerou um interesse genuíno, levando a um aprendizado mais profundo e significativo. Esse engajamento é um indicador importante da eficácia de estratégias de ensino baseadas em projetos.

2. Criatividade e Inovação

Os dados evidenciam a criatividade e inovação entre os alunos. A liberdade de explorar, criar e adaptar suas próprias soluções tecnológicas permitiu que eles expressassem

suas ideias de maneiras únicas. Esta característica é crucial no contexto atual, onde a inovação é altamente valorizada em diversas áreas.

3. Habilidades de Resolução de Problemas

Os alunos não apenas aplicaram conceitos de computação na prática, mas também desenvolveram estratégias eficazes para superar obstáculos e desafios durante o projeto. Essa habilidade é um componente essencial do Pensamento Computacional e é altamente transferível para outras áreas do conhecimento e situações da vida real.

4. Colaboração e Trabalho em Equipe

Os projetos incentivaram a colaboração e o trabalho em equipe. Os alunos aprenderam a compartilhar ideias, dividir tarefas e integrar suas habilidades individuais em prol de um objetivo comum. Esta característica é particularmente relevante em um mundo onde a capacidade de trabalhar efetivamente em equipe é uma competência fundamental.

5. Autonomia e Autoaprendizagem

Ao se depararem com desafios técnicos, muitos buscaram ativamente por recursos e estratégias para resolver problemas, demonstrando durante as atividades sujeitos ativos na construção do conhecimento.

6. Pensamento Crítico e Reflexivo

Os dados mostram que os alunos desenvolveram um pensamento crítico e reflexivo. Eles foram capazes de avaliar suas próprias soluções, identificar pontos de melhoria e refletir sobre as implicações de suas escolhas de design, algoritmo e programação, sinalizando assim o desenvolvimento do Pensamento Computacional.

7. Interdisciplinaridade

Percebemos que as atividades desenvolvidas no projeto revelaram que houve um diálogo entre os conteúdos/conceitos de Matemática com outras disciplinas. Os alunos aplicaram conceitos de matemática, ciências e artes em seus projetos de tecnologia, demonstrando como o Pensamento Computacional pode ser um elo integrador no currículo do ensino médio.

Os resultados desta pesquisa têm implicações significativas para a educação em Pensamento Computacional. Eles sugerem que a integração de projetos práticos de computação no currículo pode melhorar significativamente o engajamento dos alunos e o desenvolvimento de habilidades essenciais. Além disso, os dados indicam que tais atividades promovem uma aprendizagem mais profunda e significativa, preparando os alunos para os desafios do futuro em um mundo cada vez mais tecnológico.

5 CONSIDERAÇÕES FINAIS

As considerações finais desta dissertação destacam a relevância e os resultados alcançados na investigação sobre o impacto do desenvolvimento de jogos digitais e dispositivos robóticos no aprimoramento do pensamento computacional de estudantes do ensino médio. A pesquisa, estruturada em torno de uma pergunta central e objetivos cuidadosamente definidos, revelou insights significativos sobre a interseção entre práticas construcionistas, tecnologia educacional e habilidades computacionais.

A indagação principal, focada em entender as implicações dessas atividades práticas no pensamento computacional, foi abordada através de uma metodologia qualitativa robusta. A análise dos dados, que incluiu estudos de caso como o desenvolvimento do jogo "Bird in The Sky", proporcionou uma compreensão profunda de como conceitos computacionais são aplicados em contextos práticos e interativos. Este método mostrou-se eficaz na obtenção de uma visão detalhada do processo educacional e na identificação das habilidades específicas desenvolvidas pelos alunos.

Os resultados do projeto foram significativos. Ficou claro que a construção de jogos e dispositivos não só facilitou a compreensão dos princípios de computação, mas também engajou os estudantes de forma significativa. As atividades práticas transcendem o aprendizado tradicional, permitindo que os alunos apliquem e experimentem com conceitos como loops, eventos, condicionais, operadores lógicos e manipulação de dados. Esta abordagem prática contribuiu substancialmente para aprofundar a compreensão e o interesse dos alunos no campo da computação.

Além disso, a pesquisa teve implicações práticas além do contexto educacional. A aplicação dessas atividades no tratamento de sintomas da doença de Parkinson ilustra o potencial significativo dessas abordagens tecnológicas. Isto não só valida a importância do Pensamento Computacional na educação, mas também abre novos caminhos para sua aplicação em campos como a saúde. Pois acreditamos que estudos futuros podem expandir o escopo para incluir uma variedade maior de contextos educacionais e demográficos. Uma vez que investigar como diferentes abordagens pedagógicas no ensino do Pensamento Computacional podem influenciar o aprendizado e a retenção de conhecimento dos alunos.

Este estudo não apenas alcançou seus objetivos, mas também ofereceu uma contribuição valiosa para o campo da educação tecnológica e o desenvolvimento de soluções terapêuticas inovadoras. A integração de tecnologias educacionais no currículo no ensino

médio surge como uma estratégia promissora, preparando os estudantes para enfrentar os desafios futuros com uma base sólida em Pensamento Computacional e habilidades práticas.

O reconhecimento do Pensamento Computacional como uma competência essencial nas diretrizes curriculares da BNCC ressalta a necessidade premente de sua incorporação efetiva no ambiente educacional. Ao servir como inspiração para projetos relacionados a esse tema, o trabalho apresentado não apenas demonstra seu valor prático, mas também evidencia a capacidade transformadora do Pensamento Computacional na formação dos estudantes. Ao integrar essa habilidade fundamental ao currículo, os professores não apenas capacitam os alunos com as competências necessárias para enfrentar os desafios do mundo digital em constante evolução, mas também promovem uma educação alinhada com as demandas contemporâneas.

Para trabalhos futuros, sugere-se explorar mais a fundo o impacto das atividades práticas de desenvolvimento de jogos digitais e dispositivos robóticos em diferentes contextos educacionais e demográficos. Além disso, investigar como diferentes abordagens pedagógicas podem influenciar o aprendizado e a retenção de conhecimento dos alunos no ensino do Pensamento Computacional é uma área de pesquisa promissora. Também seria interessante expandir o escopo para incluir outras áreas de aplicação além da educação e da saúde, como a indústria e o mercado de trabalho. Por fim, considerar estudos longitudinais para avaliar o impacto a longo prazo dessas práticas educacionais no desenvolvimento cognitivo e nas habilidades dos alunos.

6 REFERÊNCIAS BIBLIOGRÁFICAS

AZEVEDO, G. T. de. **Construção de conhecimento matemático a partir da produção de jogos digitais em um ambiente construcionista de aprendizagem**: possibilidades e desafios. Dissertação (Mestrado em Educação em Ciências e Matemática) – Instituto de Matemática e Estatística, Universidade Federal de Goiás, Goiânia, 2017.

AZEVEDO, G. T. de. **Processo formativo em Matemática: invenções robóticas para o Parkinson**. 2022. Tese (Doutorado em Educação Matemática) - Universidade Estadual Paulista (Unesp). Rio Claro, 2022.

BARBOSA, Lara Martins. **Aspectos do Pensamento Computacional na Construção de Fractais com o software GeoGebra**. Dissertação (Mestrado em Educação Matemática) - Universidade Estadual Paulista (Unesp), 2019. Rio Claro, 2019, 168 f.

BARR, Valerie; STEPHENSON, Chris. **Bringing computational thinking to K-12**: what is Involved and what is the role of the computer science education community?. *Acm Inroads*, v. 2, n. 1, p. 48-54, 2011.

BARCELOS, Thiago Schumacher. **Relações entre o Pensamento Computacional e a Matemática em atividades didáticas de construção de jogos digitais**. 2014. Tese (Doutorado em Ensino de Ciências e Matemática) - Universidade Cruzeiro do Sul, São Paulo.. Defesa realizada em 27 de março de 2014.

[BARDIN, Laurence. **Análise de conteúdo**. São Paulo: Edições 70, 2011, 229 p.]

BARNES, D. J. **Debugging: The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems**. No Starch Press, 2010.

BBC BITESIZE. **Introduction to Computational Thinking**. 2015. Disponível em: <https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1>. Acesso em: 21 jun. 2022.

BELL, T.; WITTEN, I. H.; FELLOWS, M. **Computer Science Unplugged**: Off-line activities and games for all ages. [S.l.]: University of Canterbury, 1998.

BICUDO, Maria Aparecida Viggiani. **Pesquisa Qualitativa**: Características, Usos e Possibilidades. São Paulo: UNESP, 2006. 107 p.

BLIKSTEIN, Paulo. **O pensamento computacional e a reinvenção do computador na educação**. Blikstein, 2008. Disponível em: http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html. Acesso em: 22 jun. 2022.

BOGDAN, Roberto C.; BIKLEN, Sari Knopp. **Investigação qualitativa em educação**. Tradução Maria João Alvarez, Sara Bahia dos Santos e Telmo Mourinho Baptista. Porto: Porto Editora, 1994.

BOGDAN, R. C.; BIKLEN, S. K. **Qualitative Research for Education: An Introduction to Theory and Methods**. Boston: Allyn and Bacon, 1982.

BRACKMANN, Christian. **Desenvolvimento do Pensamento Computacional Através de Atividades Desplugadas na Educação Básica**. 2017. Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, RS, Brasil, 2017. Disponível em: <https://hdl.handle.net/10183/172208>.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Versão Final. Brasília: MEC, 2017.

BRASIL a. (2018). **Base Nacional Comum Curricular (BNCC)**. Educação é a Base. Brasília, MEC/CONSED/UNDIME, 2018. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 20 jun. 2023.

BRASIL e. (2022). **Resolução N° 1, de 4 de outubro de 2022**. Normas sobre Computação na Educação Básica - Complemento à BNCC. Disponível em: <https://www.in.gov.br/en/web/dou/-/resolucao-n-1-de-4-de-outubro-de-2022-434325065>. Acesso em: 20 jun. 2023.

BUZATO, M. **Cultura digital e apropriação ascendente: apontamentos para uma educação 2.0**. Educação em Revista, Belo horizonte, v. 26, n. 3, p. 283-303, dez. 2010. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=s0102-46982010000300014&lng=pt&nrm=iso. Acesso em: 18 jun. 2022.

CAMPOS, Fabrício Vieira. **Pensamento Computacional: o uso do Scratch no ensino de Ciências**. 2021. Dissertação (Programa de Educação para Ciências e Matemática) - Instituto Federal de Educação, Ciência e Tecnologia de Goiás. Defesa realizada em 24 de março de 2021.

CAMPOS, Flavio Rodrigues. **A robótica para uso educacional**. São Paulo: Senac, 2019.

DELORS, J. **Educação: um tesouro a descobrir**. 2ª. ed. São Paulo: Cortez/MEC-Unesco, 1998.

DEMO, Pedro. **Educar pela pesquisa**. 6ª edição. São Paulo: Autores Associados, 2011.

DENNING, P. J. **Remaining Trouble Spots with Computational Thinking**. **Communications of the ACM**, New York, v. 60, n. 6, p. 33-39, 2017. Disponível em: <http://denninginstitute.com/pjd/PUBS/CACMcols/cacm-trouble-ct.pdf>. Acesso em: 20 jun. 2022.

- SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. **Diretrizes para ensino de Computação na Educação Básica.** 2019. Disponível em: <https://www.sbc.org.br/educacao/diretrizes-para-ensino-de-computacao-na-educacao-basica>. Acesso em: 22 jun. 2022.
- INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **Estimativa populacional 2021.** 2021. Disponível em: <https://cidades.ibge.gov.br/brasil/go/ipameri/panorama>. Acesso em: 16 jun. 2022.
- KALELIOĞLU, Filiz ; GÜLBAHAR, Yasemin ; KUKUL, Volkan. **A Framework for Computational Thinking Based on a Systematic Research Review.** Baltic J. Modern Computing, Vol. 4 (2016).
- FREIRE, P. **Educação como prática da liberdade.** 23ª.ed. Rio de Janeiro, RJ: Editora Paz e Terra, 1999.
- FREIRE, P. **Pedagogia da autonomia:**saberes necessários à prática educativa. 31ª. ed. Rio de Janeiro: Paz e Terra, 2005.
- FREIRE, Paulo. **Pedagogia do oprimido,** 17ª. ed. Rio de Janeiro, Paz e Terra, 1987.
- FURBER, S. et al. (2012). **Shut down or restart? The way forward for computing in UK Schools.** London: The Royal Society. Disponível em: http://royalsociety.org/uploadedFiles/Royal_Society_Content/education/policy/computing-in-schools/2012-01-12-Computing-in-Schools.pdf. Acesso em: 26 jun. 2022.
- GADANIDIS, George. **Five Affordances of Computational Thinking to Support Elementary Mathematics Education.** Journal of Computers in Mathematics and Science Teaching, v.36, n. 2, p.143-151, 2017.
- GIL, Antonio Carlos. **Como Elaborar Projetos de Pesquisa.** 5ª ed. São Paulo: Atlas, 2002.
- GOOGLE FOR EDUCATION. **What is Computational Thinking?** Computational Thinking for Educators, 2015. Disponível em: <https://computationalthinkingcourse.withgoogle.com/unit?lesson=8&unit=1>. Acesso em: 22 jun. 2022.
- GROVER, S., & PEA, R. **Computational thinking in K-12:** A review of the state of the field. Educational Researcher, 42(1), 38-43, 2013.
- GUZDIAL, M. **Education in and for the computing in the disciplines.** Communications of the ACM, 51(8), 28-31, 2008.
- HAREL, I., & Papert, S. **Software design as a learning environment.** Interactive Learning Environments, 1(1), 1-32.. Educational Researcher, 42(1), 38-43, 1990.
- HAYKIN, S. **Redes neurais e máquinas de aprendizado.** 3. ed. Pearson Education, 2009.

INSTITUTO FEDERAL GOIANO (IF GOIANO). Plano de Desenvolvimento Institucional 2019-2023. 2018. Disponível em: <https://www.ifgoiano.edu.br/home/index.php/pdi-2019-2023.html>. Acesso em: 16 jun. 2022.

LANKSHEAR, Colin; KNOBEL, Michele. **New Literacies**: Changing Knowledge and Classroom Learning. Open University Press, 2004.

LIUKAS, L. **Hello Ruby**: adventures in coding. Feiwel & Friends, 2015.

LOPES, Handley Magno Bernardo. **Do Desplugado Ao Plugado**: Uma Proposta Para O Desenvolvimento Do Pensamento Computacional E Do Pensamento Matemático Avançado Em Aulas Do Ensino Médio. 2022. Dissertação (Programa de Educação em Ciências e Matemática) - Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo. Defesa realizada em 28 de junho de 2022.

KURSHAN, B. **Thawing from a Long Winter in Computer Science Education**. Forbes, p. 25, fev. 2016. Disponível em: <https://www.forbes.com/sites/barbarakurshan/2016/02/25/thawing-from-a-long-winter-in-computer-science-education/?sh=3447daca284d>. Acesso em: 16 jun. 2022.

KRAUSS, Jane; PROTTSMAN, Kiki. **Computational Thinking and Coding for Every Student**. California: Corwin, 2016. 184 p.

MARQUES, Mônica; CAVALHEIRO, Simone; FOSS, Luciana; BORDINI, Adriana; ÁVILA, Christiano. **Uma Proposta para o Desenvolvimento do Pensamento Computacional Integrado ao Ensino de Matemática**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 28, 2017, Recife. Anais... Pernambuco: Sociedade Brasileira de Computação, 2017.

MALONEY, J., RESNICK, M., RUSK, N., SILVERMAN, B., & EASTMOND, E. **The scratch programming language and environment**. ACM Transactions on Computing Education, v. 10, n. 4, 1-15, 2010.

MALTEMPI, M. V. **Construção de Páginas Web: Depuração e Especificação de um Ambiente de Aprendizagem**. Campinas: Faculdade de Engenharia Elétrica e de Computação, 2000. 197p. Tese de Doutorado.

MYERS, Glenford J. **The Art of Software Testing**. Wiley, 2011.

MORAN, J. e BACICH, L. **Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática**. Porto Alegre: Penso Editora LTDA, 2018.

PATTON, M. Q. **Enhancing the quality and credibility of qualitative analysis**. Health Services Research, v. 34, n. 5 Pt 2, p. 1189-1208, 1999. Disponível em: <https://doi.org/10.1097/00005650-199910001-00002>. Acesso em: 27 set. 2023.

- PAPERT, S. **Situating Constructionism**. In: HAREL, I.; PAPERT, S. (Eds.). *Constructionism*. Norwood, NJ: Ablex Publishing, p. 1-11, 1991.
- PAPERT, S. **A máquina das Crianças: repensando a escola na era informática**. Porto Alegre: Artes Médicas, 2008.
- PAPERT, Seymour. **A família em rede: ultrapassando a barreira digital entre gerações**. Trad. Fernando José Silva Nunes e Fernando Augusto Bensabat Lacerda e Melo. Lisboa: Relógio D Água, 1997.
- PAPERT, S. **An exploration in th espace of mathematics educations**. *International Journal of Computers for Mathematical Learning*, Boston, v. 1, n. 1, p. 95-123, 1996.
- PAPERT, S. **Logo: Computadores e Educação**. São Paulo: Brasiliense, 1988.
- PAPERT, S. **Mindstorms: Children, Computers and Powerful Ideas**. New York: Basic Books Inc., 1980.
- PAULA, B. H. d.; VALENTE, J. A. **Jogos digitais e educação: uma possibilidade de mudança da abordagem pedagógica no ensino formal**. *Revista Ibero-americana de Educação*, [S.l.], v.70, n.1, p.9–28, 2016.
- PRETTO, N.; PINTO, C. d. C. **Tecnologias e novas educações**. *Revista Brasileira de Educação*, [S.l.], v.11, n.31, p.19–30, 2006.
- PRISMA. **Transparent Reporting of Systematic Reviews and Meta-Analyses**. Disponível em:
< <https://www.prisma-statement.org/> >. Acesso em: 10 nov. 2023.
- RESNICK, M. **All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten**. *ACM SIGCHI Bulletin*, v. 39, n. 1, p. 1-9, 2007.
- RESNICK, M. **Lifelong Kindergarten: cultivating Creativity through projects, passion, peers and play**. 1. ed. Cambridge, Ma: MIT Press, 2017.
- RESNICK, M. **New frameworks for studying and assessing the development of computational thinking**. *Annual Review of Information Science and Technology*, v. 46, n. 1, p. 1-52, 2012.
- RODRIGO, R. **Pensamento computacional e seus 4 pilares**. Terra, 2021. Disponível em: <https://www.markezine.com.br/educacao/pensamento-computacional-e-seus-4-pilares/>. Acesso em: 20 jun. 2022.
- RILEY, David D.; HUNT. Kenny A. **Computational Thinking for the Modern Problem Solver**. Massachusetts: CRC Press, 2014. 389 p.

ROKCODER. SB3 to TXT. Disponível em: <https://rokcoder.com/sb3-to-txt/>. Acesso em: 12 dez. 2023.

SAWAYA, Márcia Regina. **Dicionário de Informática & Internet Inglês/Português**. São Paulo: Nobel, 3ª edição, 1999.

SERRES, M. **Polegarzinha**. Rio de Janeiro: Bertrand Brasil, 2013.

SETTE, S. S.; AGUIAR, M. A. e SETTE, J. A. 1999. **Um caminho para mudanças**. MEC/PROINFO (Coleção Informática para mudança na Educação). 1999.

SILVA, Vladimir; SOUZA, Aryesha; MORAIS, Dyego. **Pensamento Computacional no Ensino de Computação em Escolas: Um relato de Experiência de Estágio em Licenciatura em Computação em Escolas Públicas**. In: Congresso Regional sobre Tecnologias na Educação, 2016. Página 326.

SOARES, Neto, JOAQUIM José et al. **Uma escala para medir a infraestrutura escolar**. Est. Aval. Educ., São Paulo, v. 24, n. 54, p. 78-99, jan./abr. 2013.

SOUZA, Alexandra Aparecida. **Framework baseado em Ciência de Dados e Engenharia de Software para Analisar a Evolução de Habilidades e Competências do Pensamento Computacional**. 2019 . Tese (Doutorado) - Universidade Presbiteriana Mackenzie.

TRIVIÑOS, Augusto Nivaldo Silva. **Introdução ao Estudo Qualitativo da Educação**. In: Lüdke, Menga; André, Marli Eliza Dalmazo Afonso de. Pesquisa em Educação: Abordagens Qualitativas. 5ª ed. São Paulo: EPU, 2013.

VALENTE, J. A. **O uso inteligente do computador na educação**. Pátio Revista Pedagógica. Editora: Artes Médicas Sul, ano 1, v. 1, p. 19-21, 1997.

WEBB, H. C. **Injecting computational thinking into computing activities for middle school girls**. 2013. 1 recurso online (p. 1-150). Tese (doutorado) - Universidade Estadual de Pensilvânia, Pensilvânia, 2013. Disponível em: <https://etda.libraries.psu.edu/catalog/18755>. Acesso em: 10 nov. 2023.

WING, J. M. “**Computational thinking**”, Commun. ACM, vol. 49, no. 3, p. 33-35, 2006.

WING, J. M. “**Computational thinking and thinking about computing**” Philos. Trans. A. Math. Phys. Eng. Sci., vol. 366, no. 1881, pp. 3717–25, 2008.

WING, J. **PENSAMENTO COMPUTACIONAL – Um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar**. Revista Brasileira de Ensino de Ciência e Tecnologia, v. 9, n. 2, 2016. Disponível em: <https://periodicos.utfpr.edu.br/rbect/article/view/4711>. Acesso em: 03 jul. 2022.

ANEXOS



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO - PPGEA

Termo de Assentimento Livre e Esclarecido (TALE)

Conselho Nacional de Saúde, Resolução 466/2012/Resolução 510/2016

Caro aluno, você está sendo convidado(a), como voluntário(a), a participar da pesquisa intitulada **Averiguação das PRODUÇÃO DE JOGOS DIGITAIS E PENSAMENTO COMPUTACIONAL: UMA ABORDAGEM CONSTRUCIONISTA DE APRENDIZAGEM**. Após receber os esclarecimentos e as informações, a seguir, no caso de aceitar fazer parte do estudo, este documento deverá ser assinado em duas vias, sendo a primeira de guarda e confidencialidade do pesquisador responsável e a segunda ficará sob sua responsabilidade para quaisquer fins. Em caso de recusa, você não será penalizado(a) de forma alguma. Em caso de dúvida sobre a pesquisa, você poderá entrar em contato com o pesquisador responsável Deivid do Vale Nascimento, através do telefone: (64) 98438-8302 ou através do e-mail deividovale@hotmail.com. Em caso de dúvida sobre a ética aplicada à pesquisa, você poderá entrar em contato com o Comitê de Ética em Pesquisa (CEP) da Universidade de Iguazu (UNIG) - CAMPUS I, localizado no Bloco A, Sala 103. Av. Abílio Augusto Távora, nº 2134 – Jardim Nova Era. CEP: 26.275-580, Nova Iguaçu, telefone: (21) 2765-4039, e-mail: cep@campus1.unig.br.

1 – Justificativa, objetivos e procedimentos:

O objetivo desta pesquisa é identificar e compreender o desenvolvimento do Pensamento Computacional por estudantes do Ensino Médio durante a criação de jogos digitais e dispositivos robóticos associados, em um contexto construcionista de aprendizado. Isso se justifica pela importância do pensamento computacional na vida acadêmica e social do estudante, sendo que este favorece o desenvolvimento de habilidades que auxiliam o indivíduo no planejamento e resolução de problemas do dia a dia e de diferentes áreas do conhecimento.

Ao observar as dez competências gerais, que deverão ser desenvolvidas por estudantes da educação básica, propostas pela Base Nacional Comum Curricular (BNCC), nota-se que

dentre elas existe a Cultura digital que propõe a utilização e criação de tecnologias digitais de forma crítica significativa e ética para comunicar-se, acessar e produzir informações e conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva. Ela também define diversas competências e habilidades que permitem aos educandos a apropriação do conhecimento e a autonomia, de modo que eles consigam, por exemplo: “Utilizar, propor e/ou implementar soluções (processos e produtos) envolvendo diferentes tecnologias, para identificar, analisar, modelar e solucionar problemas complexos em diversas áreas da vida cotidiana, explorando de forma efetiva o raciocínio lógico, o pensamento computacional, o espírito de investigação e a criatividade”. Atualmente, diversas universidades e escolas do mundo adotam na grade curricular a disciplina de Pensamento Computacional, o objetivo é que cada vez mais os adolescentes e crianças saibam como estruturar e resolver de forma mais eficiente os problemas propostos a elas em sala de aula. Na escola, o pensamento computacional faz com que o estudante, apresente um maior desenvolvimento, quando desde cedo, começa a desenvolver esse pensamento, aprendendo a construir uma forma de pensar logicamente, fazendo com que a facilidade de resolver outros desafios de outras disciplinas acabe sendo maior.

A pesquisa será realizada no Campus Avançado Ipameri do Instituto Federal Goiano. Serão selecionados 18 estudantes para participarem da pesquisa. Na fase de coleta de dados serão utilizados 4 instrumentos, sendo eles: diário de campo, registros fotográficos dos encontros e jogos desenvolvidos, questionários, observação em sala de aula. Todas essas ferramentas serão aplicadas durante todos os encontros do projeto.

Na etapa de coleta de dados através da observação em sala de aula, o pesquisador observará como o professor introduzirá os conceitos de lógica de programação, como ele irá propor os desafios a serem resolvidos pelos alunos, o comportamento dos alunos ao se depararem com os desafios, a interação entre esses alunos, as ferramentas utilizadas por estes, além de comportamentos e eventos significativos inesperados que podem ocorrer durante todo o processo. Todos os eventos serão anotados em um diário/nota de campo, serão produzidos dois tipos de escritas, uma descritiva e outra interpretativa/reflexiva. Já na coleta de dados através de registros fotográficos, instrumento esse que será utilizado juntamente a observação em sala de aula, o pesquisador capturará trechos de códigos de jogos desenvolvidos pelos estudantes, a fim de identificar conceitos, práticas e perspectivas do Pensamento Computacional.

2 – Desconfortos, riscos e benefícios:

Ao consentir em participar deste estudo, você poderá ser submetido aos seguintes riscos: Cansaço, desconforto pelo tempo gasto para responder questionários, talvez desconforto quanto ao horário e local, além do tempo gasto em responder às perguntas ou, ainda, riscos quanto ao compartilhamento de informações pessoais ou confidenciais, bem como manifestação a respeito de algum assunto que sinta incômodo em falar. Para minimizar esses riscos, haverá garantia de local reservado, liberdade para não responder possíveis questões constrangedoras e acesso aos dados individuais e coletivos. Fica assegurada a inexistência de conflito de interesses entre o pesquisador e os participantes. Os benefícios oriundos de sua participação serão de contribuir para o melhor desenvolvimento de ações cada vez mais integradoras em relação ao ensino do IF Goiano nos cursos técnicos da modalidade integrada ao ensino médio.

3 – Forma de acompanhamento e assistência:

Você terá garantia de assistência integral em qualquer etapa do estudo. A pesquisa será readequada ou suspensa quando apresentar risco ao sigilo dos participantes ou em caso fortuito ao pesquisador, após justificativa aceita pelo CEP. Caso o estudo seja interrompido, o pesquisador firma o compromisso de encaminhar um comunicado a todos os envolvidos.

4 – Garantia de esclarecimento, liberdade de recusa e garantia de sigilo:

Você será esclarecido(a) sobre a pesquisa em qualquer tempo e aspecto que desejar, através dos meios citados acima. Você é livre para recusar-se a participar, retirar seu consentimento ou interromper a participação a qualquer momento, sendo sua participação voluntária e a recusa em participar não acarretará qualquer penalidade. O pesquisador tratará a sua identidade com padrões profissionais de sigilo e todos os dados coletados servirão apenas para fins de pesquisa. Seu nome ou o material que indique a sua participação não será liberado sem a sua permissão.

Por meio da sua autorização, fotografias poderão ser utilizadas para finalidade didática e científica, divulgadas em aulas, palestras, conferências, cursos, congressos, etc. e também publicadas em livros, artigos, portais de internet, redes sociais, revistas científicas e similares, podendo inclusive ser mostrado rostos dos participantes.

5 – Custos da participação, ressarcimento e indenização por eventuais danos:

Para participar deste estudo, você não terá nenhum custo nem receberá qualquer vantagem financeira. Caso você, participante, sofra algum dano decorrente desta pesquisa, os pesquisadores garantem indenizá-lo por todo e qualquer gasto ou prejuízo.

**TERMO DE AUTORIZAÇÃO DA PARTICIPAÇÃO DO ESTUDANTE NA
PESQUISA**

Eu, _____, RG _____ aceito participar da pesquisa de mestrado, que será realizado nas dependências do IF-Goiano, Campus Ipameri, e no Hospital do Idoso, em Anápolis. Afirmando que fui devidamente informado (a) e esclarecido (a) sobre os objetivos e os procedimentos das atividades de pesquisa. Se você autorizar, seu nome e imagem poderão ser divulgados em trabalhos científicos (artigos) e/ou de divulgação científica ou, caso prefira, ficarão sob sigilo. No caso de aceitar fazer parte deste estudo, assine ao final deste documento, que está em duas vias. Uma delas é sua e a outra é do pesquisador responsável.

Quanto ao uso do meu nome e imagem, faço a opção de:

() Autorizo o uso de meu nome e imagem () Quero manter sigilo de minha identidade

Ipameri, 01 de março de 2023.

Assinatura do estudante

Deivid do Vale Nascimento
- Pesquisador Responsável -



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO PPGEA

Termo de Consentimento Livre e Esclarecido (TCLE)

Conselho Nacional de Saúde, Resolução 466/2012/Resolução 510/2016

O menor sob sua responsabilidade está sendo convidado(a), como voluntário(a), a participar da pesquisa intitulada **Averiguação das PRODUÇÃO DE JOGOS DIGITAIS E PENSAMENTO COMPUTACIONAL: UMA ABORDAGEM CONSTRUCIONISTA DE APRENDIZAGEM**, que se realizará no Instituto Federal Goiano, campus avançado Ipameri, sob a responsabilidade do pesquisador Deivid do Vale Nascimento. Esta pesquisa tem por objeto principal compreender o processo do desenvolvimento do Pensamento Computacional de estudantes quando constroem jogos digitais e desenvolvem dispositivos robóticos destinados ao tratamento de sintomas da doença de Parkinson. A pesquisa será realizada no cenário do Projeto Mattics e o menor sob sua responsabilidade não será remunerado. Se porventura decidir autorizar a participação dele (a) nesta pesquisa, o aluno terá toda autonomia para desistir dela a qualquer momento, sem nenhuma forma de penalização ou danos. Ressaltamos que, além de estar em contato com novas tecnologias digitais, esta pesquisa pretende também contribuir para a aprendizagem do aluno para além do currículo, colaborando com uma formação mais significativa à sociedade. Em caso de dúvida sobre a pesquisa, você poderá entrar em contato com o pesquisador responsável Deivid do Vale Nascimento, através do telefone: (64) 98438-8302 ou através do e-mail deividovale@hotmail.com. Em caso de dúvida sobre a ética aplicada à pesquisa, você poderá entrar em contato com o Comitê de Ética em Pesquisa (CEP) da Universidade de Iguaçu (UNIG) - CAMPUS I, localizado no Bloco A, Sala 103. Av. Abílio Augusto Távora, nº 2134 – Jardim Nova Era. CEP: 26.275-580, Nova Iguaçu, telefone: (21) 2765-4039, e-mail: cep@campus1.unig.br.

1 – Justificativa, objetivos e procedimentos:

O objetivo desta pesquisa é identificar e compreender o desenvolvimento do Pensamento Computacional por estudantes do Ensino Médio durante a criação de jogos digitais e

dispositivos robóticos associados, em um contexto construcionista de aprendizado. Isso se justifica pela importância do pensamento computacional na vida acadêmica e social do estudante, sendo que este favorece o desenvolvimento de habilidades que auxiliam o indivíduo no planejamento e resolução de problemas do dia a dia e de diferentes áreas do conhecimento.

Ao observar as dez competências gerais, que deverão ser desenvolvidas por estudantes da educação básica, propostas pela Base Nacional Comum Curricular (BNCC), nota-se que dentre elas existe a Cultura digital que propõe a utilização e criação de tecnologias digitais de forma crítica significativa e ética para comunicar-se, acessar e produzir informações e conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva. Ela também define diversas competências e habilidades que permitem aos educandos a apropriação do conhecimento e a autonomia, de modo que eles consigam, por exemplo: “Utilizar, propor e/ou implementar soluções (processos e produtos) envolvendo diferentes tecnologias, para identificar, analisar, modelar e solucionar problemas complexos em diversas áreas da vida cotidiana, explorando de forma efetiva o raciocínio lógico, o pensamento computacional, o espírito de investigação e a criatividade”. Atualmente, diversas universidades e escolas do mundo adotam na grade curricular a disciplina de Pensamento Computacional, o objetivo é que cada vez mais os adolescentes e crianças saibam como estruturar e resolver de forma mais eficiente os problemas propostos a elas em sala de aula. Na escola, o pensamento computacional faz com que o estudante, apresente um maior desenvolvimento, quando desde cedo, começa a desenvolver esse pensamento, aprendendo a construir uma forma de pensar logicamente, fazendo com que a facilidade de resolver outros desafios de outras disciplinas acabe sendo maior.

A pesquisa será realizada no Campus Avançado Ipameri do Instituto Federal Goiano. Serão selecionados 18 estudantes para participarem da pesquisa. Na fase de coleta de dados serão utilizados 4 instrumentos, sendo eles: diário de campo, registros fotográficos dos encontros e jogos desenvolvidos, questionários, observação em sala de aula. Todas essas ferramentas serão aplicadas durante todos os encontros do projeto.

Na etapa de coleta de dados através da observação em sala de aula, o pesquisador observará como o professor introduzirá os conceitos de lógica de programação, como ele irá propor os desafios a serem resolvidos pelos alunos, o comportamento dos alunos ao se depararem com os desafios, a interação entre esses alunos, as ferramentas utilizadas por estes, além de comportamentos e eventos significativos inesperados que podem ocorrer durante todo o processo. Todos os eventos serão anotados em um diário/nota de campo, serão produzidos

dois tipos de escritas, uma descritiva e outra interpretativa/reflexiva. Já na coleta de dados através de registros fotográficos, instrumento esse que será utilizado juntamente a observação em sala de aula, o pesquisador capturará trechos de códigos de jogos desenvolvidos pelos estudantes, a fim de identificar conceitos, práticas e perspectivas do Pensamento Computacional.

2 – Desconfortos, riscos e benefícios:

Ao consentir em participar deste estudo, você poderá ser submetido aos seguintes riscos: Cansaço, desconforto pelo tempo gasto para responder questionários, talvez desconforto quanto ao horário e local, além do tempo gasto em responder às perguntas ou, ainda, riscos quanto ao compartilhamento de informações pessoais ou confidenciais, bem como manifestação a respeito de algum assunto que sinta incômodo em falar. Para minimizar esses riscos, haverá garantia de local reservado, liberdade para não responder possíveis questões constrangedoras e acesso aos dados individuais e coletivos. Fica assegurada a inexistência de conflito de interesses entre o pesquisador e os participantes. Os benefícios oriundos de sua participação serão de contribuir para o melhor desenvolvimento de ações cada vez mais integradoras em relação ao ensino do IF Goiano nos cursos técnicos da modalidade integrada ao ensino médio.

3 – Forma de acompanhamento e assistência:

Você terá garantia de assistência integral em qualquer etapa do estudo. A pesquisa será readequada ou suspensa quando apresentar risco ao sigilo dos participantes ou em caso fortuito ao pesquisador, após justificativa aceita pelo CEP. Caso o estudo seja interrompido, o pesquisador firma o compromisso de encaminhar um comunicado a todos os envolvidos.

4 – Garantia de esclarecimento, liberdade de recusa e garantia de sigilo:

Você será esclarecido(a) sobre a pesquisa em qualquer tempo e aspecto que desejar, através dos meios citados acima. Você é livre para recusar-se a participar, retirar seu consentimento ou interromper a participação a qualquer momento, sendo sua participação voluntária e a recusa em participar não acarretará qualquer penalidade. O pesquisador tratará a sua identidade com padrões profissionais de sigilo e todos os dados coletados servirão apenas para fins de pesquisa. Seu nome ou o material que indique a sua participação não será liberado sem a sua permissão.

Por meio da sua autorização, fotografias poderão ser utilizadas para finalidade didática e científica, divulgadas em aulas, palestras, conferências, cursos, congressos, etc. e também

publicadas em livros, artigos, portais de internet, redes sociais, revistas científicas e similares, podendo inclusive ser mostrado rostos dos participantes.

5 – Custos da participação, ressarcimento e indenização por eventuais danos:

Para participar deste estudo, você não terá nenhum custo nem receberá qualquer vantagem financeira. Caso você, participante, sofra algum dano decorrente desta pesquisa, os pesquisadores garantem indenizá-lo por todo e qualquer gasto ou prejuízo.

Para participantes menores de 18 anos (crianças e adolescentes) ou pessoas com transtorno ou doença mental ou em situação de substancial diminuição em sua capacidade de decisão:

Ciente e de acordo com o que foi anteriormente exposto, eu _____ CPF _____, responsável legal pelo (a) aluno _____

_____ autorizo sua participação no estudo intitulado **PRODUÇÃO DE JOGOS DIGITAIS E PENSAMENTO COMPUTACIONAL: UMA ABORDAGEM CONSTRUCIONISTA DE APRENDIZAGEM**, desde que o (a) mesmo (a) aceite de forma livre e espontânea, e que possa se retirar a qualquer momento. Ciente e de acordo com o que foi anteriormente exposto, eu _____

_____ estou de acordo em participar desta pesquisa acima descrita.

Quanto ao uso da imagem do referido aluno, faço a opção de:

() Autorizo o uso do nome e imagem () Quero manter sigilo da identidade

Ipameri, 01 de março de 2023.

Assinatura do participante (quando possível)

Assinatura do responsável legal

Deivid do Vale Nascimento

- Pesquisador Responsável -



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO PPGEA

**PRODUÇÃO DE JOGOS DIGITAIS E PENSAMENTO COMPUTACIONAL: UMA
ABORDAGEM CONSTRUCIONISTA DE APRENDIZAGEM**

Caro aluno, este formulário faz parte de uma pesquisa de mestrado conduzida por Deivid do Vale Nascimento no Programa de Pós-Graduação em Educação Agrícola da UFRRJ. O trabalho tem como foco a interseção entre a produção de jogos digitais e o pensamento computacional, adotando uma abordagem construcionista de aprendizagem.

Ao aceitar continuar, você estará consentindo voluntariamente em participar desta pesquisa. Isso implica que concorda com o compartilhamento de informações pertinentes ao estudo, como suas respostas em entrevistas, observações e possíveis documentos relacionados à produção de jogos digitais. Ressaltamos que todas as informações fornecidas serão tratadas com estrita confidencialidade, garantindo o anonimato dos participantes. Além disso, você terá o direito de retirar seu consentimento a qualquer momento, sem qualquer penalidade. Se tiver dúvidas ou preocupações sobre sua participação, sinta-se à vontade para entrar em contato com o pesquisador responsável, Deivid do Vale Nascimento, para esclarecimentos adicionais.

Este formulário faz parte de um esforço maior para promover práticas inovadoras de ensino e aprendizagem na Educação, utilizando a produção de jogos digitais como uma ferramenta construcionista para o desenvolvimento do pensamento computacional.

QUESTIONÁRIO

- 1 - Qual o seu E-mail?
- 2 - Qual o seu nome?
- 3 - Quantos anos tem?
- 4 - Qual o seu curso?
- 5 - Qual jogo você ajudou a desenvolver?
- 6 - Poderia destacar uma funcionalidade específica do seu jogo em que você aplicou com sucesso algum conceito de loops, eventos, sequências, paralelismo, condicionais, operadores lógicos ou dados?
- 7 - Como você descreveria o processo de desenvolvimento do seu último projeto de jogo? Foi um processo sequencial ou mais adaptativo e iterativo?
- 8 - Como você lida com mudanças no plano de design durante o processo de desenvolvimento do jogo? As mudanças eram constantes?
- 9 - Pode compartilhar um exemplo específico de um problema que você enfrentou durante o desenvolvimento de um jogo/dispositivo robótico e como lidou com isso?
- 10 - Quais são as estratégias que você utiliza para testar e depurar seu jogo? Você tem alguma prática específica que considera eficaz?
- 11- Você já reutilizou ou remixou código de outros projetos em seus jogos? Se sim, pode compartilhar uma experiência específica e como isso contribuiu para o seu projeto?
- 12 - Como você dividiu o jogo em partes?
- 13 - Isso facilitou na hora de consertar alguma coisa?
- 14 - O que você acha da computação nos jogos? É mais do que só mexer no computador?
- 15 - Comparando a criação de jogos no Scratch com outras atividades online, como redes sociais, o que diferencia a experiência para você?
- 16 - Como a interação com outras pessoas influencia a sua prática criativa no desenvolvimento de jogos digitais?
- 17 - Você já experimentou benefícios ao criar em colaboração com outros estudantes? Se sim, como isso impactou seus projetos?
- 18 - Na comunidade online Scratch, você acha que ter acesso a diferentes pessoas, projetos e perspectivas enriqueceu sua experiência de criação de jogos? Como?
- 19 - Você acha importante se questionar enquanto está criando jogos?
- 20 - Já pensou em mudar alguma coisa nas ferramentas que usa pra criar jogos?

21 - Como você descreveria o processo de abstração em seu trabalho de desenvolvimento de jogos digitais? Em que medida você simplifica ou representa conceitos complexos para torná-los mais compreensíveis no contexto do seu jogo?

22 - Ao enfrentar desafios complexos na criação de jogos, como você divide esses problemas em partes menores mais gerenciáveis? Pode fornecer um exemplo específico de como a decomposição foi útil em um de seus projetos?

23 - Como o reconhecimento de padrões influencia suas decisões durante o desenvolvimento de jogos? Você pode compartilhar uma situação em que identificar padrões foi fundamental para aprimorar a jogabilidade ou a experiência do usuário?

24 - Na elaboração de lógicas e sequências para as funcionalidades dos jogos, como você projeta e implementa algoritmos? Pode dar um exemplo de como um algoritmo específico foi crucial para o funcionamento de um elemento importante em um de seus jogos?

Disponível em :

<https://docs.google.com/forms/d/e/1FAIpQLSdZRI3LxZOx7HFbRFznRdpBXKDaOy1THpbgLUJ1FYK7ij0Ng/viewform>